

UNIVERZA NA PRIMORSKEM  
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN  
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA

ZAKLJUČNA NALOGA  
PROTOTIP EKSPERTNEGA SISTEMA ZA  
MODELIRANJE INTERNETNIH GROŽENJ

JANI ZAJC

UNIVERZA NA PRIMORSKEM  
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN  
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

**Prototip ekspertnega sistema za modeliranje internetnih groženj**

(A Prototype Expert System for Modelling Internet Threats)

Ime in priimek: Jani Zajc

Študijski program: Računalništvo in informatika

Mentor: prof. dr. Cene Bavec

Koper, april 2013

**Ključna dokumentacijska informacija**

Ime in PRIIMEK:	Jani ZAJC				
Naslov zaključne naloge:	Prototip ekspertnega sistema za modeliranje internetnih groženj				
Kraj:	Koper				
Leto:	2013				
Število listov:	65	Število slik:	23	Število tabel:	11
Število prilog:	0	Št. strani prilog:	0		
Število referenc:	33				
Mentor:	prof. dr. Cene Bavec				
Somentor:	/				
UDK:					
Ključne besede:	ekspertni sistemi, sistemi za podporo odločanju, inteligentni sistemi, internetna varnost, internetne grožnje, Vensim, model, sistemska dinamika				
Izvleček:	<p>V zaključnem delu je obravnavana tematika ekspertnih sistemov. Med ekspertne sisteme se uvrščajo vsi pametni sistemi, ki so zmožni samostojno reševati probleme, ki jih normalno rešujejo visoko kvalificirane osebe na svojem področju – eksperti. Namen zaključnega dela je preučiti delovanje ekspertnih sistemov in zasnovati ter zgraditi lasten prototip ekspertnega sistema. V prvem delu je predstavljena teorija in uporaba ekspertnih sistemov, in sicer : definicija, prednosti, slabosti, karakteristike, zgodovinski razvoj in primeri uporabe v realnem svetu. V drugem delu smo teorijo uporabili v praksi in izgradili prototip ekspertnega sistema za modeliranje internetnih groženj. Opisali smo strukturo, logiko in razvojni postopek sistema (od idejne zasnove do končne celote). Prototip smo izgradili v okolju programskega paketa Vensim – orodja za simulacijo modelov v sistemske dinamiki. Izgrajen model je bilo potrebno uravnati, preden smo s simulacijo dobili smiselne rezultate. Na podlagi rezultatov zaključnega dela lahko ugotovimo, da ima paradigma ekspertnih sistemov v sebi velik potencial, saj lahko ekspertni sistem v veliki meri nadomesti človeškega eksperta</p>				

pri reševanju problemov in pojasnjevanju rešitev, a je njegova izdelava večinoma zelo zahtevna in draga, zato se velikokrat pojavi vprašanje, ali se sploh izplača.

**Key words documentation**

Name and SURNAME:	Jani ZAJC				
Title of the final project paper:	A Prototype Expert System for Modelling Internet Threats				
Place:	Koper				
Year:	2013				
Number of pages:	65	Number of figures:	23	Number of tables:	11
Number of appendix:	0	Number of appendix pages:	0		
Number of references:	33				
Mentor:	prof. dr. Cene Bavec				
Co-Mentor:	/				
UDC:					
Keywords:	expert systems, decision support systems, intelligent systems, internet security, internet threats, Vensim, model, system dynamics				
Abstract:	<p>In this final work it is treated the topic of expert systems. The topic includes all intelligent systems that are able to solve problems independently, problems that are normally resolved by highly qualified people in a field - experts. The objective of the final work is to examine the operation of expert systems to such an extent to be able to design and build our prototype expert system. In the first part we go through the theory and applications of expert systems, i.e.: definition, advantages, disadvantages, characteristics, historical development and real world applications. In the second part, we used the theory in practice and built a prototype expert system for modulation of Internet threats. We describe the structure and logic of the development process (from the concept to the final whole) of the system. We built the prototype in the software environment package Vensim - a tool for simulation of models in system dynamics. When the model was built it had to be adjusted with little corrections before getting meaningful simulation results. Final conclusion: the expert systems paradigm has great potential, because an expert system can (not fully, but to a large extent) replace a human expert in problem solving and solution explanation, but its production is mostly very complicated and expensive, often arises the question whether is it even worth the effort.</p>				

### **Zahvala**

Zahvaljujem se staršem in prijateljem za podporo. Zahvala gre tudi mentorju in lektorici za strokovno vodenje pri pisanju zaključne naloge.

**KAZALO**

1	Uvod .....	1
1.1	Opredelitev področja in opis problema .....	1
1.2	Namen in cilji .....	1
1.3	Predvidene metode raziskovanja .....	2
1.4	Pregled vsebine zaključne naloge .....	2
2	Teoretične osnove ekspertnih sistemov .....	3
2.1	Definicija .....	3
2.2	Prednosti ekspertnih sistemov .....	7
2.3	Splošni pojmi v ekspertnih sistemih .....	8
2.4	Karakteristike ekspertnih sistemov .....	11
2.5	Razvoj ekspertnih sistemov .....	14
2.5.1	Človeško reševanje problemov in produkcije .....	15
2.6	Rast ekspertnih sistemov baziranih na pravilih .....	19
2.7	Aplikacije ekspertnih sistemov in njihove domene .....	21
2.7.1	Aplikacije ekspertnih sistemov .....	21
2.7.2	Primerne domene za ekspertne sisteme .....	24
3	Uporaba ekspertnih sistemov .....	28
3.1	Sodobna uporaba ekspertnih sistemov na različnih področjih .....	29
3.1.1	Ekspertni sistemi v kmetijstvu .....	29
3.1.2	Ekspertni sistemi v izobraževanju .....	30
3.1.3	Ekspertni sistemi v medicini .....	33
4	Prototip ekspertnega sistema .....	36
4.1	Modelacijsko okolje .....	36
4.1.1	Vensim .....	36
4.1.2	Sistemska dinamika .....	36
4.1.3	Paradigma strukture in vedenja .....	37
4.1.4	Uporabniški vmesnik .....	37
4.2	Logika prototipa .....	37
4.2.1	Domensko znanje .....	37
4.2.2	Splošen opis logike .....	38

---

4.2.3	Motivacija za domeno .....	40
4.2.4	Relacijski diagram prototipa.....	40
4.3	Razvoj prototipa.....	40
4.3.1	Zajemanje znanja .....	40
4.3.2	Uporabniški vmesnik.....	41
4.3.3	Razvojna metoda .....	41
4.4	Opis (strukture) prototipa.....	43
4.4.1	Vir groženj – ključna spremenljivka 1 .....	44
4.4.2	Osebna ogroženost – ključna spremenljivka 2 .....	45
4.4.3	Razumevanje groženj – ključna spremenljivka 3 .....	46
4.5	Izvedba simulacije .....	47
4.5.1	Običajna simulacija modela .....	48
4.5.2	SyntheSim simulacija modela .....	48
4.6	Primer izvedbe .....	49
5	Zaključek .....	53
6	Literatura .....	54



## KAZALO TABEL

Tabela 1: Pogledi različnih udeležencev na IT tehnologijo; prirejeno iz Giarratano, C. J. and Riley, G. (2004) .....	7
Tabela 2: Pregled zgodovine razvoja ekspertnih sistemov; prirejeno iz Giarratano, C. J and Riley, G. (2004).....	15
Tabela 3: Različni tipi ekspertnih sistemov; prirejeno iz Giarratano, C. J.and Riley, G. (2004) .....	22
Tabela 4: Ekspertni sistemi v kemiji; prirejeno iz Giarratano, C. J. and Riley, G. (2004)..	23
Tabela 5: Ekspertni sistemi v elektroniki; prirejeno iz Giarratano, C. J and Riley, G. (2004) .....	23
Tabela 6: Ekspertni sistemi v medicini; prirejeno iz Giarratano, C. J and Riley, G. (2004)	23
Tabela 7: Ekspertni sistemi v strojništvu; prirejeno iz Giarratano, C. J and Riley, G. (2004) .....	23
Tabela 8: Ekspertni sistemi v geologiji; prirejeno iz Giarratano, C. J and Riley, G. (2004) .....	24
Tabela 9: Ekspertni sistemi v računalništvu; prirejeno iz Giarratano, C. J and Riley, G. (2004) .....	24
Tabela 10: Primer slabo strukturiranega problema (vprašanja in odgovori); prirejeno iz Giarratano, C. J. and Riley, G. (2004).....	25
Tabela 11: Odnos parametrov po vhodnih podatkih – državah.....	49

**KAZALO SLIK**

Slika 1: Nekatera področja umetne inteligence; prirejeno iz Giarratano, C. J. and Riley, G. (2004) .....	4
Slika 2: Osnovni koncept funkcij ekspertnega sistema; prirejeno iz Giarratano, C. J and Riley, G. (2004).....	5
Slika 3: Odnos problemske domene z domenskim znanjem; prirejeno iz Giarratano, C. J and Riley, G. (2004) .....	6
Slika 4: Razvoj ekspertnega sistema; prirejeno iz Giarratano, C. J. and Riley, G. (2004)..	10
Slika 5: Konvergenca treh pomembnih faktorjev pri razvoju modernih ekspertnih sistemih, osnovanih na pravilih; prirejeno iz Giarratano, C. J. and Riley, G. (2004).....	20
Slika 6: Struktura sistema PT za poučevanje študentov strojništva. Pridobljeno iz Zorica Nedić, Vladimir Nedić in Jan Machotka (2002).....	31
Slika 7: Arhitektura sistema PT. Prirejeno iz Jim Prentzas, Ioannis Hatzilygeroudis, C. Koustojannis (2001) .....	32
Slika 8: Osnovni gradniki Leonardovega ekspertnega sistema. Prirejeno iz Negnevitsky, M. (1998).....	32
Slika 9: Relacijski diagram prototipa .....	39
Slika 10: Vmesni prototip št. 1 – začetna ideja .....	42
Slika 11: Vmesni prototip št. 2 – dodelana ideja, dodelane entitete.....	42
Slika 12: Vmesni prototip št. 3 – izboljšava ideje z vključenimi spremembami .....	43
Slika 13: Končni prototip: struktura .....	44
Slika 14: Vzročno drevo spremenljivke <i>vir groženj</i> .....	45
Slika 15: Posledično drevo spremenljivke <i>vir groženj</i> .....	45
Slika 16: Vzročno drevo <i>osebne ogroženosti</i> .....	46
Slika 17: Vzročno drevo <i>razumevanja groženj</i> .....	47
Slika 18: Vplivno drevo <i>razumevanja groženj</i> .....	47
Slika 19: Preizkus modela v načinu sočasnega spreminjanja parametrov .....	49
Slika 20: Rezultat izvedbe – Razumevanje groženj .....	50
Slika 21: Rezultat izvedbe – Vir groženj.....	51
Slika 22: Rezultat izvedbe – Osebna ogroženost.....	52
Slika 23: Rezultat izvedbe – dodaten graf – Izkušnje uporabnikov .....	52

## 1 UVOD

### 1.1 Opredelitev področja in opis problema

Umetna inteligenca je področje v računalništvu, katerega cilj je razvoj naprav, ki se vedejo kot človek, torej kot da razpolagajo z inteligenco. Področje umetne inteligence združuje v sebi več ved, in sicer: psihologijo, filozofijo, matematiko, logiko in nevrologijo; njene veje pa so: vizualna, govorna, manipulativna in racionalna inteligenca.

V tem delu je aktualna podveja racionalne inteligence, ki obravnava *ekspertne sisteme*. Ekspertni sistemi so izraz za računalniški sistem, ki oponaša delo eksperta. Tak sistem zna s sofisticiranim sklepanjem sam priti do določenih zaključkov kot človeški strokovnjak in ne s predhodno napisanim postopkom, kakršen je običajen v konvencionalnem programiranju.

Velja opozoriti, da smo se odločili poimenovati področje dela področje ekspertnih sistemov. Na to področje v okviru te zaključne naloge spadajo poleg ekspertnih sistemov tudi inteligentni sistemi in sistemi za podporo odločanju.

Vsak sistem, ki se odziva s kompleksnim sklepanjem glede na vhodne podatke iz okolja, je inteligentni sistem. Ta izraz je precej splošen in se uporablja predvsem v zvezi s tehnično platjo pri tehnični strani vede o umetni inteligenci. Izraz ekspertni sistem naslavlja sistem, ki oponaša eksperta točno določenega področja. Taki sistemi so pogosti na področju prava, medicine in borznitstva [2]. Nazadnje se sistemi za podporo odločanju uporabljajo v velikih podjetjih, ker so v podporo odločanju vrhovnemu managementu, ko gre za pomembne strukturne ali druge odločitve glede podjetja.

### 1.2 Namen in cilji

Namen zaključnega dela je preučiti delovanje ekspertnih sistemov in zasnovati prototip ekspertnega sistema s ciljem predstavitve postopka zasnove in delovanje prototipa.

Tukaj smo bili primorani izbrati neko področje znanja prototipa ekspertnega sistema. Cilj prototipa je, da modelira varnostne grožnje za povprečnega uporabnika na internetu z različnimi načeli sklepanja in se, kolikor je možno, približa obnašanju uporabnika tako, da je velikost modela omejena za smiselnost zaključne naloge.

Cilje bomo poskusili doseči, tako da bomo v nadaljevanju zaključne naloge najprej dobro teoretično preučili delovanje ekspertnih sistemov, nato pa bomo s poudarkom na delovanju in smiselnosti izgradili prototip.

### 1.3 Predvidene metode raziskovanja

Za pripravo zaključne naloge izhajamo iz teoretičnih spoznanj, ki jih v nadaljevanju uporabimo pri zasnovi prototipa. Uporabljena je bila predvsem tuja literatura, saj je domače literature relativno malo.

Po raziskovalnem delu o teoriji ekspertnih sistemov smo se osredotočili na konkreten model. Postopek zasnove je tekel v tej smeri:

- najprej smo na papir skicirali idejo in jo dodelovali do nekega nivoja zapletenosti,
- nato smo model prevedli v programsko okolje Vensim,
- v Vensimu je bilo potrebno zagotoviti pravilnost modela ter pravilnost enot,
- ko je model brezhibno deloval, smo vanj vstavili realne podatke in ga smiselno obtežili.

Tako zgleda potek izgradnje modela, razdeljen v faze, čeprav smo model nenehno malenkostno spreminjali in dopolnjevali v vsaki od faz. Pri zasnovi modela so se nenehno pojavljale ovire, na katere nismo vnaprej načrtovali. To je tudi ena izmed prednosti pri zasnovi prototipov, da nam ni treba imeti točno določenega načrta vnaprej, saj bi le-ta zahteval veliko truda in časa.

### 1.4 Pregled vsebine zaključne naloge

Zaključno delo je razdeljeno v dva vsebinska sklopa: teoretične osnove ekspertnih sistemov ter opis izgradnje in delovanja prototipa ekspertnega sistema.

V prvem delu je obravnavana vsa potrebna teorija ekspertnih sistemov od njihove strukture in karakteristik do uporabe. Opredelili bomo osrednji pojem zaključne naloge, ki je ekspertni sistem. Področje je s strani več avtorjev opisano iz različnih zornih kotov, uporabljeni so različni izrazi itd, kar bomo povzeli in strnili v širšo smiselno celoto.

Drugi del obravnava postopek zasnove prototipa od začetnih skic do konkretnega, delujočega modela. Ker modeliramo osebno ogroženost povprečnega uporabnika na internetu danes glede na uporabniško stopnjo razumevanja ter nekaj drugih spremenljivk, bo v določeni meri vključena tudi tematika računalniške varnosti. V nadaljevanju je predstavljen končen model, ki modelira nekaj realnih podatkov in prikazani ter komentirani so izhodni podatki in grafi. Poglavje se konča s celovitim pogledom na prototip in z ugotovitvami.

V zaključnem delu smo na kratko povzeli sklepe, do katerih smo prišli med izdelavo ekspertnega sistema.

## 2 TEORETIČNE OSNOVE EKSPERTNIH SISTEMOV

To poglavje vsebuje širok uvod v ekspertne sisteme. Predstavljena so temeljna načela ekspertnih sistemov. Na ustreznih področjih uporabe so navedene in opisane tudi prednosti in slabosti ekspertnih sistemov.

### 2.1 Definicija

Prva stvar pri reševanju kateregakoli problema je definirati problemsko področje ali **domeno** za reševanje. Tak način obravnave je enak v umetni inteligenci (UI) kot tudi v konvencionalnem programiranju. To omenjamo zato, ker je področje umetne inteligence pogosto povezano z visoko stopnjo zapletenosti, programerji pogosto uporabljajo rek »*To je UI problem, ki še ni bil rešen*«. Drugi priljubljen rek je »*UI povzročča, da se računalniki vedejo kot v filmih*«. To je bilo v 70. letih 20. stoletja, ko je bila umetna inteligenca še v povojih. Danes pa se že veliko problemov iz realnega sveta rešuje s pomočjo umetne inteligence in tudi veliko komercialnih aplikacij uporablja umetno inteligenco.

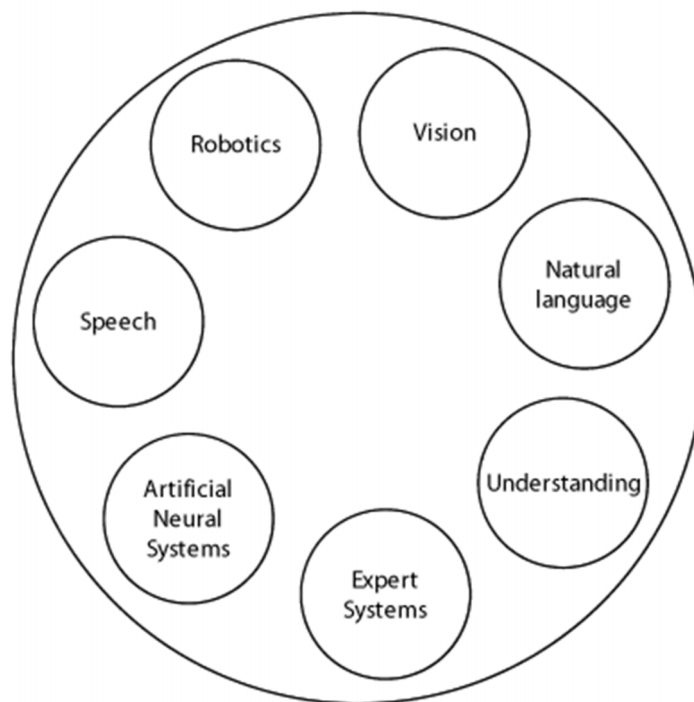
Čeprav splošne rešitve za klasične probleme s področja umetne inteligence, kot so na primer prevod naravnega jezika, razpoznavo govora in računalniški vid, še niso našli, lahko z omejitvijo problemskega področja izdelamo uporabno rešitev. Tako na primer ni težko zgraditi enostaven sistem za naravni jezik, če so vhodni podatki omejeni na stavke, ki so grajeni iz povedka, osebka in predmeta.<sup>1</sup> Trenutno so se taki sistemi izkazali za delujoče v uporabniško prijaznem vmesniku v kar nekaj izdelkih, kot so bazni sistemi ter preglednice. Pravzaprav so razpoznavalniki v popularnih pustolovskih računalniških igrah pokazali zanimivo stopnjo razumevanja naravnega jezika.

Umetna inteligenca ima več delovnih področij (Slika 1). Področje ekspertnih sistemov se zelo uspešno približuje rešitvi klasičnega problema umetne inteligence, ki je prav programiranje inteligence. Edward Feigenbaum, profesor na Stanfordski univerzi in zgodnji začetnik na področju ekspertnih sistemov, je definiral ekspertni sistem kot »inteligentni računalnik, ki uporablja znanje in postopke sklepanja za reševanje problemov, ki so dovolj zapleteni, da za rešitev zahtevajo znatno količino človeškega (strokovnega) znanja«. Ekspertni sistem je tudi računalniški sistem, ki **emulira** (angl. »emulate«) sposobnost odločanja človeškega eksperta. Izraz *emulirati* pomeni, da se ekspertni sistem v vseh vidikih vede kot človeški ekspert. Emulirati je močnejši izraz kot simulirati, ki zahteva samo posnemanje vedenja nekoga ali nečesa.

---

<sup>1</sup> Vzeto iz angleškega jezika, kjer imajo taki tipi stavkov smisel, saj nam na vzorčen način povejo, kaj nekdo počne.

Čeprav splošne rešitve še niso odkrili, ekspertni sistemi dobro delujejo v svojih omejenih domenah. Kot dokaz za njihovo uspešnost lahko izvelčemo iz veliko aplikacij ekspertnih sistemov danes v poslovnem svetu, medicini, znanost in inženiringu kot tudi v vseh knjigah, časopisih in drugih publikacijah, ki so posvečene ekspertnim sistemom.



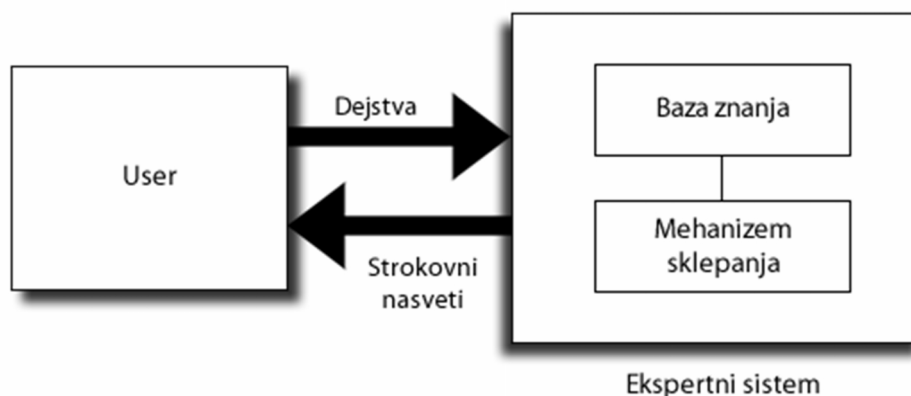
Slika 1: Nekatera področja umetne inteligence; prirejeno po Giarratano, C. J. in Riley, G. (2004)

Ekspertni sistemi so ena izmed vej v umetni inteligenci (UI), ki izrablja strokovno znanje za reševanje problemov na nivoju človeških ekspertov. Ekspert je oseba, ki ima strokovno znanje in izkušnje z določenega področja. Torej, ekspert ima znanje, ki ni na voljo večini ljudem. Ekspert zna reševati probleme, ki jih večina ljudi ne zna ali pa jih lahko reši mnogo bolj učinkovito, ampak ne po najnižji ceni. Ker so bili ekspertni sistemi prvotno oblikovani v 70. letih, so zajemali izključno ekspertno znanje. Izraz *ekspertni sistem* se danes velikokrat uporablja kar za vsak sistem, ki uporablja tehnologijo ekspertnih sistemov. Ta tehnologija lahko vključuje posebne jezike za ekspertne sisteme, programe in strojno opremo zasnovano za v pomoč pri razvoju in izvedbi ekspertnih sistemov.

Znanje v ekspertnih sistemih je lahko strokovno znanje z izkušnjami ali javno znanje, ki je dostopno v knjigah, revijah in pri izobraženih osebah. Izraza *sistem, osnovan na znanju* in *ekspertni sistem* sta enakovredna in pogosto poimenujeta isto. Večina ljudi uporablja izraz *ekspertni sistem* zaradi tega, ker je krajši, čeprav lahko v takem sistemu ni ekspertno strokovnega znanja, ampak samo splošno znanje.

Osnovni koncept ekspertnega sistema je sestavljen iz treh delov: uporabnika, mehanizma za sklepanje in baze znanja (Slika 2). Uporabnik poda dejstva ali katere druge vhodne

podatke ekspertnemu sistemu in pridobi ekspertne nasvete kot izhodne podatke. Od znotraj je ekspertni sistem sestavljen iz dveh ključnih delov: iz baze znanja, ki vsebuje znanje, in mehanizma za sklepanje, ki glede na znanje v bazi, oblikuje sklepe. Sklepi so ekspertni nasveti na predhodno uporabnikovo poizvedbo po strokovnem znanju.



Slika 2: Osnovni koncept funkcij ekspertnega sistema; prirejeno po Giarratano, C. J in Riley, G. (2004)

Razvijalci so zasnovali uporabne sisteme, osnovane na znanju kot pametne pomočnike človeškemu ekspertu. Pametni pomočniki so bili oblikovani s tehnologijo ekspertnih sistemov zaradi razvojnih prednosti. Več znanja, ko dodamo pametnemu pomočniku, bolj se vede kot ekspert. Tako je lahko grajenje pametnega pomočnika zelo koristna prelomnica za razvoj celovitega ekspertnega sistema. Poleg tega lahko časovno sprostijo eksperta s tem, ko pohitri reševanje problemov. Pametni mentor je druga oblika aplikacije umetne inteligence. V nasprotju s starimi računalniško podprti sistemi za navodila znajo novi sistemi podati kontekstno odvisna navodila [7].

Znanje eksperta je specifično za eno določeno problemsko domeno za razliko od splošnega znanja za splošne tehnike reševanja problemov. Problemska domena je posebno problemsko področje, kot so na primer medicina, finance, znanost, inženiring itd., v katerem zna ekspert zelo uspešno reševati probleme. Ekspertni sistemi so po navadi zasnovani za izvoz v eno problemsko domeno. Na primer: ni normalno pričakovati, da ima šahovski ekspert kaj ekspertnega znanja na področju medicine, ekspertno znanje v eni problemski domeni se ne samodejno prenaša v drugo.

Ekspertnemu znanju na določenem področju pravimo **domensko znanje** (angl. Knowledge Domain) eksperta. Na primer: medicinski ekspertni sistem, zasnovan za diagnostiko nalezljivih bolezni, ima veliko znanja o različnih bolezenskih znakih, ki jih povzročajo nalezljive bolezni. V tem primeru je domensko znanje medicina, ki vključuje znanje o boleznih, bolezenskih znakih in zdravljenju. Slika 3 prikazuje odnos med problemom in

domenskim znanjem. Opazite, da je domensko znanje v celoti znotraj problemske domene. Del, ki je izven obsega domenskega znanja ponazarja prostor, v katerem ni znanja o vseh problemih.

Medicinski ekspertni sistemi navadno nimajo znanja o drugih vejah medicine, recimo o kirurgiji ali pediatriji. Čeprav je znanje ekspertnega sistema o nalezljivih boleznih enakovredno človeškemu ekspertu, ekspertni sistem nima drugih domenskih znanj, če ni bil programiran za le-te.



Slika 3: Odnos problemske domene z domenskim znanjem; prirejeno iz Giarratano, C. J and Riley, G. (2004)

Ekspertni sistem v domeni znanja (področje, od koder izhaja ekspertno znanje) sklepa podobno, kot bi to počel človeški ekspert, da bi prišel do rešitve problema. To pomeni, da sistem glede na nekaj dejstev s sklepanjem pride do zaključka (končnega sklepa).

Kot pri vsaki novi tehnologiji se moramo tudi v zvezi z ekspertnimi sistemi še veliko naučiti. V Tabeli 1 so povzeti pogledi različnih udeležencev v tehnologiji. V tabeli je tehnolog lahko inženir ali oblikovalec programske opreme in tehnologija je lahko programska ali strojna oprema.



Tabela 1: Pogledi različnih udeležencev na IT tehnologijo; prirejeno po Giarratano, C. J. in Riley, G. (2004)

Oseba	Vprašanje
Manager	Za kaj naj to uporabljam?
Inženir	Kako to najbolje izvedem?
Raziskovalec	Kako lahko to razširim?
Uporabnik	Kako mi bo to v pomoč? Ali je vredno časa in stroškov? Kako zanesljiv je?

## 2.2 Prednosti ekspertnih sistemov

Ekspertni sistemi imajo nekaj privlačnih funkcij, te so:

- *Večja dostopnost.* Strokovno znanje in izkušnje (angl. *expertise*) so dostopni na vsaki primerni računalniški opremi. Ekspertni sistem je kot serijska proizvodnja strokovnega znanja in izkušenj v pravem pomenu besede.
- *Nižji stroški.* Strošek zagotavljanja strokovnega znanja na uporabnika je veliko manjši.
- *Omejena nevarnost.* Ekspertne sisteme lahko uporabljamo v okoljih, ki so nevarni za človeka.
- *Trajnost.* Strokovno znanje in izkušnje so trajni. Za razliko od človeških ekspertov, ki se lahko upokojijo, prenehajo z delom ali umrejo, je znanje v ekspertnem sistemu trajno.
- *Večkratno znanje.* Znanje več ekspertov je lahko programirano, da deluje sočasno in nenehno na nekem problemu ob kakršnemkoli času, podnevi ali ponoči. Raven znanja, združenega od več ekspertov lahko presega znanje enega eksperta.
- *Večja zanesljivost.* Ekspertni sistemi povečajo samozavest odločitve s tem, da predlagajo drugo mnenje človeškemu ekspertu ali prelomijo neodločenost v primerih, ko si človeški eksperti nasprotujejo. Taka metoda sicer ne bo delovala, če je ekspertni sistem programiral eden od ekspertov. Ekspertni sistem se mora vedno strinjati z ekspertom, v nasprotnem primeru je najverjetneje napako naredil ekspert. To se lahko zgodi, če je človeški ekspert utrujen ali pod stresom.
- *Pojasnjevanje.* Ekspertni sistem zna eksplicitno in v detajlih pojasniti, katero sklepanje je privedlo do določene rešitve. Človek je lahko preveč utrujen, nevoljen, ali nezmožen vseskozi pojasnjevati. Smiselna razlaga poveča samozavest, da je to bila pravilna odločitev.
- *Hiter odziv.* Hiter odziv ali odgovor v realnem času je včasih nujen za nekatere aplikacije. Odvisno od uporabljenega strojne in programske opreme je lahko ekspertni sistem bolj ali manj na voljo človeškemu ekspertu. Nujni primeri

zahtevajo kratek odzivni čas, krajši od človeškega, v takih primerih je ekspertni sistem v realnem času dobra izbira.

- *Stabilen, čustveno neodvisen in popolnoma odziven ves čas.* To je zlasti pomembno v izrednih primerih, kadar človeški ekspert ne more delovati z najvišjo učinkovitostjo zaradi stresa ali izčrpanosti.
- *Pametni tutor.* Ekspertni sistem se lahko vede kot pametni tutor, tako da študentu dovoli pognati nekaj vzorčnih programov, ob tem mu zna razložiti smer sklepanja.
- *Pametna podatkovna baza.* Ekspertni sistem se lahko uporabi za dostop do podatkovne baze na pameten način.

Proces razvijanja ekspertnih sistemov ima eno posredno korist – znanje ekspertov mora biti podano v eksplicitni obliki za vnos v računalnik. Prav zato, ker je znanje v eksplicitni obliki, namesto da se samo v implicitni obliki nahaja v ekspertovem umu, je tako možno preučiti pravilnost, doslednost in popolnost tega znanja. Znanje je sicer nato lahko potrebno spremeniti ali ponovno preučiti, ampak prav ta proces izboljša kakovost znanja samega.

### 2.3 Splošni pojmi v ekspertnih sistemih

Znanje v ekspertnem sistemu je lahko zastopano na več načinov, in sicer je lahko grajeno s pravili in objekti. Primer splošnega načina za predstavitev znanja je v obliki IF-THEN **pravil**, kot kaže spodnji primer.

IF gori rdeča luč THEN ustavi

Če je res, da na semaforju gori rdeča luč, potem se to ujema z vzorcem »gori rdeča luč«. Vstopni pogoj je izpolnjen in pravilo izvede »ustavljanje«. Četudi je ta primer zelo enostaven, obstaja veliko pomembnih ekspertnih sistemov, katerih znanje je izraženo v pravilih. Dejansko so z na znanju osnovanim pristopom za razvoj ekspertnih sistemov izpodrinili prejšnje pristope umetne inteligence iz 50. in 60. let, ki so skušali uporabljati zapletene metode sklepanja z malo ali nič zanašanjem na znanje. Nekateri tipi ekspertnih sistemov, kot je na primer CLIPS, omogočajo **objekte** kot tudi pravila. Pravila se vzorčno ujemajo z objekti kot tudi dejstvi. Druga možnost so objekti, ki lahko delujejo samostojno – neodvisno od pravil.

Do danes je bila zgrajena široka paleta ekspertnih sistemov, ki temeljijo na znanju. Na eni strani vsebujejo veliki sistemi na tisoče pravil, kot je na primer sistem XCON/R1 (Digital Equipment Corporation), in tako vedo mnogo več kot posamezni človeški ekspert na področju nastavitve računalniških sistemov. Po drugi strani pa obstaja veliko manjših sistemov za specializirana opravila, ki vsebujejo po več sto pravil. Ti manjši sistemi morda ne delujejo na nivoju ekspertov, saj so bili oblikovani, da bi izkoriščali prednosti

tehnologije ekspertnih sistemov za izvajanje na znanju zahtevnih nalog. Znanje za te manjše sisteme je povzeto iz knjig, revij in drugih javno dostopnih publikacij.

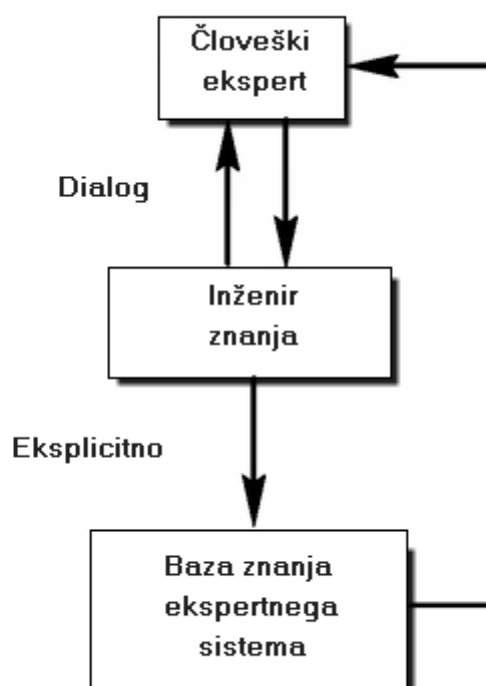
V nasprotju pa običajni ekspertni sistem posebej nenapisano znanje, ki ga mora pridobiti **inženir za znanje** (angl. *knowledge engineer*) od eksperta z dolgotrajnimi intervjuji v daljšem časovnem obdobju. Proces izgradnje ekspertnega sistema se imenuje **inženiring na znanju** (angl. *knowledge engineering*) in ga izvaja inženir za znanje. Inženiring na znanju se nanaša na prevzem znanja od človeškega eksperta na drug vir in programiranje le-tega v ekspertni sistem.

Glavne faze v razvoju ekspertnih sistemov so prikazane v Sliki 4. Inženir za znanje najprej vzpostavi dialog s človeškim ekspertom, da bi pridobil ekspertno znanje. Faza je analogna fazi, v kateri sistemski oblikovalec v klasičnem programiranju s stranko, za katero bo program zgrajen, razpravlja o sistemskih specifikacijah. Nato inženir za znanje eksplicitno zakodira znanje v bazo znanja. Potem ekspert ovrednoti ekspertni sistem in poda kritike inženirju za znanje. Ta proces se ponavlja, dokler sistemsko delovanje ni zadovoljivo po mnenju eksperta.

Izraz **sistem, osnovan na znanju**, (angl. *knowledge-based system*) je boljši za imenovanje aplikacij na znanju osnovane tehnologije, ki se lahko uporabi v izgradnji ekspertnega sistema ali na znanju osnovanih sistemov. Čeprav je današnja praksa taka, da se uporablja izraz ekspertni sistem, kadar se nanašamo na tako ekspertne sisteme kot na znanju osnovane sisteme tudi takrat, ko znanje ni na nivoju človeškega eksperta.

Navadno so ekspertni sistemi oblikovani na drugačen način kot običajni programi predvsem zato, ker nimajo enostavne algoritmične rešitve in se ti zanašajo na sklepanje, da bi dosegli smiselno rešitev. Vedeti je treba, da je smiselna rešitev največ, kar lahko v takih primerih pričakujemo, saj ni nikakršnega uporabnega algoritma, da bi nam lahko pomagal priti do optimalne rešitve. Ker se ekspertni sistem zanaša na sklepanje, mora biti sposoben razložiti svoje sklepe, tako da se jih lahko preveri. **Razlagalni modul** (angl. *explanation facility*) je eden od sestavnih delov ekspertnih sistemov. Pravzaprav bolj dodelani razlagalni moduli so lahko oblikovani tako, da omogočajo uporabniku, da razišče več možnih »kaj če« (angl. »*what if*«) vprašanj – **hipotetično sklepanje** (angl. *hypotetical reasoning*) – in celo, da prevede naravni jezik v pravila.

Naprednejši ekspertni sistemi celo omogočajo, da se sistem uči pravil iz primerov preko **indukcije pravil** (angl. *rule induction*), v kateri sistem ustvari pravila iz podatkovne tabele. Formalizacija ekspertnega znanja v pravila ni enostavna, še zlasti takrat, ko ekspertno področje še ni bilo sistematično raziskano. Pride lahko namreč do neskladij, dvoumnosti, podvajanj ali drugih problemov z ekspertnim znanjem, ki niso očitni, dokler ne poskusimo tega znanja formalno opredeliti za uporabo v ekspertnem sistemu.



Slika 4: Razvoj ekspertnega sistema; prirejeno po Giarratano, C. J. in Riley, G. (2004)

Človeški eksperti znajo podati odgovor glede na svoje **meje nevednosti** (angl. *limits of ignorance*). Z drugimi besedami to pomeni, da se človeški ekspert zaveda, ali je rešitev problema (ali samo del) izven njegovih kompetenc, in to upošteva v smislu, da ne more podati zelo zanesljive rešitve. Zna tudi kdaj *ne* upoštevati pravil. Ekspertni sistemi pa bodo, če niso posebej programirani za obvladovanje negotovosti, priporočali določene rešitve z enako samozavestjo tudi, če so vhodni podatki napačni ali nepopolni. Nasveti ekspertnih sistemov kot tudi človeških ekspertov bi morali biti smiselno manj samozavestni ob mejah nevednosti.

Praktična omejitev ekspertnih sistemov danes je pomanjkanje **vzročnega znanja** (angl. *causal knowledge*). To pomeni, da ekspertni sistemi dejansko ne razumejo temeljnih vzrokov in posledic v sistemu. Veliko lažje je sprogramirati ekspertni sistem s **plitvim znanjem** (angl. *shallow knowledge*), ki bazira na empiričnih in hevrističnih spoznanjih, kot pa z **globljim znanjem** (angl. *deep knowledge*), ki bazira na osnovnih strukturah, funkcijah in na vedenju objektov. Veliko bolj enostavno je na primer sprogramirati ekspertni sistem za predpisovati aspirin ljudem, ki jih boli glava, kot pa sprogramirati vso biokemično, psihološko, anatomsko in nevrološko znanje o človeškem telesu. Programiranje vzročnega modela o človeškem telesu bi bila neprimerljivo večja naloga in poleg tega bi bili odzivni časi po vsej verjetnosti zelo dolgi zaradi ogromne količine informacij, ki bi jih sistem moral analizirati.

Ena vrsta plitvega znanja je **hevristično znanje** (angl. *heuristic knowledge*; hevristično v grščini pomeni »odkrit«). Hevristika, sicer ne zanesljivo, vedno pride do zaključka, kot se to vedno zgodi pri algoritmih. Hevristika označuje skupek empiričnih pravil, ki v splošnem veljajo. Ta pravila so bila pridobljena iz izkustva, ki lahko pripomore k rešitvi, a ni zagotovljeno, da bo delovalo. Kljub temu pa na več področjih, kot sta na primer medicina in inženiring, hevristika igra glavno vlogo v nekaterih vrstah reševanja problemov. Četudi je natančna rešitev poznana, jo je lahko nepraktično uporabiti zaradi večjih stroškov in časovnih rokov. Hevristika lahko prinese pomembne prihranke tako v času kot v stroških.

Drugi problem pri ekspertnih sistemih je, da je njihovo ekspertno znanje omejeno na domensko znanje, za katero je bil ekspertni sistem zgrajen. Tipičen ekspertni sistem ne zna posplošiti svojega znanja na tak način, da bi lahko z uporabo **analogije** znal sklepati o podobnih problemih, kot so to zmožni eksperti. Sposobnost indukcije pravil tukaj koristi, a je na način možno pridobiti samo omejene vrste znanj. Običajen način izgradnje ekspertnih sistemov, ko inženir za znanje izvaja faze intervjuvanja, zasnove prototipa, testiranje tega in ponovno intervjuvanja itd., je časovno potratna in zelo zahtevna naloga. V bistvu je problem prenosa *človeškega znanja* v ekspertni sistem tako velik, da ima svoje ime – **ozko grlo pridobivanja znanja** (angl. *knowledge acquisition bottleneck*). Izraz je opisne narave, ker pridobivanje znanja omejuje izgradnjo ekspertnega sistema, kot ozko grlo omejuje pretakanje tekočine v steklenico.

Ne glede na omejitve so ekspertni sistemi precej uspešni pri reševanju problemov realnega sveta, katerih običajno (algoritemsko) programiranje ni sposobno reševati, še posebej take, ki se ukvarjajo z nepopolnimi ali negotovimi podatki. Pri vsem tem se je pomembno zavedati prednosti in slabosti te tehnologije in potemtakem jo ustrezno izkoristiti.

## 2.4 Karakteristike ekspertnih sistemov

Ekspertni sistemi so po navadi razviti s sledečimi značilnostmi:

- *Visoka zmogljivost.* Sistem mora biti sposoben odgovarjati na enakem ali višjem nivoju pristojnosti kot ekspert sam. To pomeni, da mora biti odgovor, ki ga da ekspertni sistem, visoko kvaliteten. Na primer: ekspertni sistem za statiko v gradbeništvu si ne more privoščiti nikakršnih napak v izračunih, številke morajo biti točne za več decimalnih mest, saj v nasprotnem primeru obstaja tveganje, da zgradba ne bo trdna, kot bi morala biti. Podobno v medicini, zanesljivost diagnoze mora biti zelo visoka, saj gre za človeška življenja.
- *Ustrezen odzivni čas.* Sistem se mora odzivati v razumnem času, ki je primerljiv ali boljši od odzivnega časa eksperta. Primerjava ekspertnega sistema, ki potrebuje teden časa, ekspert pa eno uro, pojasni, zakaj je tak ekspertni sistem neuporaben. Časovne omejitve so posebej pomembne, če je ekspertni sistem v

uporabi kot sistem v realnem času, kadar mora biti odgovor dan v določenem časovnem intervalu. Sistemi v realnem času v letalih ne bi bili uporabni, če bi se odzivali z zamikom, pa čeprav samo nekaj sekund.

- *Dobra zanesljivost.* Ekspertni sistem mora biti zanesljiv in nedovzeten za sesutja, saj v nasprotnem primeru ne bo uporaben. Nesmiseln je ekspertni sistem za pomoč pri učenju, če se vsake pol ure zaustavi in ga je potrebno ponovno zagnati. Sistem mora tudi znati delovati z nestandardnimi in nepričakovanimi podatki, v takem primeru naj sistem enostavno uporabnika opozori, da preveri in popravi vnesene podatke.
- *Razumljiv.* Sistem mora znati razložiti korake sklepanja med samim izvajanjem, da je razumljiv. Biti mora več kot samo »črna škatla«, ki proizvede čudežen odgovor, mora imeti možnost razlaganja na enak način, kot eksperti lahko razložijo njihovo sklepanje. Ta lastnost je posebej pomembna zaradi več razlogov. Na tem področju lahko ekspertni sistem pridobi nekaj prednosti pred človeškim ekspertom, ker sistem se nikoli ne utruje razlagati iste stvari. Dober razlagalni modul z veriženjem ponazori točno določeno pot sklepanja – katero dejstvo vpliva na katero odločitev in katere odločitve so sistem pripeljale do končnega sklepa. Ta lastnost je pomembna iz več razlogov.

Eden od razlogov je ta, da je lahko človeško življenje odvisno od odgovora ekspertnega sistema. Zaradi velike potencialne škode, mora ekspertni sistem biti zmožen razložiti določen zaključek. Na tak način razlagalni objekt zagotavlja razumljivo preverjanje razlogov za človeka (uporabnika).

Drugi razlog za potrebo razlagalnega modela se pojavi v razvojni fazi ekspertnega sistema, da se potrdi pravilno pridobitev in pravilno uporabo znanja v sistemu. To je pomembno pri razhroščevanju, ker je znanje lahko bilo vneseno nepravilno, torej primer tipkarske napake ali kot posledica nesporazumov med inženirjem za znanje in ekspertom. Dober razlagalni modul dovoljuje ekspertu in inženirju za znanje preverbo pravilnosti znanja.

Možno je spregledati še en vir napak, ki ga lahko odkrijemo tako, da ekspertni sistem poženemo za nekaj testnih primerov z znanim sklepanjem, s čimer preverimo njegovo delovanje. Tok izvajanja programa ni zaporeden, zato ne moremo samo pogledati kode tako, da beremo vrstico za vrstico in razumeti sklepanje. Na primer, ki ga preverjamo, lahko vpliva več pravil hkrati, ki tudi niso bila nujno vnesena v svojem vrstnem redu. Ekspertni sistem se obnaša podobno kot paralelni program, v katerem se izvaja vsako pravilo na svojem neodvisnem procesorju.

- *Fleksibilnost.* Zaradi velike količine znanja, ki ga ekspertni sistem lahko ima, je pomembno, da ima vgrajen enostaven in učinkovit modul za dodajanje, spreminjanje ter brisanje znanja. Tako so sistemi, osnovani na pravilih popularni zaradi učinkovitega modula za obdelavo pravil (znanja). Kvaliteta ekspertnega sistema je odvisna od kvalitete znanja v njegovi bazi. Proces pridobivanja znanja olajša asimilacijo znanja in ekspertiz. Lažje kot je vzdrževalcem omogočeno dodajanje in organiziranje novega ekspertnega znanja in izkušenj boljše se bo sistem izkazal pri reševanju problemov, ko se ti bodo pojavili.

Ekspertni sistem ima lahko razlagalni modul, ki je enostaven ali visoko dodelan, odvisno od sistema. Enostaven razlagalni modul, na primer, poda vsa dejstva, ki so bila uporabljena v tem koraku sklepanja. Dodelani razlagalni moduli pa lahko naredijo tudi slednje:

- *Podajo vse razloge za in proti določeni hipotezi.* Hipoteza je cilj, katerega skušamo dokazati, na primer, »pacient ima infekcijo tetanusa« v medicinskem ekspertnem sistemu za diagnostiko. V realni situaciji imamo lahko več hipotez, tako kot ima lahko pacient več bolezni hkrati. Na hipotezo lahko gledamo kot na dejstvo z določeno količino dvoma, katero skušamo dokazati.
- *Poda vse hipoteze, ki lahko razložijo opaženo stanje.* Primer je enak kot zgoraj, več hipotez ponazarja več potencialnih bolezni, ki jih pacient lahko ima. Sistem mora upoštevati vse bolezni, katerim bi lahko pripadali simptomi, zraven tega da tudi predvideva, da je lahko kakšen simptom napačen ali podoben kakšnemu drugemu. To je precej enostavno pri človeškem sklepanju, v računalniškem svetu pa je potrebna precej bolj dodelana logika, ki ni samo preveri vse bolezni, ki imajo te simptome, ampak preveri tudi vse bolezni, katere imajo vse simptome razen enega, vse simptome razen dveh in tako dalje do neke smiselne meje.
- *Razloži vse možne posledice neke hipoteze.* Denimo, da ima pacient tetanus, torej mora obstajati tudi znak povišane telesne temperature vzdolž trajanja bolezenskega stanja. Če je bil opažen tudi ta znak bolezni, doda svojo težo na stran pravilnosti hipoteze. Obratno odsotnost znaka bolezni zmanjša kredibilnost hipoteze. Bolj splošno bi to pomenilo da za vse potencialne bolezni, ki smo jih izračunali v predhodnem koraku dodelimo težo na stran verjetnosti bolezni, katera znak ima in odvzamemo ali postavimo na negativno stran tisti bolezni za katero znak ni primarno značilen.
- *Poda **napoved**, v primeru, ko je hipoteza pravilna.* Za hipotezo, ki se izkaže v največji meri poda detajlno razlago. Medicinski ekspertni sistem za diagnostiko poda vse simptome, ki bili za izkazano bolezen podani in so zanjo značilni,

morebiti če je kakšen od teh, ki zanjo ni najbolj značilen, kateri simptomi manjkajo, itd.

- *Upraviči vprašanja za uporabnika, ki jih postavi zaradi potrebe po več informacijah.* Ta vprašanja se uporabijo za usmerjanje sklepanja k bolj verjetnostnim potem. Pri večini realnih problemov je predrago (ali časovno potratno) preveriti vse možne poti, zato je potrebno sistem do določene mere voditi k smiselnemu sklepu. Predstavljajte si, na primer, ceno, čas in uporabo rezultata raziskovanja vseh možnih boleznih pacientu, ki se potoži za vnetega grla. Zato v takih primerih s pomočjo pomožnih vprašanj, ki jih zastavimo pacientu (uporabniku), po krajši in računsko manj zahtevni poti pridemo do želenih rezultatov. Pacientu z vnetim grlom sistem postavi vprašanje kakšen tip bolečine (možen odgovor: »pekoč«), kdaj ga boli grlo (»ob požiranju«), tako pride sistem precej hitreje do sklepa o resnosti pacientovih težav.
- *Razloži svoje sklepanje oziroma znanje.* Kot primer vzemimo hipotezo, da ima pacient tetanus, ki se izkaže kot pravilna. Uporabnik želi dokaz. Sistem upraviči pravilnost na bazi pravila, ki pravi, da če ima pacient pozitiven test krvi na tetanus, potem ima tetanus. Sedaj uporabnik želi dokaz tudi tega pravila. Sistem odgovori, da pozitiven test krvi na bolezen je gotov dokaz, da ima to bolezen.

V zadnjem primeru sistem razloži **metapravilo**, ki predstavlja znanje o pravilih samih.

## 2.5 Razvoj ekspertnih sistemov

Umetna inteligenca ima veliko vej, ki se ukvarjajo z računalniškim govorom, vidom, robotiko, razpoznavo in učenjem naravnega jezika ter ekspertnimi sistemi. Korenine ekspertnih sistemov ležijo v različnih disciplinah. Ena glavnih vej tehnologije ekspertnih sistemov izhaja prvenstveno s področja obdelave informacij, imenovane **kognitivna znanost** [10]. *Kognicija* je študija o tem, kako ljudje obdelujejo informacije. Z drugimi besedami povedano je to znanost, ki se ukvarja, kako ljudje razmišljajo, zlasti takrat ko rešujejo probleme.

Študija kognicije je posebej pomembna, če hočemo, da se sistemi vedejo kot človeški eksperti. Pogosto ekspertni sami ne znajo razložiti, kako so prišli do določene rešitve. Če ekspert ne zna razložiti, kako je do zaključka prišel, ni mogoče to zakodirati v znanje ekspertnega sistema, ki bazira na eksplicitnem znanju. V takem primeru je edina rešitev sistem, ki se sam uči oponašati eksperta. Taki sistemi so bazirani na indukciji in nevronskih mrežah, kar pa presega meje te zaključne naloge.



### 2.5.1 Človeško reševanje problemov in produkcije

Razvoj ekspertnih sistemov pritegne k sebi široko ozadje. V Tabeli 2 so na kratko povzete prelomnice v zgodovini razvoja ekspertnih sistemov, ki so aplicirane v modernih ekspertnih sistemih. Podane so začetne letnice projektov, kjer je to mogoče, saj se je veliko projektov odvijalo več let. Zelo obsežno so zgodnji sistemi prikazani v tridelni zbirki *Handbook of Artificial Intelligence*.

Tabela 2: Pregled zgodovine razvoja ekspertnih sistemov; prirejeno iz Giarratano, C. J and Riley, G. (2004)

Leto	Dogodek
1943	Postprodukcijaska pravila; McCulloch and Pitts Neuron Model
1954	Markov algoritem za nadzor nad tokom izvajanja pravil
1956	Konferenca v Darmouthu; teorija logike; hevristično iskanje; skovan izraz »UI«
1957	Izumljena nevronska mreža Perceptron (Rosenblatt); splošni reševalec problemov (angl. GPS – general problem solver; Newell, Shaw in Simon)
1958	LISP UI jezik (McCarthy)
1962	Rosenblattovi principi neurodinamike o zaznavanju
1965	Rešitvena metoda o avtomatičnem dokazovanju izrekov (Robinson) Mehka logika in mehko sklepanje (Zadeh) DENDRAL, prvi ekspertni sistem (Feigenbaum, Buchanan et al.)
1968	Semantične mreže, model asociativnega pomnilnika (Quillian)
1969	MACSYMA matematični ekspertni sistem (Martin in Moses)
1970	Začne se projekt PROLOG (Colmerauer, Roussel et al.)
1971	HEARSAY 1 za razpoznavo govora <i>Človeško reševanje problemov</i> popularizira pravila (Newell in Simon)
1973	Ekspertni sistem MYCIN za medicinsko diagnosticiranje (Shortliffe et al.), kar vodi do GUIDON, pametno tutorstvo (Clancey), TEIRESIAS, koncept pojasnjevalnega objekta (Davis) in EMYCIN, prva lupina (Van Melle, Shortliffe in Buchanan) HEARSAY 2, tabla modela več sodelujočih ekspertov
1975	Okvirji, predstavitev znanja (Minsky)
1976	UM – Umetni matematik (angl. Artificial Mathematician) ustvarjalno odkritje matematičnih konceptov (Lenat) Dempster-Shafterjeva teorija dokazov za sklepanje v negotovosti Pričetek dela na ekspertnemu sistemu PROSPECTOR za raziskovanje rudnin (Duda, Hart et al.)
1977	Lupina za ekspertne sisteme OPS (Forgy), vgrajena v XCON/R1

1978	Pričetek dela na XCON/R1 (McDermott, DEC) za nastavitve DEC računalniških sistemov Meta-DENDRAL, metappravila in indukcija pravil (Buchanan)
1979	Rete algoritem za hitro iskanje vzorcev (Forgy) UI se komercializira Nastalo Inference Corp. Podjetje (leta 1985 izda ART – orodje za razvoj ekspertnih sistemov)
1980	Ustanovljen LMI za izdelavo LISP naprav, podjetje Symbolics
1982	Matematični ekspertni sistem SMP; Nevronska mreža Hopfield
1983	Orodje za razvoj ekspertnih sistemov KEE (InteliCorp)
1985	Orodje za razvoj ekspertnih sistemov CLIPS (NASA)

V drugi polovici 50. let in v prvi polovici 60. let je bilo napisanih precej programov za splošno reševanje problemov. Med temi je najbolj znan Splošni reševalec problemov (angl. General Problem Solver), ki sta ga napisala Newell in Simon. Opisala sta ga v seriji člankov, ki se konča z monumentalnim 920 strani dolgim delom o kogniciji z naslovom *Človeško reševanje problemov* (angl. *Human Problem Solving*).

Eden najbolj pomembnih spoznanj, ki sta jih dokazala Newell in Simon, je bilo, da je velik del človeškega reševanja problemov ali **kognicije** možno izraziti z IF-THEN tipi **produkcijskimi pravili**. Če (IF) na primer kaže, da bo deževalo, potem (THEN) vzemi dežnik, ali če (IF) je tvoj zakonski partner slabe volje, potem (THEN) ne bodi vesel. Pravilo ustreza majhnemu, modularnemu skupku, ki se imenuje **košček** (angl. *chunk*). Koščki znanja so ohlapno urejeni z drugimi povezanimi koščki. Obstaja teorija, ki razlaga, da je celotni človeški spomin urejen v koščke. Spodaj je naveden primer pravila, ki ponazarja košček znanja.

IF avtomobil ne vžge in

merilnik goriva kaže prazno

THEN napolni rezervoar za gorivo

Newell in Simon sta odgovorna za množično uporabo pravil za predstavitev človeškega znanja in pokazala, sta kako poteka sklepanje z pravili. Kognitivni psihologi uporabljajo pravila kot modele za razlago človeške obdelave podatkov [15]. Osnovna ideja je, da senzorični vhodni podatki proizvajajo dražljaje v možganih. Dražljaji sprožijo ustrezna pravila v dolgoročnem spominu, kar proizvede primeren odziv. Naše znanje je shranjeno v dolgoročnem spominu. Verjetno vsi sklepamo, kot je prikazano v spodnjih pravilih.

IF opazil plamen THEN je tam požar

IF opazil dim THEN obstaja možnost za požar

IF slišim zvok sirene THEN obstaja možnost za požar

Zadnji dve pravili nista izraženi s popolno gotovostjo, saj je lahko ogenj že pogašen, pa je dim še vedno v zraku. Podobno, sirena ni dokaz za požar, saj je bil lahko alarm samo lažni. Dražljaji, ki se sprožijo ob tem, ko vidimo plamen, zavohamo dim in slišimo sireno, sprožijo taka in podobna pravila.

Dolgoročni spomin je sestavljen iz veliko pravil z IF-THEN strukturo. Šahovski mojstri naj bi namreč poznali 50 000 ali več koščkov znanja o šahovskih potezah. V nasprotju z dolgoročnim spominom se **kratkoročni spomin** uporablja za kratkotrajno shranjevanje znanja v času reševanja problemov. Čeprav se lahko v dolgoročni spomin shrani na stotine tisoč ali več koščkov, je kapaciteta kratkoročnega spomina presenetljivo majhna – štiri od **sedem** koščkov.

Drugi element, ki je prav tako potreben za človeško reševanje problemov, je **kognitivni procesor**. Kognitivni procesor skuša najti pravila, ki bodo **aktivirana** s posebnimi dražljaji, a ne pride poštev katerokoli pravilo. Na primer: človek ne bi hotel napolniti rezervoar za gorivo vsakič, ko sliši sireno. Aktivirano bo samo pravilo, ki se ujema. Če se aktivira več pravil hkrati, mora kognitivni procesor izvesti reševanje sporov, da izbere pravilo z višjo prioriteto. To pravilo bo izvedeno, na primer, če sta sledeči pravili aktivirani:

IF je tam požar THEN zapusti prostor

IF moja oblačila gorijo THEN pogasi ogenj

Potem izvedemo dejanja enega pravila – z višjo prioriteto – preden izvedemo drugo pravilo. **Mehanizem za sklepanje** v modernih ekspertnih sistemih ustreza kognitivnemu procesorju.

Newellov in Simonov model človeškega reševanja problemov je v smislu dolgoročnega spomina (pravila), kratkoročnega spomina (delovni pomnilnik) in kognitivnega procesorja (mehanizem za sklepanje) začetek modernega ekspertnega sistema, ki temelji na pravilih.

Taka pravila so en tip **produkcijskega sistema**. Danes so produkcijski sistemi, osnovani na pravilih, popularna način za izvedbo ekspertnih sistemov. Posamezna pravila, vsebovana v produkcijskem sistemu, se imenujejo **produkcijska pravila**. V fazi oblikovanja ekspertnega sistema je pomemben dejavnik količina znanja ali **razdrobljenost** (angl. *granularity*) pravil. Premajhna razdrobljenost oteži razumevanje določenega pravila glede na ostala. Prevelika razdrobljenost pa povzroči, da je ekspertni sistem težek za spreminjanje, saj je kar nekaj koščkov znanja vgrajenih v enem pravilu.

Do polovice 60. let so si veliko prizadevali za proizvodnjo inteligentnih sistemov, ki se niso veliko nanašali na domeno znanja, ampak bolj na zmožljive metode sklepanja. Celotno ime Splošni reševalec problemov ponazarja usmeritev na naprave, ki niso bile oblikovane za eno samo specifično domeno, ampak so bile namenjene za reševanje več tipov problemov. Čeprav so bile metode za sklepanje v splošnih reševalcih problemov dokaj zmožljive, so bile te naprave večni začetniki. Ob prehodu na novo domeno so morali ponovno odkriti vse od začetka in se niso odrezali tako dobro kot človeški eksperti, ki so se nanašali na domeno znanja za višjo zmožljivost.

Primer moči znanja je igra šah. Čeprav računalniki tekmujejo proti ljudem, se ljudje pogosto odrežejo zelo dobro kljub dejstvu, da računalniki lahko izračunavajo milijonkrat hitreje. Raziskave so pokazale, da šahovski eksperti nimajo posebnih super moči za sklepanje, ampak se namesto tega zanašajo na vzorce šahovskih potez, zbrane v preteklih letih igranja. Kot je bilo že navedeno, imajo lahko šahovski eksperti v znanju shranjenih 50 000 in več vzorcev. Ljudje smo zelo dobri v prepoznavanju vzorcev, na primer v različnih postavitvah šahovskih figur na mizi. Namesto da bi predvidevali 10 ali 20 možnih korakov vnaprej, kot bi to počel računalnik, človek analizira igro v smislu vzorcev, ki razkrivajo dolgoročne grožnje, medtem ko ostane pozoren na kratkoročne presenetljive poteze.

Čeprav je lahko domensko znanje zelo močno, je tudi na splošno omejeno na domeno. Na primer, oseba, ki postane ekspertni igralec šaha, ne postane samodejno tudi ekspert na področju reševanja matematičnih problemov in niti ekspert v dami. Medtem ko je lahko nekaj znanja prenosljivega na drugo domeno kot pazljivo planiranje potez, je to v bistvu bolj večščina kot strokovno znanje.

Do začetka 70. let je postalo jasno, da je domensko znanje ključen pristop za izgradnjo strojnih reševalcev problemov, ki bi lahko delovali na nivoju človeških ekspertov. Četudi so metode sklepanja pomemben faktor, so raziskave pokazale, da se eksperti pri reševanju problemov v prvi vrsti ne zanašajo na sklepanje. Pravzaprav igra sklepanje manjšo vlogo pri ekspertnem reševanju problemov. Namesto tega se eksperti opirajo na obsežno količino poznavanja hevrstike in na izkušnje, ki so jih pridobili z časom. Če ekspert ne zna rešiti problema s svojim strokovnim znanjem in izkušnjami, potem mora sklepati iz principov in teorije (ali se, bolj verjetno, posvetuje z drugim ekspertom). Zmožnost sklepanja eksperta v splošnem ni nič boljša od povprečnega človeka v nepoznani situaciji. Zgodnji poskusi pri izgradnji sposobnih reševalcih problemov, osnovanih izključno na sklepanju, so pokazali, da so taki sistemi dokaj ohromljeni.

Uvid, da je domensko znanje ključ pri izgradnji reševalcev problemov v realnem svetu, je pripeljal ekspertne sisteme do splošnega uspeha. Uspešni ekspertni sistemi danes so, torej, ekspertni sistemi osnovani na znanju in ne toliko splošni reševalci problemov. Ob tem je ista tehnologija, ki je omogočila razvoj ekspertnih sistemov, omogočila tudi razvoj

sistemov, osnovanih na znanju, ki ne nujno vsebujejo strokovnega znanja in človeških izkušenj.

Medtem ko se strokovno znanje obravnava kot specializirano in je poznano samo nekaterim, je splošno znanje običajno dostopno v knjigah, revijah, člankih in drugih tipih publikacij. Na primer znanje kako se rešujejo kvadratne enačbe ali odvaja in integrira, je splošno dostopno. Programi na osnovi znanja kot MACSYMA im SMP so na voljo za samodejno izvajanje takih in drugačnih matematičnih operacij na obojih, številčnih ali simboličnih operandih. Danes se izraza *sistem, osnovan na znanju* in *ekspertni sistem* pogosto uporabljata kot *sinonima*. Še več, ekspertni sistemi so dandanes obravnavani kot alternativen programski model ali **paradigma** običajnemu algoritmičnemu programiranju.

## 2.6 Rast ekspertnih sistemov baziranih na pravilih

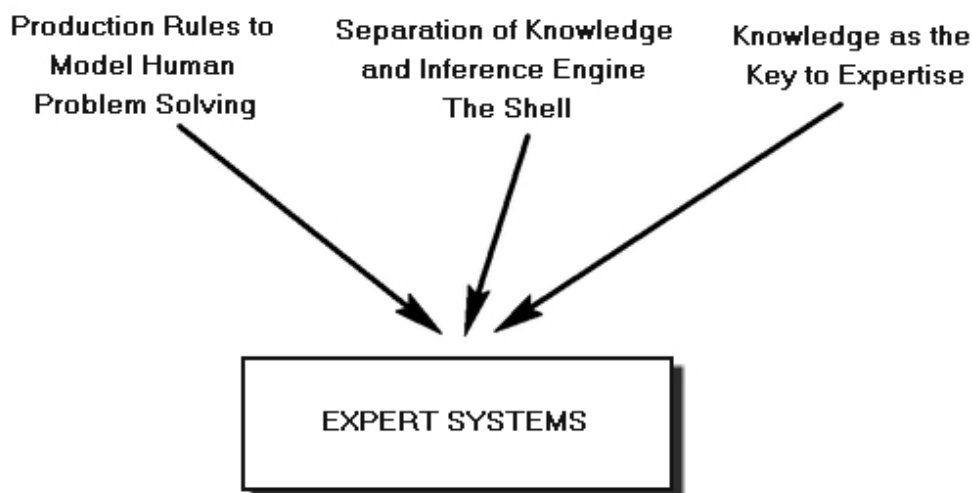
Sredi 70. let, po prevzemu paradigme modela, osnovanega na znanju, je nastalo več uspešnih prototipov ekspertnih sistemov. Ti sistemi so znali interpretirati masovne spektrograme za prepoznavanje kemijskih sestavin (DENDRAL), postaviti diagnozo bolezni (MYCIN), analizirati geološke podatke v naftnih poslih (DIPMETER) in mineralov (PROSPECTOR) ter konfigurirati računalniške sisteme (XCON/R1). Novici, da je PROSPECTOR odkril mineralno jamo v vrednosti 100 milijonov dolarjev, XCON/R1 pa podjetju Digital Equipment Corporation (DEC) prihranil na milijone dolarjev na leto, sta znatno povešali splošno zanimanje za take sisteme. Veja umetne inteligence, ki je začela razvijati v 50. letih kot študij človeškega procesiranja informacij, je sedaj dosegla komercialen uspeh z izdelavo praktičnih sistemov za uporabo v realnem svetu.

Ekspertni sistem MYCIN je bil pomemben zaradi več razlogov. Prvi je ta, da je pokazal, da je lahko umetna inteligenca uporabna tudi za praktične probleme v realnem svetu. Kot drugo je bil MYCIN poskusni zajček za nove koncepte, kot so razlagalni modul, modul za avtomatsko zajemanje znanja in modul pametnega mentorja, ki ga lahko najdemo v veliko današnjih ekspertnih sistemih. In tretji razlog, MYCIN je bil pomemben tudi zato, ker je pokazal izvedljivost lupine za ekspertne sisteme.

Predhodni ekspertni sistemi, kot je na primer DENDRAL, so bili unikatni sistemi, v katerih je bila baza znanja prepletena s programsko opremo, ki je uporabljala to znanje. MYCIN je eksplicitno ločil ta dva pojma, tako je bil mehanizem za sklepanje popolnoma neodvisen od znanja, hranjenega v bazi. To je bilo takrat ekstremnega pomena, saj je pomenilo, da je lahko jedro, ki je neodvisno od znanja, ponovno uporabljeno. To nedvomno pomeni, da je bil lahko naslednji ekspertni sistem veliko hitreje izgrajen kot sistem tipa DENDRAL – potrebno je bilo le izprazniti bazo znanja in jo zamenjati z novo problemsko domeno. Del ekspertnega sistema MYCIN, ki je bil odgovoren sklepanje in razlaganje – lupina, je bil ponovno napolnjen z znanjem novega sistema. Lupina z odstranjenim znanjem MYCIN

ekspertnega sistema se je imenovala EMCYIN bistvena (angl. *essential*) ali prazna (angl. *empty*) MYCIN lupina.

Do 70. let so bili razviti trije glavni koncepti današnjih ekspertnih sistemov, kot kaže slika spodaj (Slika 5). Ti koncepti so pravila, lupina in znanje.



Slika 5: Konvergenca treh pomembnih faktorjev pri razvoju modernih ekspertnih sistemih, osnovanih na pravilih; prirejeno po Giarratano, C. J. in d Riley, G. (2004)

Do 80. let pa so se ekspertni sistemi preselili iz raziskovalnih oddelkov na univerzah na trg. Podjetja so začela izdelovati ekspertne sisteme za komercialno rabo. Uvedena je bila nova zmogljiva programska oprema za razvoj ekspertnih sistemov, kot je na primer Orodje za samodejno sklepanje (podjetje Inference Corp.), Orodje za inženiring znanja (InteliCorp), Gospodar pravil (Radian Corp). Kot dodatek so razvili tudi novo strojno opremo, ki je poganjala omenjeno programsko opremo z višjo hitrostjo kot kadarkoli prej. Podjetja kot Symbolics in LMI so izdelala računalniške sisteme, poznane pod imenom LISP računalniki, ker so bili razviti za LISP, to je osnovni temeljni jezik za programsko opremo za razvoj ekspertnih sistemov. V LISP računalnikih je bil osnovni zbirni jezik, operacijski sistem in vsa druga temeljna koda pisana v LISP-u.

Problem visoke tehnologije pa je bil cenovni vidik. Čeprav je bil LISP računalnik za posameznega uporabnika precej bolj produktiven kot sistem za splošno uporabo, na katerem je tekel LISP, je bil LISP računalnik in licenca za programsko opremo precej drag (okrog 100.000 dolarjev). Ustanovitev UI laboratorija s šestimi programerji je lahko hitro dosegla ceno 500.000 dolarjev.

Zgoraj navedeno težavo so rešili sredi 80. let z vpeljavo zmogljivejše programske opreme, in sicer CLIPS iz NASE. CLIPS je bil napisan zaradi hitrosti in prenosljivosti v C-ju ter je

uporabljal učinkovito ujemanje vzorce, imenovano Rete algoritem. CLIPS je brezplačen za vladne uporabnike in vladne pogodbenike. Poleg tega je dostopen za ostale uporabnike v zameno za nominalno ceno podjetja COSMIC, ki distribuira vso programsko opremo, razvijajočo pri NASI; univerze jo lahko dobijo po polovični ceni. CLIPS je brezplačen in kdor pridobi zakonito kopijo od COSMIC, jo lahko prosto posreduje drugim.

CLIPS je možno namestiti na katerikoli C prevajalnik, ki podpira osnovni Kernigan in Richie C jezik. Bil je nameščen tudi na IBM-PC in podobnih, VAX, Hewlett-Packard, Sun, Cray in mnogo drugih. Obstajajo tudi različice za Macintosh in Windows pri podjetju COSMIC.

## 2.7 Aplikacije ekspertnih sistemov in njihove domene

Konvencionalne računalniške programe uporabljamo za rešitev različnih tipov problemov. Taki problemi imajo v splošnem algoritmične rešitve, ki se v svoji strukturi dobro ujemajo z običajnimi programskimi jeziki kot FORTRAN, Pascal, Java, C ipd. V veliko poslovnih in inženirskih aplikacijah so numerični izračuni primarnega pomena. V nasprotju pa so bili ekspertni sistemi prvotno oblikovani za simbolično sklepanje.

Čeprav se jezike, kot sta na primer LISP in PROLOG, uporablja tudi za simbolično obdelovanje, so v glavnem za bolj splošno rabo kot lupine za ekspertne sisteme. To sicer ne pomeni, da se v LISP-u in PROLOG-u ne da razviti ekspertnih sistemov. Več ekspertnih sistemov je bilo namreč razvitih v teh dveh jeziki, še posebej v PROLOG-u [4], saj ima ta vrsto prednosti za diagnozo sistemov zaradi vgrajenega obratnega veriženja. Ne glede na to, je bolj priročno in učinkovito razvijati večje ekspertne sisteme v lupinah in drugih pripomočkih (programih) posebej oblikovanih za razvoj ekspertnih sistemov. Namesto »odkrivanja tople vode«, vedno ko se razvija nov ekspertni sistem, je učinkoviteje uporabiti specializirana orodja, namenjena razvoju ekspertnih sistemov, kot pa orodja za splošno uporabo.

### 2.7.1 Aplikacije ekspertnih sistemov

Znanost ekspertnih sistemov je bila uporabljena v dejansko vseh strokah. Nekateri so bili razviti kot raziskovalni projekti, medtem ko drugi izpolnjujejo pomembne poslovne in industrijske naloge. Tak primer ekspertnega sistema v rutinski poslovni uporabi je XCON sistem v podjetju Digital Equipment Corp. (DEC). Sistem XCON (predhodno imenovan R1) je bil razvit v sodelovanju z John McDermott iz Carnegie-Mellon univerze. XCON je ekspertni sistem za konfiguracijo računalniških sistemov v podjetju DEC.

Konfiguracija računalniškega sistema pomeni, da kadar kupec odda naročilo za računalniški sistem, se dobavi vse prave dele – programsko opremo, strojno opremo in dokumentacijo. Za večje sisteme mora biti računalniški sistem kupca nastavljen in

konfiguriran v tovarni ter tudi testiran, da izpolnjuje vse kupčeve zahteve. Za razliko od nakupa televizije ali domačega računalnika obstaja pri večjem sistemu veliko opcij in povezav, ki jih je treba nastaviti. Za sestavo večjega računalniškega sistema ni dovolj samo zahtevati število procesorskih enot, trdih diskov in terminalov in tako dalje. Tudi ožičenje in pravilne povezave morata biti določene in potrjene, da taka konfiguracija dejansko deluje.

Sistem XCON je verjetno eden najbolj uspešnih ekspertnih sistemov v rutinski uporabi in prihrani DEC-u milijone dolarjev letno, zmanjša čas procesiranja naročil in povečuje točnost izvedbe naročil. XCON zmore konfigurirati povprečno naročilo v dveh minutah, kar je 15-krat hitreje, kot to zmore človek. Poleg tega so ljudje v konfiguraciji točni v 70 % naročil, sistem XCON beleži točnost v 98 %. To so pomembni aspekti v tovarniški konfiguraciji računalniških sistemov, a to zahteva veliko truda. Drago je napol konfigurirati sistem in potem ugotoviti, da sistem ne izpolnjuje kupčevih zahtev ali odkriti da so potrebne še druge komponente, zaradi česar bo naročilo potrebno odložiti, dokler le-te ne prispejo.

Izdelanih in zabeleženih je bilo na stotine ekspertnih sistemov v računalniških revijah, knjigah in konferencah, a to predstavlja verjetno samo vrh ledene gore, saj veliko podjetij in vojaških organizacij ne poroča o svojih sistemih in dosežkih zaradi lastniškega ali skrivnega znanja, zajetega v le-teh. Na bazi sistemov, opisanih v javno dostopni literaturi, lahko ločimo več tipov ekspertnih sistemov (Tabela 3).

Tabela 3: Različni tipi ekspertnih sistemov; prirejeno po Giarratano, C. J. in Riley, G. (2004)

Razred	Splošno področje
Konfiguracija	Pravilna namestitev komponent znotraj sistema.
Diagnoza	Predlagati sklepe osnovnih problemov glede na opazovano stanje.
Navodila	Pametno poučevanje tako, da lahko študent zastavi vprašanja tipa <i>Zakaj?</i> , <i>Kako?</i> in <i>Kaj če?</i> , kot če bi ga poučeval človek.
Interpretacija	Razlaga opazovanih podatkov oz. stanja.
Spremljanje	Primerjava opazovanih podatkov s pričakovanimi za določitev stopnje uspešnosti.
Planiranje	Pripraviti ukrepe, ki bojo prinesli želen rezultat.
Prognoza	Napoved zaključka danega stanja.
Kurativa	Predpis zdravljenja danih težav.
Upravljanje in nadzor	Regulacija procesa. Lahko zahteva interpretacijo, diagnozo, spremljanje, planiranje, prognozo in kurativo.



Nekaj primerov ekspertnih sistemov je prikazanih v Tabelah 5, 6, 7, 8 in 9 (Waterman 1986).

Tabela 4: Ekspertni sistemi v kemiji; prirejeno po Giarratano, C. J. in Riley, G. (2004)

<b>Ime</b>	<b>Kemija</b>
CRYALIS	interpretira 3-D strukturo beljakovin
DENDRAL	interpretira molekularno strukturo
TQMSTONE	odpravljanje incidentov v trojno kvadrupolnem masnem spektrometru (vzdržuje naravnost)
CLONER	načrtuje nove biološke molekule
MOLGEN	načrtuje eksperimente kloniranja genov
SECS	načrtuje kompleksne organske molekule
SPEX	planira eksperimente molekularne biologije

Tabela 5: Ekspertni sistemi v elektroniki; prirejeno po Giarratano, C. J. in Riley, G. (2004)

<b>Ime</b>	<b>Elektronika</b>
ACE	diagnoza napak v telefonskih omrežjih
IN-ATE	diagnoza napak v osciloskopih
NDS	diagnoza nacionalnih komunikacijskih mrež
EURISKO	načrtuje 3-D mikroelektroniko
PALLADIO	načrtuje in testira nove VLSI vezja
REDESIGN	prenova digitalnih vezij
CADHELP	navodila za računalniško vodeno načrtovanje
SOPHIE	navodila za diagnozo napak v vezjih

Tabela 6: Ekspertni sistemi v medicini; prirejeno po Giarratano, C. J. in Riley, G. (2004)

<b>Ime</b>	<b>Medicina</b>
PUFF	diagnoza pljučnih bolezni
VM	spremlja pacienta na intenzivni negi
ABEL	diagnoza kislno-bazičnih elektrolitov
AI/COAG	diagnoza krvnih bolezni
AI/RHEUM	diagnoza revmatičnih bolezni
CADUCEUS	diagnoza internih medicinskih bolezni
ANNA	spremlja digitalis terapijo
BLUE BOX	diagnoza/zdravi depresijo
MYCIN	diagnoza/zdravi bakterijske okužbe
ONCOCIN	zdravi/spremlja kemoterapevtske bolnike
ATTENDING	navodila za upravljanje anestezije
GUIDON	navodila za bakterijske okužbe

Tabela 7: Ekspertni sistemi v strojništvu; prirejeno po Giarratano, C. J. in Riley, G. (2004)

<b>Ime</b>	<b>Strojništvo</b>
------------	--------------------

REACTOR	diagnoza/odpravljanje incidentov v reaktorju
DELTA	diagnoza/odpravljanje napak v GE lokomotivah
STEAMER	navodilo za upravljanje parnih strojev

Tabela 8: Ekspertni sistemi v geologiji; prirejeno po Giarratano, C. J. in Riley, G. (2004)

Ime	Geologija
DIPMETER	interpretira dnevnik globinomerov
LITHO	interpretira dnevnik naftnih vrtin
MUD	diagnoza/odprava napak v izkopavanju
PROSPECTOR	intrepretira geološke podatke za rudnine

Tabela 9: Ekspertni sistemi v računalništvu; prirejeno po Giarratano, C. J. in Riley, G. (2004)

Ime	Računalniški sistemi
PTRANS	prognoza za upravljanje DEC računalnikov
BDS	diagnoza slabih delov v preklopni mreži
XCON	nastavitve DEC računalniških sistemov
XSEL	nastavitve/obdelava prodajnih naročil DEC računalnikov
XSITE	nastavitve prodajnih mest za DEC računalnike
YES/MVS	upravljanje IBM MVS operacijskih sistemov
TIMM	diagnoza DEC računalnikov

### 2.7.2 Primerne domene za ekspertne sisteme

Pred izdelavo ekspertnega sistema je bistvenega pomena določiti, ali je paradigma ekspertnih sistemov ustrezna. Vprašanje je, ali je odločitev za ekspertni sistem sploh smiselna namesto alternativne paradigme – uporaba običajnega programiranja. Domena primerna za nek ekspertni sistem je odvisna od vrste faktorjev:

- *Ali lahko problem učinkovito rešimo z običajnim programiranjem?* Če je odgovor pritrdilen, potem ekspertni sistem ni prava odločitev. Kot primer vzemimo problem diagnosticiranja računalniške opreme. Če so vsi simptomi za vse okvare znani vnaprej, potem je enostaven tablični pregled ali odločitveno drevo okvar primerno. Ekspertni sistemi so najprimernejši za primere, ko ni učinkovite algoritmične rešitve. Taki primeri so imenovani **slabo strukturirani problemi** (angl. *ill-structured problems*) in sklepanje je lahko edina možna pot do dobre rešitve.

Kot primer slabo strukturiranega problema je primer osebe, ki razmišlja o potovanju in običje potovalno agencijo. Medtem ko večina ljudi ima izbrano destinacijo in plan, obstajajo izjeme. Tabela 10 opisuje nekaj lastnosti slabo strukturiranega problema kot odgovore, ki jih je oseba podala na vprašanja delavca v potovalni agenciji. Čeprav je to

verjetno ekstremen primer, nam primer pomaga prikazati osnovne koncepte slabo strukturiranih problemov. Kot lahko opazimo, slabo strukturiran problem se ne da enostavno prevesti v algoritmično rešitev, saj je možnosti veliko. V takih primerih, ko vse drugo spodleti, bi morala obstajati nekakšna osnovna možnost, ki bi jo lahko uveljavljali. Delavec v potovalni agenciji bi lahko na primer rekel: »Aha, imam rešitev za vas – odlično potovanje: turo okrog sveta. Prosim izpolnite sledečo vlogo za kreditne kartice in vse bo poskrbljeno.«

Tabela 10: Primer slabo strukturiranega problema (vprašanja in odgovori); prirejeno po Giarratano, C. J. in Riley, G. (2004)

Vprašanja potovalne agencije	Odgovori
Vam lahko pomagam?	Nisem siguren.
Kam želite potovati?	Nekam.
Kakšna posebna destinacija?	Tu in tam.
Koliko ste zmožni zapraviti?	Ne vem.
Koliko ste pripravljeni zapraviti?	Ne vem.
Kdaj želite potovati?	Prej ali slej.

V ravnanju z slabo strukturiranimi problemi obstaja nevarnost, da ekspertni sistem nehote oponaša algoritmično rešitev. To pomeni, da nam razvoj ekspertnega sistema odkrije algoritmično rešitev. Eden od pokazateljev, da se je ravno to zgodilo, je ta, da rešitev zahteva **togo nadzorno strukturo**. To pomeni, da morajo biti pravila izvedena v določenem vrstnem redu. To eksplicitno zahteva inženir znanja in poda prioriteto večini pravil. Siljenje v togo nadzorno strukturo prekine večjo prednost tehnologije ekspertnih sistemov – ravnanje z nepričakovanimi vhodnimi podatki, ki ne spadajo v noben vnaprej opredeljen vzorec. Tako se ekspertni sistemi odzivajo oportunistično glede na vhodne podatke, kakršnikoli le-ti so. Običajni programi v splošnem zahtevajo, da vhodni podatki sledijo nekemu vzorcu. Ekspertni sistem z veliko nadzora je pogosto prikrit algoritem in dober kandidat za običajno programiranje.

- *Ali je domena dobro omejena?* Pomembno je imeti dobro definirane omejitve – kaj pričakujemo, da ekspertni sistem zna in katere so njegove zmogljivosti. Predpostavimo, da želimo izdelati ekspertni sistem za diagnozo glavobolov. Zagotovo bo v bazo znanja vključeno medicinsko znanje zdravnika. Čeprav za globlje razumevanje glavobolov se lahko vključi znanje nevrokemije, potem še kot nadgradnjo področje biokemije, nato kemije, molekularne biofizike in tako dalje do subnuklearne fizike. Druge domene, kot so bioresonanca, psihologija, psihiatrija, fiziologija, joga, obvladovanje stresa in vadba, lahko tudi vsebujejo pomembno znanje o glavobolih. Pojavi se vprašanja torej: kdaj nehamo dodajati nove domene. Več kot je domen, bolj kompleksen je ekspertni sistem.

Zlasti koordinacija ekspertnega znanja postane zelo zahtevna naloga, ki pa je ključnega pomena. Iz izkušenj vemo, kako težko je voditi in koordinirati ekipe ekspertov za delo na projektu, še posebej, kadar se pojavijo konfliktna priporočila med skupinami. Če bi znali sprogramirati dobro koordinacijo ekspertnega znanja, potem bi lahko poskušali razviti ekspertni sistem z znanjem več ekspertov. Poskusi koordinacije znanja z različnih področij so bili narejeni s sistemi HEARSAY II in HEARSAY III. Vendar je taka koordinacija kompleksen problem, na katerega moramo gledati bolj kot raziskovalni projekt in ne kot na produkcijo dostavljivega ekspertnega sistema.

- *Ali obstajata želja in potreba po ekspertnem sistemu?* Čeprav je razvoj ekspertnega sistema koristna izkušnja, je tudi nesmiselna, če ekspertnega sistema ne bo nihče uporabljal. Če v organizaciji že obstaja veliko ekspertov, bo težko upravičiti ekspertni sistem z izgovorom, da primanjkuje ekspertnega znanja. Poleg tega, če uporabniki ali eksperti nočejo nobenega sistema, ta ne bo sprejet, četudi obstaja potreba po le-tem.

Predvsem management mora biti pripravljen sprejeti ekspertni sistem. To je še večjega pomena za ekspertne sisteme v primerjavi z običajnimi programi, saj so ekspertni sistemi nova tehnologija. To pomeni, da obstaja malo izkušenih ljudi na tem področju in več negotovosti, kaj se lahko s tako tehnologijo reši. Kljub temu bi si področje ekspertnih sistemov zaslužil več podpore, ker skuša rešiti tipe problemov, ki jih običajno programiranje ne zmore. Tveganja so sicer večja, a nagrade tudi.

- *Ali obstaja vsaj en ekspert, ki je voljan sodelovati?* Mora obstajati vsaj en ekspert, ki privoli v sodelovanje, po možnosti prostovoljno. Potrebno je razumeti, da vsak ekspert ni pripravljen, da se njegovo znanje preveri za morebitna neskladja in napake ter se ga kasneje vstavi v računalniški sistem. Četudi je v sodelovanje privolilo več ekspertov, je smiselno število sodelujočih omejiti. Različni eksperti imajo različne načine reševanja problemov, kot na primer zahtevajo različne diagnostične teste. V določenih primerih lahko celo pride do različnih sklepov. Skušati kodirati različne metode sklepanja v eno bazo znanja lahko povzroči interne konflikte in neskladja.
- *Ali lahko ekspert poda znanje v obliki, ki ga bo inženir za znanje razumel?* Čeprav je ekspert privolil v sodelovanje, so možne težave pri izražanju znanja v eksplicitni obliki. Kot enostaven primer te težave si predstavljamo razložiti, kako premaknemo prst na roki. Čeprav lahko rečemo, da se prst premakne s tem, ko skrčimo mišico, ki je v prstu, je naslednje vprašanje, kako skrčimo mišico. Druga težava v komunikaciji med ekspertom in inženirjem za znanje je ta, da inženir za znanje ne pozna tehničnih izrazov. Ta problem je precej viden pri medicinskih izrazih. Inženir za znanje lahko porabi leto dni za razumevanje govornice eksperta, kaj šele da to znanje potem prevede v enega od računalniških jezikov.

- *Ali je znanje, potrebno za reševanje problemov, večinoma hevristično in negotovo?*  
Ekspertni sistemi so primerni, kadar je znanje večinoma hevristično in negotovo. To pomeni, da je znanje osnovano na izkušnjah – izkustveno znanje – in ekspertni sistem bo moral znati sklepati po več pristopih, če začetni pristop ne vrne zelenih rezultatov. Z drugimi besedami, ekspertno znanje je lahko pristop poskusov in napak (angl. *trial-and-error*), ne pa baziran na logiki in algoritmi. Vseeno lahko ekspert reši problem hitreje kot kdorkoli, ki ni ekspert. To je dobra aplikacija za ekspertne sisteme. Če je problem enostavno rešljiv z logiko in algoritmi, je najbolje zanj napisati običajen program.

### 3 UPORABA EKSPERTNIH SISTEMOV

Ključna naloga ekspertnih sistemov je olajšati opravila na področjih, kot so računovodstvo, medicina, nadzor, finančne storitve, proizvodnja in človeški viri itd. Po navadi je problemska domena dovolj zahtevna, da enostaven tradicionalni algoritem ni sposoben proizvesti ustrezne rešitve. Temelji uspešnega ekspertnega sistema so odvisni od vrste tehničnih postopkov in razvoja, ki jih oblikujejo tehniki in eksperti. Zaradi tega ekspertni sistemi običajno ne zagotavljajo dokončne rešitve, ampak podajo neko verjetnostno priporočilo.

Primer take uporabe ekspertnega sistema na področju finančnih storitev je ekspertni sistem za hipoteke. Oddelek za posojila zanima ekspertni sistemi za hipoteke zaradi povečevanja cene delovne sile, kar naredi ravnanje in vodenje majhnih posojil manj donosno. Tam vidijo možnost standardiziranega in učinkovitega ravnanja s hipotekarnimi posojili z uporabo ekspertnih sistemov kot zelo cenjeno, saj pri odobravanju hipotekarnih posojil obstajajo toga in hitra pravila, katerih ni pri drugih tipih posojil. Druga pogosta uporaba ekspertnih sistemov na področju finančnih storitev so pri borznem trgovanju na različnih trgih. Ti finančni trgi vključujejo številne spremenljivke in človeška čustva, ki bi jih bilo nemogoče določiti deterministično, zato ekspertni sistemi uporabljajo pravila palca od ekspertov in simulacijske podatke. Ekspertni sistemi takega tipa se razlikujejo od takega ekspertnega sistema, ki podaja regionalna priporočila za trgovino na drobno, ali na primer ekspertnega sistema Wishabi, ki ga uporabljajo finančne institucije in državne vlade za pomoč pri denarnih odločitvah.

Drugi primer uporabe ekspertnih sistemov je ekspertni sistem, ki je nastal v 70. in 80. letih v računalniških igricah, ki jih danes poznamo pod imenom UI (umetna inteligenca, angl. *AI – artificial intelligence*). Tak primer sta računalniški igri simulacije bejzbola Earl Weaver Baseball in Tony La Russa Baseball. Obe sta vsebovali visoko dodelane simulacije igralnih strategij omenjenih bejzbol managerjev. Ko je uporabnik igral proti računalniku, je računalnik poizvedel Earl Weaver ali Tony La Russa ekspertni sistem za odločitev, katero strategijo igrati. Celotne odločitve, ki so imele nekaj naključnosti v sebi (na primer kdaj vreči presenetljiv met, da pretentaš igralca, ki skuša ukrasti bazo), so bile izbrane glede na verjetnosti, podane s strani Weaver ali La Russa.

Nov primer uporabe ekspertnih sistemov je avtomatizirano generiranje računalniških programov. V zadnjem času je bila komercialno predstavljena aplikacija, financirana z donacijo s strani ameriških zračnih sil, osnovana na ekspertnem sistemu, (pod imenom hprcARCHITECT) ki generira računalniške programe za sisteme z mešanimi procesorji brez potrebe po tehničnih specialcih.

Sodobno raziskovanje in razvoj je usmerjeno v uporabo ekspertnih sistemov za modeliranje človeškega vedenja in sistemih za podporo odločanju. Prvi so posebej pomembni na področju medkulturnih odnosov, zadnji pa za izboljšanje managerskih odločitev v manjših podjetjih.

### **3.1 Sodobna uporaba ekspertnih sistemov na različnih področjih**

Obstaja več glavnih področij za uporabo ekspertnih sistemov, kot so kmetijstvo, šolstvo, varovanje okolja, zakonodaja, medicina, pogonski sistemi itd. V zaključni nalogi se bomo omejili na primere ekspertnih sistemov s treh področij: kmetijstvo, šolstvo in medicina.

#### **3.1.1 Ekspertni sistemi v kmetijstvu**

Ekspertni sistem v kmetijstvu je enak drugim ekspertnim sistemom. Ekspertni sistem v kmetijstvu, tako kot tudi drugi, je baziran na znanju, kjer so izkušnje in znanje človeškega eksperta zajeta v obliki IF-THEN pravil, ki jih sistem uporablja za reševanje problemov, in to so odgovori na vprašanja glede različnih tem, vnesena preko tipkovnice, povezane nanj. Na primer: ali obstaja pri nadzoru škodljivcev potreba po pesticidih, izbira pesticidov, način mešanja in izvajanja, optimalna izbira strojev, omejevanje škode po naravnih nesrečah kot toča, zmrzal, suša itd.

##### *3.1.1.1 Rice-Crop Doctor*

Ameriški državni inštitut kmetijskega razširitvenega upravljanja (MANAGE) je razvil ekspertni sistem za diagnozo škodljivcev in bolezni za riž ter predlaga preventivne in/ali kurativne ukrep [22]. »Rice-crop doctor« predstavlja široko uporabo ekspertnih sistemov na področju kmetijstva, še zlasti na področju proizvodnje riža preko razvoja prototipa, ki vključuje nekaj glavnih škodljivcev in bolezni ter nekaj pomanjkljivosti riževih polj.

Sledeče bolezni in škodljivci so bili vključeni v sistem za identifikacijo in predlaganja preventivnih/kurativnih ukrepov. Vključene bolezni so: rižev ožig, rjave pege, ožig plašča, rižev tungro virus, lažne glive, bakterijski ožig listov, gnili plašč in bolezen pomanjkanja cinka. Vključeni škodljivci pa so: stebelasti črv, mali rižev gal, skakalec na riževe liste, zeleni listav skakalec, gandijev hrošč.

##### *3.1.1.2 CALEX*

To je na teoriji osnovan ekspertni sistem za podporo odločanja pri upravljanju kmetijskih površin, ki je bil razvit na kalifornijski univerzi [22]. CALEX lahko uporabljajo kmeti, svetovalci za zatiranje škodljivcev, ostali svetovalci in drugi upravitelji.

##### *3.1.1.3 AGREX*

Center za raziskave in napredke v informatiki Kerala je razvil ekspertni sistem, imenovan AGREX [33], kot pomoč kmetijskemu osebju za točnejše in pravilnejše svetovanje

kmetom [28]. Ta ekspertni sistem ima pomembno vlogo na področjih, kot so gnojenje, zaščita pridelka, namakanje, diagnoza bolezni pred in po žetvi sadja in zelenjave.

#### 3.1.1.4 *Ekspertni sistem za obdelavo bombaževega pridelka*

Ekspertni sistem je razvila vladna organizacija za kmetijstvo – oddelek za kmetijstvo Združenih držav Amerike za podporo predelovalcem bombaža pri upravljanju.

#### 3.1.1.5 *DSS4Ag*

DSS4Ag je ekspertni kmetijski sistem za podporo pri odločanju kmetijskim raziskovalcem. Razvit je bil v projektu *Site-Specific Technologies for Agriculture (DSS4Ag)* pod krovno organizacijo INEEL [27].

#### 3.1.1.6 *DairyPert*

DairyPert je primer razvoja ekspertnega sistema za uporabo in proizvodnjo mleka in mlečnih izdelkov v Združenih državah Amerike. Pristop izdelave združuje na pravilih baziran del ter na modelu baziran del v sistemu, ki omogoča diagnostično vrednotenje trenutne pridelave mleka in mlečnih izdelkov [13].

### 3.1.2 **Ekspertni sistemi v izobraževanju**

Na področju izobraževanja je veliko ekspertnih sistemov, ki so z uporabo različnih tehnik – od prilagodljivega hiperteksta do hipermedijev – grajeni znotraj Pametnega mentorja (angl. *Intelligent Tutoring System – ITS*). Večina sistemov navadno pomaga študentu pri učenju z uporabo prilagoditvenih tehnik tako, da prilagodi okolje glede na študentovo predhodno poznavanje tematike in študentove sposobnosti učenja.

V smislu uporabljene tehnologije ima ekspertni sistem v izobraževanju dolgo zgodovino razvoja od mikrokrmnilnikov do današnje spletne aplikacije [32] in zastopniškega ekspertnega sistema [31]. Spletni ekspertni sistem je lahko odlična alternativa zasebnemu šolanju ali šolanju doma, seveda pod pogojem, da obstaja možnost internetnega dostopa [18]. Polega tega zastopniški ekspertni sistem študentu pomaga z iskanjem gradiva od spletnega ekspertnega sistema na uporabniškem profilu. Zastopniški ekspertni sistem naj bi tudi bil sposoben oceniti uporabnike in jim vračati rezultate glede na rešene probleme.

Poleg uporabljene tehnologije se je v ekspertnem sistemu močno spremenilo tudi apliciranje metod in tehnik, začeni z enostavnim na pravilih baziranemu sistemu, sedaj pa so se tehnike spremenile, da lahko podpirajo zelo popularno mehko logiko [29] in druge hibridne tehnologije [9].

#### 3.1.2.1 *Potreba po ekspertnih sistemih v izobraževanju*

Po prepričanju Markhama so ekspertni sistemi koristni kot učni pripomoček, ker vsebujejo unikatne funkcije, ki dovoljujejo uporabnikom, da postavljajo vprašanja v obliki oblike kaj, kako in zakaj [18]. Kadar to uporabljamo v okolju predavalnice, prinaša veliko prednosti,



saj pripravi odgovor na postavljeno vprašanje brez potrebe sklicevanja na predavatelja. Poleg tega lahko ekspertni sistem poda razloge za določen odgovor. Ta funkcija je zelo dobra, saj študenti bolje razumejo in so bolj samozavestni pri nekem odgovoru [21].

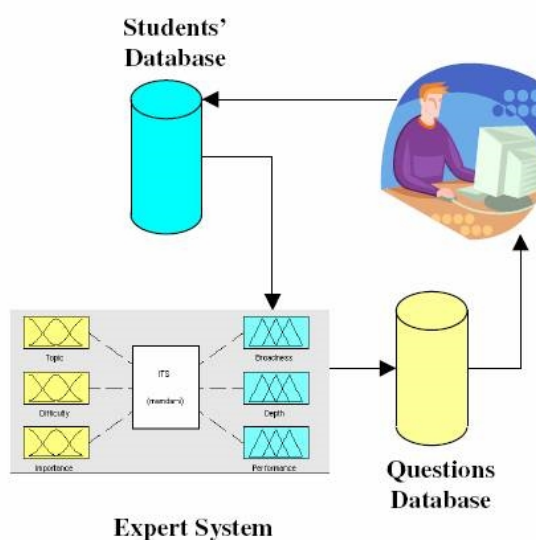
Zmožnost ekspertnega sistema, da konstantno prilagaja način in hitrost učenja vsakemu študentu posebej, je še ena funkcija, zaradi katere so ekspertni sistemi priljubljeni pri študentih. Tak način je v uporabi pri učenju bodočih strojnih inženirjev [19]. Funkcija mora biti sposobna spremljati študentov učni napredek in se odločati, kateri bo naslednji korak v učnem procesu.

### 3.1.2.2 Ekspertni sistem za poučevanje uvodnih podatkovnih struktur

Ta ekspertni sistem je uporabljal svetovni splet kot medij za dostop do informacij. Razvit je bil s pomočjo CLIP mehanizma za sklepanje in HTML programa kot uporabniški vmesnik za sistem. Po pričanju Markham-a je bil omenjeni ekspertni sistem odlična alternativa privatnemu učenju [18]. Še več, ekspertni sistem je bil razvit z javansko tehnologijo, kar pomeni, da je bil sistem neodvisen od platforme.

### 3.1.2.3 Ekspertni sistem za strojne inženirje

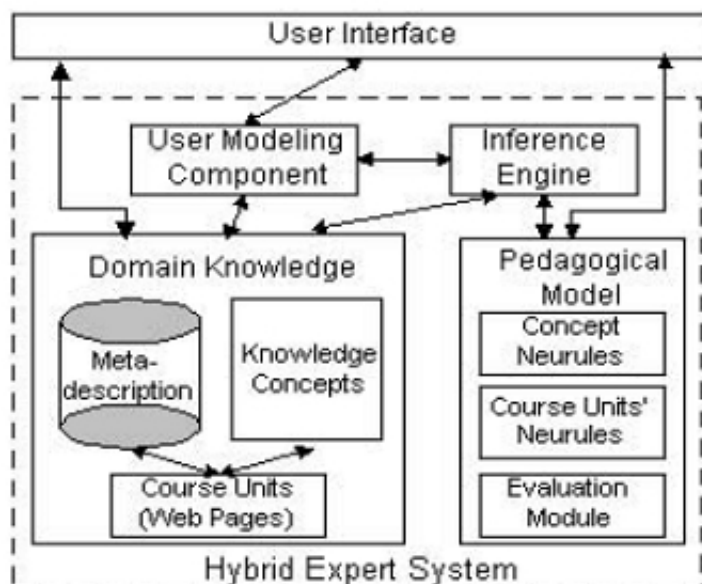
Inženirski ekspertni sistem uporablja mehko logiko namesto standardne Boolove logike v mehanizmu za sklepanje, kar omogoča bolj prilagodljivo delovanje. Ta ekspertni sistem so razvili kot pomoč študentom prvih letnikov inženirskih študijev, da bi pridobili globlje razumevanje osnovnih temeljev in da bili tako zmožni slediti naprednejšim temam na različnih inženirskih področjih. Tak pametni mentor pomaga študentu s prilagodljivim delovanjem tako, da prilagodi tempo učenja vsakemu študentu, tako kot le-temu najbolj ustreza. Pametni mentor spremlja študentov učni proces in se zna odločiti za naslednji korak v učnem procesu.



Slika 6: Struktura sistema PT za poučevanje študentov strojništva. Pridobljeno iz Zorica Nedić, Vladimir Nedić in Jan Machotka (2002)

### 3.1.2.4 Ekspertni sistem za učenje interneta

Hibridna ekspertna sestra je bila razvita za pomoč učiteljem v učenju novih tehnologij, kot je na primer internet [9]. Razvit je bil pametni mentor za učenje novih tehnologij za srednješolske učitelje. Slika spodaj prikazuje arhitekturo razvito za omenjeni pametni mentor.

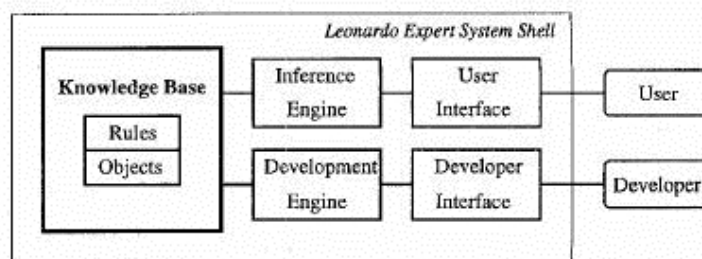


Slika 7: Arhitektura sistema PT. Prirejeno po Jim Prentzas, Ioannis Hatzilygeroudis, C. Koustojanis (2001)

Taka arhitektura uporablja prednost formalnega zapisa znanja ekspertnega sistema, baziranega na »nevropравilih«, ki so tip hibridnih pravil, vsebujejo simbolična pravila in nevroračunalništvo. Ta nevropравila izboljšujejo zmogljivost simboličnih pravil in istočasno ohranijo naravnost in modularnost.

### 3.1.2.5 Ekspertni sistem za poučevanje procesa analize napak

Aplikacijo ekspertnega sistema so uporabljali predavatelji za poučevanje študentov na temo analize napak. Z uporabo takega sistema je predavatelj cilj na bolj zanimivo in bolj produktivno učenje brez potrebe po večjem številu osebja. Ta pametni mentor je bil razvit z uporabo Leonardove lupine za ekspertne sisteme, to je objektno usmerjeno orodje za razvoj aplikacij ekspertnih sistemov.



Slika 8: Osnovni gradniki Leonardovega ekspertnega sistema. Prirejeno po Negnevitsky, M. (1998)

Slika 8 predstavlja arhitekturo Leonardove lupine za ekspertne sisteme. To arhitekturo je uporabil Negnevitsky za razvoj ekspertnega sistema za poučevanje procesa analize napak [20]. Po njegovem pričanju je Leonardov tutorski sistem zelo uporabno orodje za poučevanje procesa analize napak v elektroenergetskih sistemih. Oktobra 1994 je bil ta sistem nameščen na računalnik, ki je bil povezan v omrežje, tako je bil sistem dostopen iz kateregakoli računalnika na Oddelku za električni in elektronski inženiring. Tako so ugotovili, da je mrežni prenos računalniških vodičev stroškovno najbolj učinkovita metoda [20].

### 3.1.3 Ekspertni sistemi v medicini

Zelo zgodaj je kazalo, da so bili podobno misleči doktorji in znanstveniki očarani nad potencialnim vplivom tehnologije v medicini [14]. Z pojavitvijo pametnih računalnikov, ki lahko hranijo in procesirajo ogromne količine znanja, se je pojavilo upanje, da bojo postali »perfektni doktorji« in znali asistirati ali celo presegli zdravnike v opravilih, kot je na primer diagnoza.

Pri opisovanju takrat novega področja sta leta 1984 Clancey in Shortliffe oblikovala tako definicijo [5]:

»Umetna inteligenca v medicini je primarno usmerjena v razvoj aplikacij, ki opravljajo diagnozo in predlagajo terapevtska priporočila. Za razliko od drugih aplikacij v medicini, ki temeljijo na drugih programerskih metodah, kot so na primer čisto statistične ali verjetnostne metode, medicinske aplikacije, osnovane na umetni inteligenci, so grajene na osnovi simboličnih modelov bolezni in njihovih odnosov do pacientovih dejavnikov in kliničnih znakov.«

Do težav pa je prihajalo zaradi neustreznega načina vključevanja v klinično prakso – bodisi so reševali vprašanja, ki niso bila zaznani kot problemi, bodisi so uvajali spremembe načina dela doktorjev. Kar se sedaj ugotavlja, je to, da le ko se take »pametne programe« vključi v primerno vlogo, zares prinašajo pomembne koristi. Ena pomembnejših nalog sedanjih razvijalcev sistemov osnovanih na umetni inteligenci je ta, da morajo natančno opredeliti tiste vidike medicinske prakse, ki so najprimernejši za uvedbo sistemov z umetno inteligenco.

Ekspertni sistem in sistem, osnovan na znanju, sta dva najbolj pogosta tipa umetno-inteligenčnih sistemov v medicinski vsakdanji rabi (angl. *Artificial Intelligence in Medicine*). Taki sistemi vsebujejo medicinsko znanje, po navadi o nekem posebej določenem opravilu, in so zmožni sklepati s podatki o posameznem pacientu tako, da podajo utemeljen sklep. Čeprav obstaja veliko različic, je znanje v ekspertnem sistemu tipično hranjeno v obliki množice pravil.

### 3.1.3.1 *CaDet*

CaDet je računalniško osnovan klinični sistem za podporo odločanju pri zgodnjem odkrivanju rakovih obolenj [6]. Ocena tveganja rakovega obolenja in zgodnje odkrivanje so predmet številnih omejitev, odvisnih od človeškega faktorja in lastnosti vključenih podatkov. Za pomoč preseganja teh omejitev so razvili računalniški sistem, ki zdravnikom prinaša bolj jasno klinično sliko in pomaga pri usmerjanju pacientov k primernemu ukrepanju.

Začetne klinične in epidemiološke podatke za odkrivanje rakovih obolenj in rakovih tveganj so pridobili iz literature in jih vnesli v podatkovno bazo, skupaj z hevrističnimi pravili za vrednotenje podatkov. Vhodni podatki v CaDet so individualni podatki, pridobljeni z vprašalnikom neposredno od pacientov. CaDet kot izhod poda poročilo, v katerem so povzeti pacientovi podatki in hipoteza o rakovem tveganju s sistemom točkovanja, ki odraža alarmno stopnjo.

### 3.1.3.2 *DXplain*

DXplain je primer kliničnega sistema za podporo odločanju. Razvili so ga v splošni bolnišnici v Massachusettsu [3]. Sistem se aktivno uporablja v fazi diagnostike in vključuje množico kliničnih ugotovitev, vključujoč tudi bolezenske znake, simptome in podatke iz laboratorija, ter tako proizvede razvrščen seznam diagnoz. Poleg tega ponuja razlago za vsako od teh diagnoz in predlaga nadaljnje preglede. Sistem vsebuje neobdelane verjetnosti za več kot 4500 kliničnih znakov, ki so povezani z več kot 2000 boleznimi.

DXplain je v rutinski uporabi v veliko bolnišnicah in zdravstvenih ustanovah, večinoma za medicinsko izobraževalne namene. Možno ga je uporabiti tudi za klinično posvetovanje ali kot elektronski zdravstveni učbenik. Ponuja razlage za več kot 2000 različnih bolezni z znaki in simptomi, ki se pojavijo pri le-teh, ter ustrezno nedavno referenco za vsako bolezen.

Ni potrebno, da so sistemi za podporo odločanju samostojni sistemi, ampak so lahko integrirani kar v elektronski sistem zdravstvenih kartotek. V resnici je bolje, da sistem ni samostojen. Tak način namreč zmanjša število ovir za uporabo takega sistema s tem, da jih približa kliničnim delovnim procesom, in ne ustvarja pričakovanja, da bojo zdravniki pri sebi ustvarjali nove procese, ki bi jih lahko uporabljali.

### 3.1.3.3 *Germwatcher*

Germwatcher je bil razvit z namenom, da bi pomagal v oddelkih za obvladovanje okužb v bolnišnicah. Bolnišnice, v katerih so se izvajale aktivnosti za obvladovanja okužb, so bile povezane z univerzo oz. z univerzami. Te aktivnosti so vključevale tudi nadzor podatkov mikrobioloških kultur [12].

Bolnišnične okužbe – okužbe, pridobljene v bolnišnici (angl. *nosocomial infections*) – predstavljajo pomemben razlog za povečanje števila dni hospitaliziranosti in dodatnih stroškov bolnišnice. Germwatcher z uporabo baze pravil, sestavljene iz kombinacije NNIS kriterijev in kontrolne politike lokalnih bolnišnic za obvladovanje okužb, pregleduje podatke različnih kultur in razpozna tiste, ki predstavljajo bolniške okužbe. O teh okužbah se potem poroča Centru za obvladovanje in preprečevanje okužb (CDC).

## 4 PROTOTIP EKSPERTNEGA SISTEMA

Eden od ciljev diplomske naloge je pokazati, da ekspertni sistemi niso tako komplicirani, kot se morda sprva zdi. Cilj diplomske naloge je pokazati način, kako se dejanski problem po analitični poti preslika v svet ekspertnih sistemov (in po možnosti tudi reši).

*Sistemska dinamika* je pristop za razumevanje vedenja kompleksnih sistemov. Naravni pojav je kompleksen sistem. Prav tako je tudi ekspertni sistem kompleksni sistem – naravni pojav na enem koncu, ekspertni sistem na drugem. Vmes je sistemska dinamika. Sistemska dinamika predstavlja metodologijo za razumevanje, predvidevanje, posploševanje in poenostavitev naravnih problemov. Od tod povezava med dejanskim problemom in ekspertnim sistemom.

V tem sklopu zaključne naloge bomo predstavili prehod od dejanskega problema do računalniškega modela sistemske dinamike. Opisali bomo tudi model in njegovo delovanje.

### 4.1 Modelacijsko okolje

Obstaja več orodij za modeliranje problemov sistemske dinamike. Velika večina orodij je ozko usmerjenih in plačljivih, ker so namenjena točno določeni tržni niši. V diplomski nalogi smo se odločili za modelacijsko okolje Vensim PLE, ker je brezplačen za študijske namene. Vensim PLE je enostaven za uporabo, a ima vseeno dovolj zmogljivo logiko in vse potrebne funkcije za dovolj dobro modeliranje sistemske dinamike. Omeniti velja, da je modelacijsko okolje Vensim najbolj uporabljano med študenti ekonomije in managementa, saj je bil zasnovan z namenom, da bi študentje bolje razumeli kompleksne družbene pojave.

#### 4.1.1 Vensim

Vensim je modelacijsko orodje, ki omogoča konceptualizacijo, dokumentacijo, simulacijo, analizo in optimizacijo modelov sistemske dinamike. Omogoča da na enostaven in fleksibilen način izdelamo svoje modele iz vzročnih zank in diagramov.

#### 4.1.2 Sistemska dinamika

Vensim uporablja za modeliranje **sistemske dinamike**. Sistemska dinamika je ena od simulacijskih metodologij, ki definira sistem, sestavljen iz spremenljivk, ki neposredno ali posredno vplivajo na predstavljen problem, in rešuje probleme z opredeljevanjem dinamičnih zank sistema s simulacijo, potem ko je bila opravljena analiza odvisnosti med spremenljivkami iz kvantitativnega zornega kota. Z drugimi besedami je to primeren

način za raziskovanje rešitev problemov kot optimizacijska tehnika za analizo, kako sistemi delujejo kot celota, namesto da bi analizirali njihove podrobnosti. Sistem se predstavi z vzročno-posledičnim grafom z uporabo tehnik sistemske dinamike, in »+« predstavlja pozitiven, »-« pa negativen učinek.

Simulacijski model se sestavi tako, da se poveže besede z puščicami, odnosi med sistemskimi spremenljivkami se vnesejo in shranijo kot vzročne povezave. Te informacije uporabi *urednik enačb* kot pomoč pri simulaciji kompletnega simulacijskega modela. Poleg tega Vensim omogoča, da analiziramo svoj model v fazi izgradnje z različnimi pogledi na *vzročno drevo* in/ali *drevo posledic* določene spremenljivke ter tudi s pogledom, ki nam prikaže zanke v modelu (če te obstajajo), tudi nanašajoč na neko spremenljivko. Ko smo model dokončno zgradili in je že pripravljen za simulacijo, nam Vensim omogoča, da temeljito raziščemo obnašanje le-tega.

### 4.1.3 Paradigma strukture in vedenja

Vensim je organiziran okrog modelov in izhodnih podatkov ali rezultatov simulacije, ki se nanašajo na te modele. Ta dva koncepta sta pogosto označena kot *struktura* in *vedenje*, ki jih Vensim strogo loči. Če pogosto uporabljamo razpredelnice, smo verjetno navajeni na mišljenje, da razpredelnica vsebuje oboje: enačbe in razmerja ter podatke (številke), ki te jih ustvarijo. V Vensimu enačbe in razmerja sestavljajo model. Številke, ki jih ustvarijo, pa so obravnavane kot poskus modela in se hranijo kot podatkovni nizi ločeno od modela. Na tak način lahko uporabnik poskusi model na veliko načinov in ohrani vse rezultate, ne da bi potreboval dodatno mesto za hranjenje.

### 4.1.4 Uporabniški vmesnik

Vensim uporablja *okolje delovne mize* kot metaforo za obravnavo modelov in podatkov. Program je podoben delovni mizi, ki omogoča izgradnjo in analizo modela in na njega nanašajoče podatke. »*Delovna miza*« vsebuje meni, model, spremenljivko, podatkovne nize, orodno vrstico, eno ali več garnitur, nadzorne gumbe, orodje za izris oken in okna za izgradnjo modela.

## 4.2 Logika prototipa

### 4.2.1 Domensko znanje

Domensko znanje ekspertnega sistema, predstavljenega v zaključni nalogi, je *varnost posameznika na internetu*. S tem imamo v mislih izpostavljenost grožnjam, ki pretijo nad povprečnim uporabnikom na internetu. Izhajali bomo iz statističnih podatkov, s koliko grožnjami v povprečju se povprečni uporabnik na internetu sreča v določenem časovnem obdobju pod določenimi pogoji. Za poenostavitev modela smo se odločili, da bosta glavna faktorja kot vhodna podatka **stopnja uporabe družbenih omrežij** in **stopnja uporabe e-**

**poslovanja.** To se nam je zdelo smiselno, saj 65 % odraslih na internetu uporablja družbena omrežja in skoraj da ni zaposlenega, ki ne bi uporabljal elektronske pošte in sorodnih aplikacij po službeni dolžnosti [16].

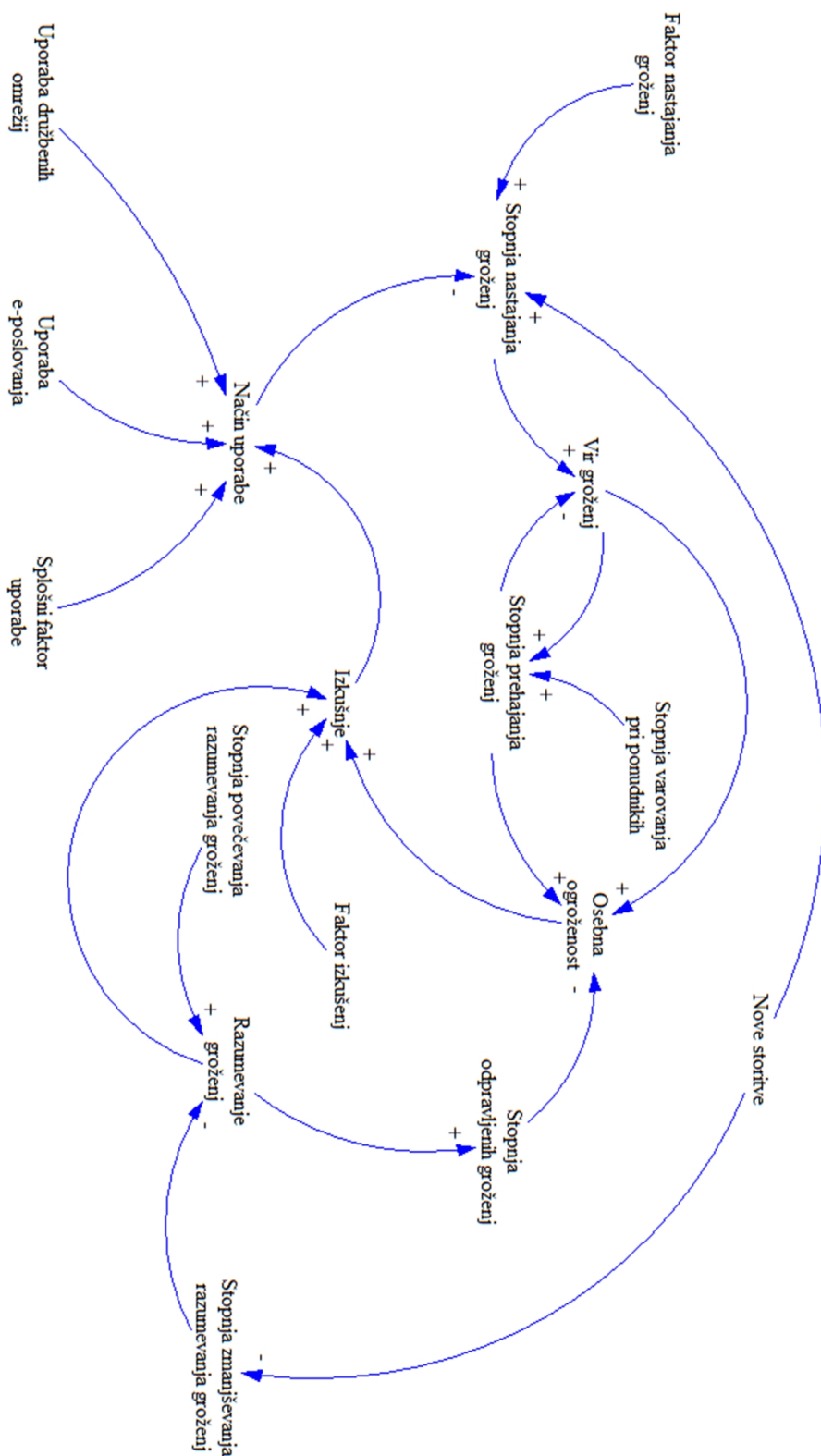
#### 4.2.2 Splošen opis logike

Grožnje na internetu kot tudi v realnem svetu imajo neko naravno stopnjo zaradi zlonamernih motivov posameznikov ali skupin (kriminal). Zato je stopnja nastajanja odvisna tudi od novih storitev, ki prehajajo v uporabo in s tem tudi na te storitve povezane grožnje. Na tem delu je zato tudi precej pomembno varovanje na strani ponudnikov storitev. Oni imajo s sistematičnim pristopom možnost, da v znatni meri zmanjšajo število groženj, ki prispejo do končnega uporabnika. Tak način sicer ne deluje na grožnje tipa *družbenega inženiringa* (angl. »social engineering«), saj je te grožnje z tehnične strani zelo zahtevno preprečiti oziroma jih je nemogoče preprečiti v celoti. Primer take grožnje je, ko nepridiprav pošlje svoji žrtvi sporočilo, (naj bo to elektronska pošta ali osebno sporočilo na strani kakšnega družbenega omrežja), v katerem se predstavlja kot skrbnik spletne storitve in v večini primerov zahteva uporabniško geslo zaradi vzdrževanja ali kaj podobnega. Izkušen uporabnik to seveda mirno označi kot nezaželeno pošto, a po drugi strani nekoliko bolj naiven uporabnik z izgovorom v glavi, češ da se sam ne razume na te stvari, veselo ustvari sporočilo z zaupno vsebino – s svojim uporabniškim geslom. Ker smo ljudje različni in se spoznamo na različne stvari, smo zaradi tega v logiko modela vključili faktor izkušenj in način uporabe. V teh dveh spremenljivkah, ki sta si tesno povezani, je logika izkušenega in naivnega uporabnika, ter tudi uporabnika, ki je nekaj že prestal in pozna nevarne točke ter ozka grla varnosti na spletu.

V modelu je zajeta tudi proaktivnost uporabnika – kaj lahko uporabnik stori sam, preventivno ali kurativno. Lahko si na primer namesti protivirusni program, požarni zid, programsko opremo za preprečevanje internetnih črvov in neželene pošte.

Model mora vsebovati vso omenjeno logiko in se nanjo glede na različne vhodne podatke in preteklega časovnega obdobja smiselno odzvati.





Slika 9: Relacijski diagram prototipa

### 4.2.3 Motivacija za domeno

Odločitev, da je domensko znanje prototipa družboslovne narave, je predvsem zaradi lažjega razumevanja ekspertnega znanja prototipa, saj je bistvo izgradnja in delovanje prototipa in ne pa njegovo domensko znanje. Pomembna je logika sklepanja, saj znanje se spreminja, logika reševanja problemov pa ostaja enaka.

### 4.2.4 Relacijski diagram prototipa

Slika 7 prikazuje relacijski diagram logike prototipa, ki prikazuje količine, ki smo jih pri obravnavi problema upoštevali, in njihove medsebojne odnose. Količine so predstavljene s opisnimi besednimi zvezami, medsebojni odnosi pa s puščicami in znakoma plus (+) in minus (-). Plus pomeni, da količina na izvoru puščice pozitivno vpliva na količino na ponoru puščice, torej več prve količine poveča drugo količino, minus pa obratno, več prve količine pomeni manj druge količine. Kot primer nove storitve na eni strani povečujejo stopnjo ogroženosti, ker prinašajo nove potencialne grožnje za uporabnika, po drugi strani pa zmanjšujejo razumevanje groženj iz istega razloga.

## 4.3 Razvoj prototipa

Ekspertno znanje prototipa je iz področja internetne varnosti. Cilj je, da ta prototip ekspertnega sistema zna modelirati osebno ogroženost povprečnega uporabnika na internetu glede na uporabniško stopnjo razumevanja in drugih spremenljivk, ki opredeljujejo stopnjo ogroženosti uporabnikov.

### 4.3.1 Zajemanje znanja

Področje je bilo izbrano, kot že omenjeno, izključno iz razloga lažjega razumevanja znanja, saj je bistven postopek in pogled na svet skozi oči ekspertnih sistemov. Postopek zajemanja znanja – najzahtevnejši del razvoja – je precej drugačen kot pri običajnem programiranju, zato smo temu posvetili več časa in pozornosti. Običajni postopek zajemanja znanja je sestavljen iz vrste tehnik za intervjuvanje in ostalih tehnik izpraševanja, pri tem prototipu ekspertnega sistema smo s kombinacijo raziskovanja in iskanja informacij iz različnih virov (internet, knjige, t. i. eksperti) empirično znanje internetne varnosti poskušali združiti v smiselno celoto, ki bo primerna za prototip. Namenoma smo so odločili za splošno znanje, da ne bi brez potrebe zapletli logike sistema, saj bi se v nasprotnem primeru lahko kar hitro izgubili na področju internetne varnosti, kar pa ni tema te zaključne naloge. Po strukturiranju znanja na področju internetne varnosti je bilo potrebno pogledati s strani človeškega faktorja – kaj povprečni uporabnik lahko stori (ali lahko zavestno zmanjša ogroženost ali se lahko odzove na konkretne grožnje in nenazadnje koliko od tega dogajanja sploh razume).

### 4.3.2 Uporabniški vmesnik

Komunikacijo z ekspertnim sistemom se izvaja preko uporabniškega vmesnika, kar smo rešili tako, da smo kot sistem za izdelavo prototipa uporabili programski paket Vensim, ki ima svoj dobro razdelan uporabniški vmesnik. Preko tega smo lahko prevedli ta idejni prototip v računalniški model. Končni produkt je model sistemske dinamike.

### 4.3.3 Razvojna metoda

Prototip smo razvijali po metodi **hitrega prototipiranja**.<sup>2</sup> Model smo naredili najprej na list papirja, nato ga vnesli v programsko okolje Vensim.

Vensim ima vgrajena dva načina preverjanja pravilnosti modelov: **preverjanje modela** in **preverjanje količin**. V Vensimu je potrebno model pravilno urediti, to pomeni povezati vse količine med seboj na smiseln način in določiti medsebojno odvisnost oziroma enačbe (preverjanje modela) ter količinam pravilno dodeliti enote (preverjanje količin). Preverjanje modela na grobo preveri, če uporabnik ni pozabil kašne količine nedoločene ali nepovezane. Pri preverjanju količin pa gre za matematično preverjanje pravilnosti enot. Enote moramo določiti tako, da se glede na enačbe v količinah pravilno krajšajo, saj v nasprotnem primeru nam Vensim javi napako.

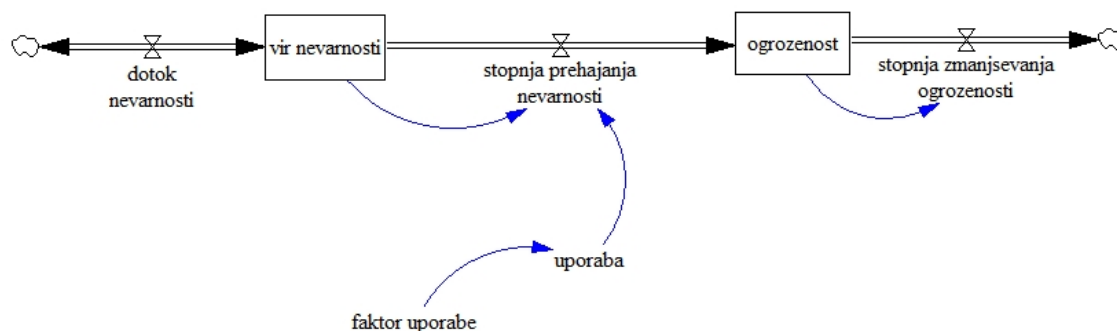
Ko je pravilnost modela zagotovljena, model testiramo z vhodnimi podatki. Potegnemo črto, ugotovimo, v katerih pogledih se model izkaže za dodelanega in v katerih je potreben dodaten trud. Tukaj se cikel ponovi. Ponovno smo v fazi razmisleka, kako narediti model oziroma ga dodelati iz prejšnjega cikla, nato ga narišemo na papir in dalje v Vensim in spet nastopi faza zagotavljanja pravilnosti modela, testiranje itd.

Preden smo prišlo do končnega modela, smo opravili 8 takih ciklov.<sup>3</sup> Tri cikle z večjimi spremembami smo vključili v Slike 10, 11, 12. Slika 10 prikazuje začetno idejo prototipa, kjer je bilo jasno, da se bomo ukvarjali s količinami kot virom nevarnosti, ogroženost, človeškim faktorjem uporabe in da gre pretok nevarnosti od izvira v merljiv vir nevarnosti, ki bo prepuščal grožnje v ogroženost uporabnika, tukaj grožnje opravijo svojo voljo ali pač ne in poniknejo v ponor groženj.

---

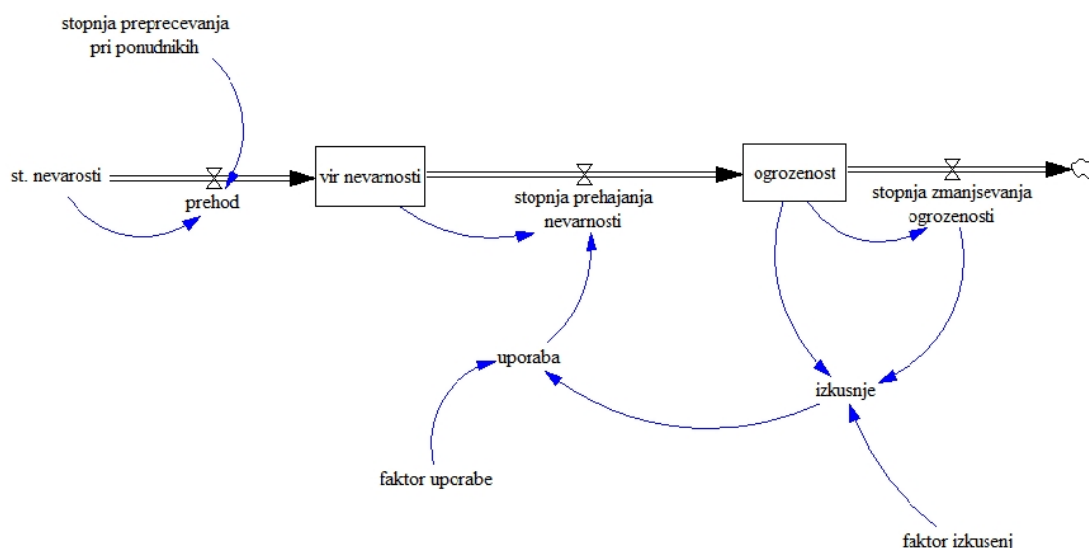
<sup>2</sup> Hitro prototipiranje ali ciklično prototipiranje je metoda za razvoj programske opreme s ponavljajočimi fazami – zajemanje znanja, izdelava prototipa nato testiranje in ponovno zajemanje znanja itd.

<sup>3</sup> Število ciklov je težko točno določiti zaradi manjših popravkov, če se naj sploh smatrajo kot celoten cikel.



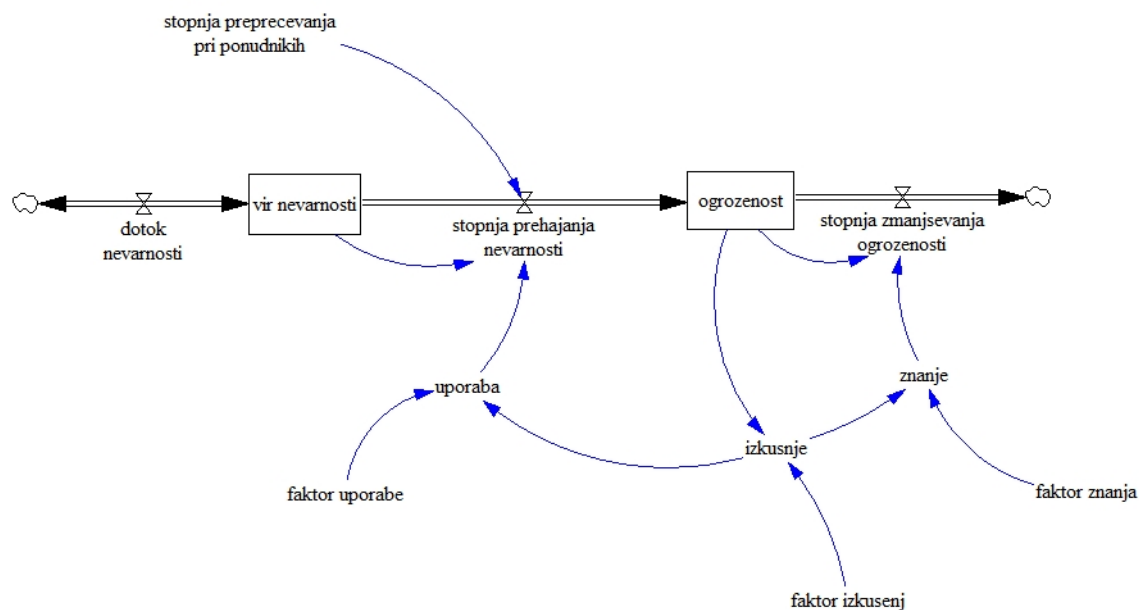
Slika 10: Vmesni prototip št. 1 – začetna ideja

Prva večja sprememba v prototipu je prišla z vključitvijo stopnje preprečevanja groženj pri ponudnikih internetnih storitev in razdelanjem uporabnikovega faktorja preprečevanja – dodali smo koncept izkušenj v model (Slika 11). Ponudniki internetnih storitev lahko kot eno od strategij privabljanja novih uporabnikov uporabijo preprečevanje groženj oziroma povečanje varnosti. Uporabniki mirno uporabljajo ponudnikove storitve brez pomislekov, saj se počutijo varne. Na drugi strani, ko se človeku zgodi neljuba prigoda, bo najverjetneje drugič želel to preventivno preprečiti. Z izkušnjami so uporabniki bolj veščji in ne padajo ponovno na osnovne potegavščine, ki se jih spomnijo nepridipravi. Oba koncepta vplivata na zmanjšanje osebne ogroženosti uporabnika.



Slika 11: Vmesni prototip št. 2 – dodelana ideja, dodelane entitete

Nadaljnja sprememba je bila, ko smo dodali koncept znanja v model (Slika 12). Uporabnik lahko ve, da ni varno pošiljati osebnih podatkov po socialnih omrežjih, ker je to slišal v medijih, mu je povedal prijatelj – splošno znano znanje ali izkustveno znanje, ko je sam bil ogoljufan pri kakšnem internetnem plačilu.

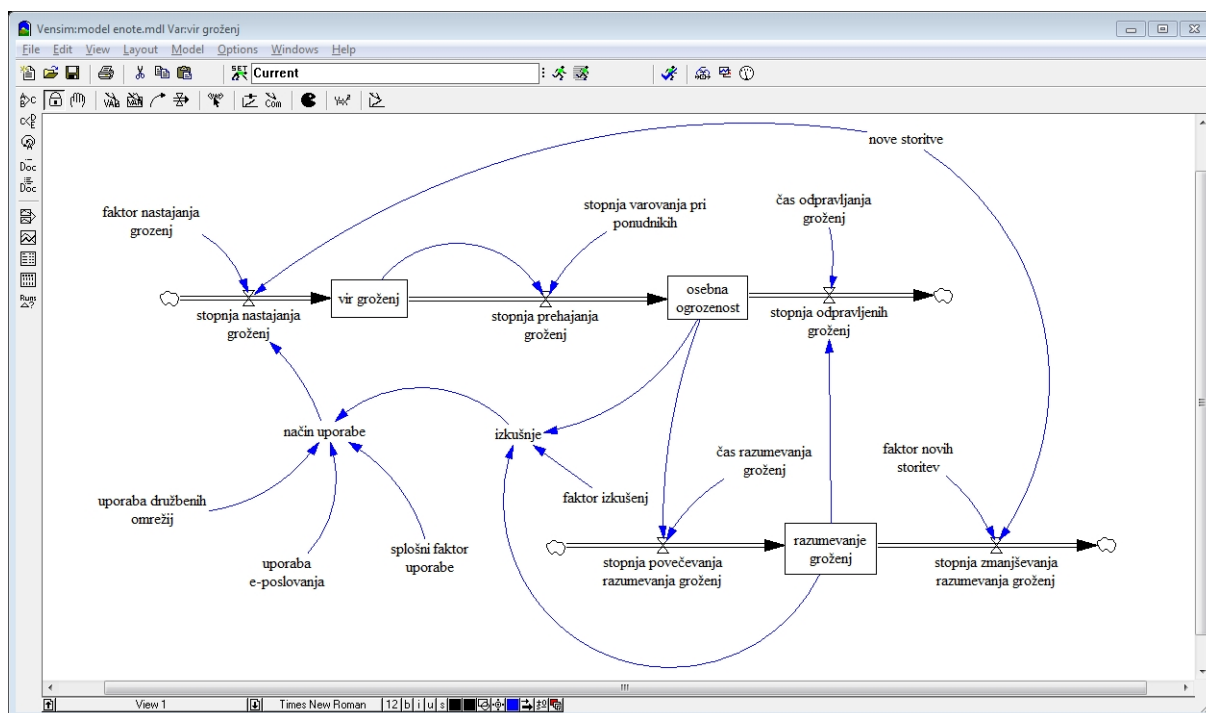


Slika 12: Vmesni prototip št. 3 – izboljšava ideje z vključenimi spremembami

#### 4.4 Opis (strukture) prototipa

Slika 13 prikazuje celoten model. V strukturi modela so 3 ključne spremenljivke in te so (očrtani pravokotniki na Sliki 13):

- **vir groženj**; ponazarja skupek vseh virov najbolj pogostih groženj na internetu, ki je odvisen tudi od uporabnikovega *načina uporabe* internetnih storitev;
- **osebna ogroženost**, ki je skupek vseh groženj, ki v določenem časovnem obdobju pretijo na povprečnega uporabnika;
- **razumevanje groženj**, kar predstavlja delež groženj, ki pretijo na uporabnika, a se teh uporabnik zaveda, jih razume in zaradi tega zna izbrati smiselne protiukrepe.



Slika 13: Končni prototip: struktura

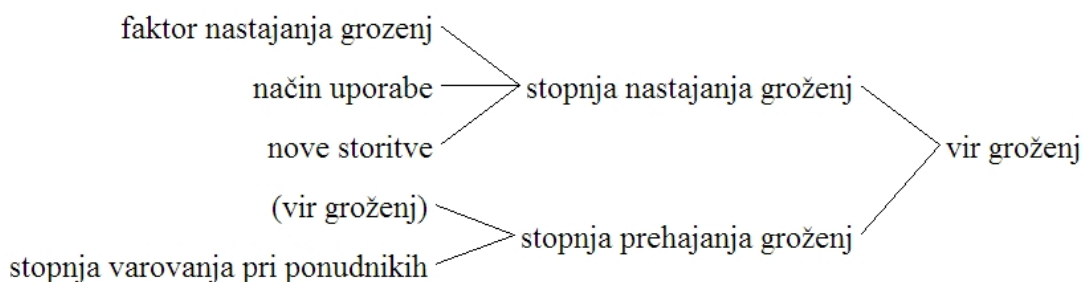
V nadaljevanju bomo za vsako od teh spremenljivk predstavil, katere spremenljivke nanje vplivajo in katere so od njih odvisne. To pomeni, da bodo na eni strani predstavljene spremenljivke, ki povečujejo dano spremenljivko, na drugi strani pa spremenljivke, ki zmanjšujejo količino v spremenljivki, poleg tega bo opredeljena tudi logika za posamezne spremenljivke in medsebojne vplive.

#### 4.4.1 Vir groženj – ključna spremenljivka 1

Na ključno spremenljivko **vir groženj** vplivata dve spremenljivki: *stopnja nastajanja groženj* in *stopnja prehajanja groženj*. Stopnja nastajanja groženj vpliva na *vir groženj* povečevalno, sama pa je odvisna od dveh drugih spremenljivk, in sicer od *faktorja nastajanja groženj* in *načina uporabe*. Način uporabe je definiran tako, da je odvisen od spremenljivk *uporaba družbenih omrežij*, *uporaba e-poslovanja* in *splošni faktor uporabe*. Faktor nastajanja groženj predstavlja naravno stopnjo nastajanja groženj, na katero uporabnik ne more vplivati, nasprotno pa je s spremenljivko način uporabe, na katero lahko uporabnik precej vpliva in to tako, da na podlagi svojega znanja in dosedanjih izkušnjah ter z odločitvami v ključnih trenutkih lahko znatno omeji svojo izpostavo internetnim grožnjam. Uporabnik tako na primer v praksi ne brska po nepoznanih straneh, ne klika na sumljive oglase, omeji uporabo družbenih omrežij, poveča stopnjo zaščite pri uporabi e-poslovanja (tukaj spadajo vse vrste kartičnega poslovanja) itd. Kot odgovor na vprašanje, zakaj taka postavitev spremenljivk pri načinu uporabe, velja omeniti podatek, da 65 odstotkov odraslih uporablja socialna omrežja ali e-poslovanje vsakodnevno [16].

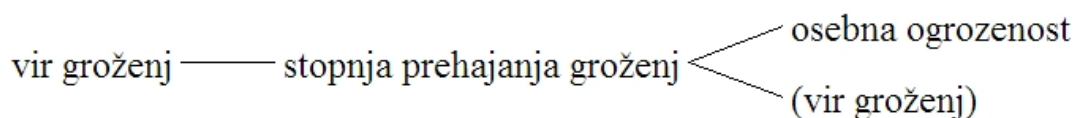
Programski paket Vensim vsebuje orodje za izrisovanje dveh tipov *dreves*: vzročna drevesa in drevesa uporabe. Vzročno drevo predstavlja vse spremenljivke, ki vplivajo na dano spremenljivko, drevo uporabe pa vse spremenljivke, na katere dana spremenljivka posredno ali neposredno vpliva. V nadaljevanju smo to orodje uporabili in izrisali nekaj slik za lažjo predstavo vzročno-posledičnih razmerij med spremenljivkami v modelu.

Iz Slike 14 lahko ugotovimo, kaj vpliva na vir groženj: neposredno vplivata stopnja nastajanja groženj in stopnja prehajanja groženj, posredno pa faktor nastajanja groženj, način uporabe, nove storitve. Zadnje tri spremenljivke prihajajo iz strani stopnje nastajanja groženj, nato še stopnja varovanja pri ponudnikih in vir groženj sam, saj kot sem že prej omenil je stopnja prehajanja groženj odvisna od stopnje varovanja pri ponudnikih, ta pa je definirana kot *delež* groženj, ki jih ponudniki uspejo uspešno preprečiti. Kako velik je delež, je odvisno od celote, torej vira groženj.



Slika 14: Vzročno drevo spremenljivke *vir groženj*

Iz Slike 15 lahko razberemo, da *vir groženj* neposredno vpliva samo na stopnjo prehajanja groženj, posredno pa še na osebno ogroženost ter sam nase. Ponovno, *vir groženj* posredno vpliva sam nase zaradi specifične definicije stopnje varovanja pri ponudnikih.



Slika 15: Posledično drevo spremenljivke *vir groženj*

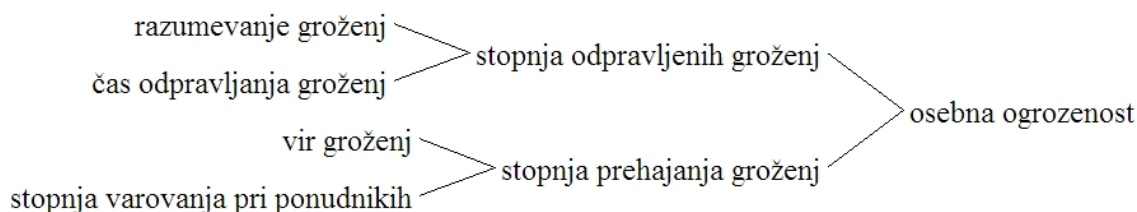
#### 4.4.2 Osebna ogroženost – ključna spremenljivka 2

Drugi ključen koncept v modelu je **osebna ogroženost**. Ta spremenljivka predstavlja količino groženj, ki v danem trenutku pretijo nad našim povprečnim internetnim uporabnikom. Grožnje v *osebno ogroženost* pritekajo iz *vira groženj* preko *stopnje prehajanja groženj* in se tako kopičijo. A grožnje se v osebni ogroženosti ne samo kopičijo, pač pa tudi odtekajo iz nje. To se dogaja na drugi strani spremenljivke, kjer se nahaja *stopnja odpravljenih groženj*, ki skupaj z *časom odpravljanja groženj* skrbi za

smiselno zmanjševanje osebne ogroženosti. Kam točno grožnje odtekajo za ta model, ni važno, edino, kar je važno razumeti, je, da te grožnje na uporabnika ne vplivajo več.

Torej iz Slike 16 je razvidno, da je na vhodni strani osebne ogroženosti stopnja prehajanja groženj, prek katere se osebna ogroženost povečuje, stopnja prehajanja groženj pa, kot že omenjeno, je odvisna od vira groženj in stopnje varovanja pri ponudnikih.

Na izhodni strani osebne ogroženosti se nahaja stopnja odpravljenih groženj, ta pa je odvisna od tretjega ključnega koncepta prototipa *razumevanja groženj* in *časa odpravljanja groženj*.



Slika 16: Vzročno drevo *osebne ogroženosti*

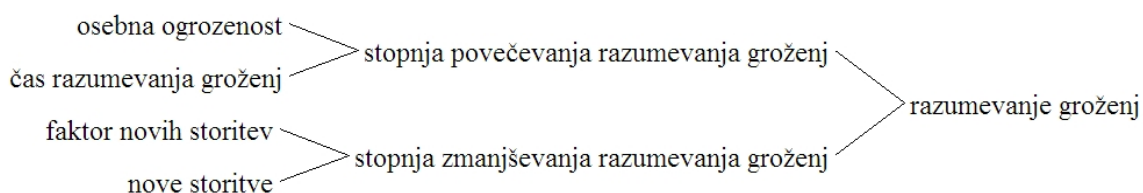
#### 4.4.3 Razumevanje groženj – ključna spremenljivka 3

Tretji in zadnji ključni koncept v modelu je **razumevanje groženj**. Logika za tem konceptom je sledeča: kot prvo uporabnik z razumevanjem groženj direktno vpliva na zmanjšanje svoje osebne ogroženosti (kurativa), kot drugo pa se to znanje prelevi v izkušnje, ki vplivajo na način uporabe, in sicer tako, da uporabnik spremeni svoj način uporabe internetnih aplikacij (preventiva).

*Stopnja povečevanja razumevanja groženj* je direktno odgovorna za večanje razumevanja groženj. Nanjo vplivata *osebna ogroženost* ter *čas razumevanja groženj*. Logika za tem je preprosta – večja kot je osebna ogroženost glede na nek konstanten čas razumevanja groženj, večja je stopnja povečevanja razumevanja groženj.

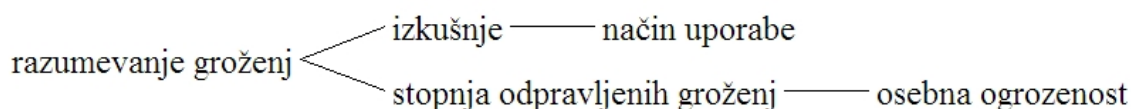
Na drugi strani *stopnja zmanjševanja razumevanja groženj* vpliva na zmanjševanje razumevanja groženj, sama pa je odvisna od *novih storitev* in *faktorja novih storitev*. To pomeni, da več kot je novih storitev glede na vpliv novih na uporabniško razumevanje letih (*faktor novih storitev*), večja je stopnja zmanjševanja razumevanja groženj, posledično je tudi razumevanje groženj manjše.



Slika 17: Vzročno drevo *razumevanja groženj*

Na spremenljivko *razumevanje groženj* torej vplivata osebna ogroženost s časom razumevanja groženj preko stopnje povečevanja razumevanja groženj ter nove storitve s faktorjem novih storitev preko stopnje zmanjševanja razumevanja groženj (Slika 17).

Od razumevanja groženj sta posredno odvisna *način uporabe* in *osebna ogroženosti*, način uporabe preko nabiranja izkušenj, osebna ogroženost pa preko stopnje odpravljenih groženj zaradi razumevanja le-teh (Slika 18).

Slika 18: Vplivno drevo *razumevanja groženj*

Na način uporabe poleg izkušenj vplivajo tudi uporaba družbenih omrežij, uporaba e-poslovanja in splošni faktor uporabe. Drugi primer je pa spremenljivka *faktor izkušenj*, ki direktno vpliva na izkušnje, logika za to spremenljivko je ta, da se določeno število razumljenih in nerazumljenih groženj pretvori v izkušnje. V praksi to pomeni, da če ima uporabnik slabo izkušnjo z neko storitvijo, aplikacijo ali s čimerkoli drugim na internetu, ne glede na to, ali razume, zakaj je bila to slaba izkušnja, bo prilagodil svoj način uporabe najverjetneje tako, da se bo aplikacije izogibal.

## 4.5 Izvedba simulacije

V tej fazi je naš prototip zgrajen. Sedaj lahko izvajamo simulacije, saj imamo delujoč prototip ekspertnega sistema, ki modelira grožnje, ki pretijo nad povprečnim uporabnikom interneta. Vhodni podatki so: faktor uporabe interneta (predvsem družbenih omrežij in e-poslovanja), stopnja zaščite pri ponudnikih internetnega dostopa, stopnja spreminjanja tehnologij in novih storitev na trgu ter uporabniško izkušnost oziroma spretnost. Izhodni podatki nastanejo v prototipu ob izvedbi simulacije, to so: količina potencialnih groženj, ki pretijo nad uporabnika, resne grožnje, ki z njegovo vednostjo ali brez vplivajo na njegovo uporabo internetnih storitev, ter grožnje, ki jih je odkril in jih na kakršenkoli način odpravil ali se jim izognil.

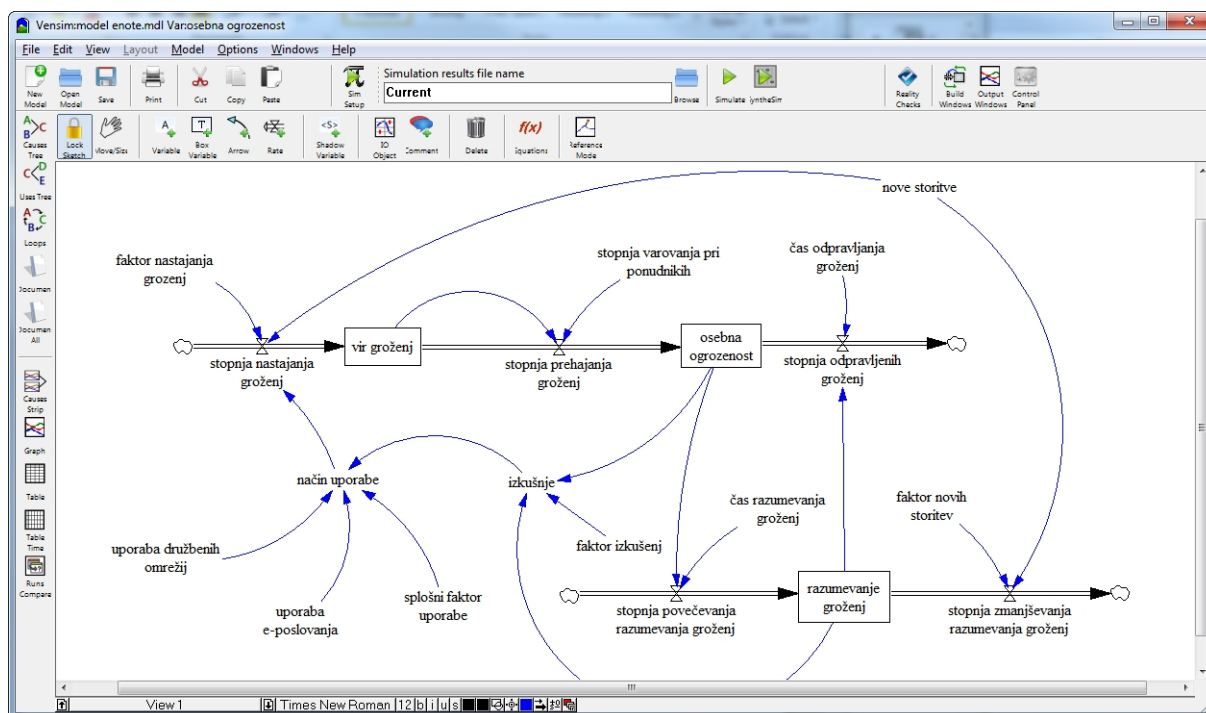
Smisel prototipa je, da uporabniku služi kot pomoč pri razumevanju dinamike groženj na uporabnike na internetu. Ob izvedbi simulacije s predhodnim vnosom določenih podatkov o načinu in količini uporabe internetnih storitev ter zaščite pri klientu ali ponudniku prototip izdelava hipotetično dinamiko gibanja groženj, spretnosti uporabnikov in njihovega medsebojnega odnosa. Za preslikavo vhodnih podatkov v izhodne pa poskrbi logika prototipa, ki je bila razložena v predhodnih poglavjih.

#### 4.5.1 Običajna simulacija modela

Ko imamo nastavljene enačbe za vse spremenljivke in pravi model (povezave med spremenljivkami), je naslednji korak simulacija. Poženemo simulacijo, da vidimo potencialen izid. Navadno je potrebnega nekaj prilagajanja parametrov, da dobimo želen oziroma smiseln izid, zato ponuja Vensim tudi posebno vrsto simulacije – **SyntheSim**.

#### 4.5.2 SyntheSim simulacija modela

Programski paket Vensim poleg običajne simulacije modela za določeno časovno obdobje s izračunom vseh podatkov ponuja tudi način preizkusa modela s sočasnim spreminjanjem vhodnih parametrov imenovan tudi *SyntheSim*. To v veliki meri olajša določanje njihove optimalne vrednosti. Predstavljajte si, da bi za vsako spremembo določenega parametra mogli izvesti simulacijo in šele nato bi si lahko ogledali simulacijske podatke: grafe, vmesne in končne vrednosti posamezne spremenljivke ali z drugimi besedami vidimo vpliv spremembe šele po opravljeni simulaciji in za vsako spremembo moramo pognati simulacijo. Ta drugi način preizkusa oziroma simulacije modela omogoča spreminjanje vhodnih parametrov, istočasno pa vidimo tudi spremembe na grafih vsake spremenljivke, na katero parameter vpliva neposredno ali posredno. To je predstavljeno na Sliki 19, kjer ima vsak parameter drsni, s katerim spreminjamo njegovo vrednost, v polju imena vsake spremenljivke pa graf, ki se spreminja ob določanju vrednosti vplivajočih parametrov.



Slika 19: Preizkus modela v načinu sočasnega spreminjanja parametrov

## 4.6 Primer izvedbe

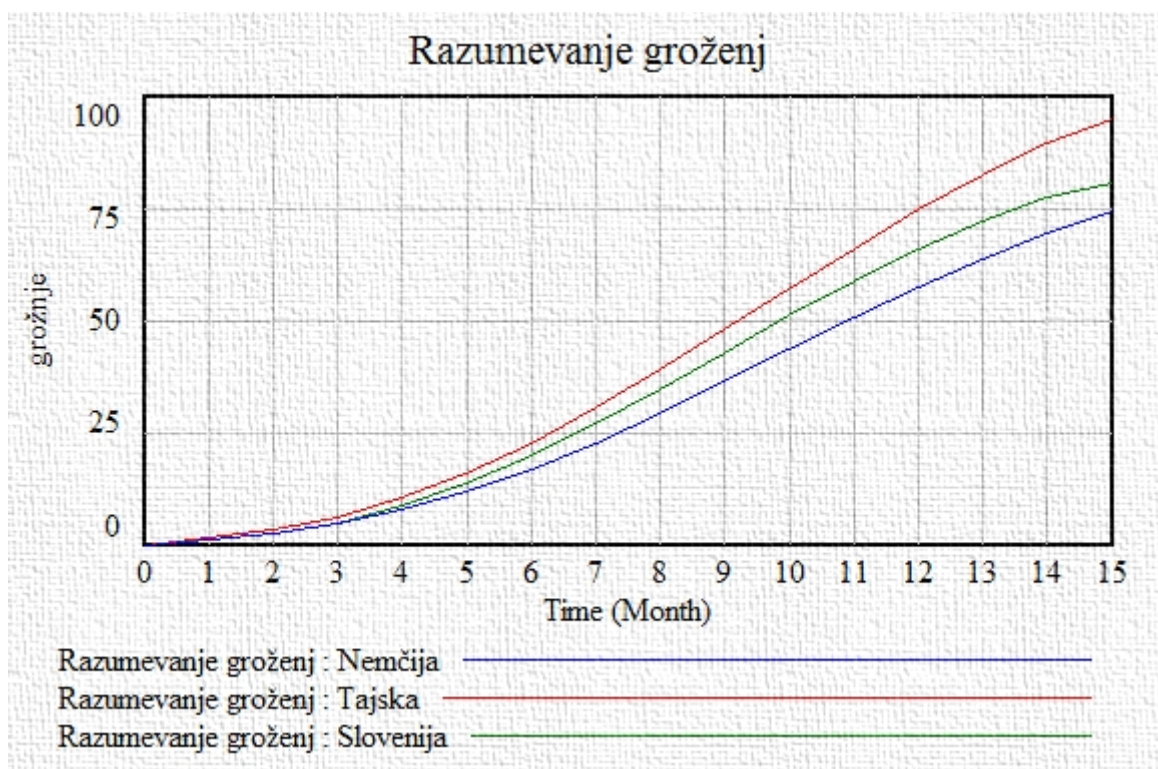
Za predstavitev izvedbe smo želeli izbrati nazoren primer, in sicer primerjanje nečesa našega s tujim. Recimo, umestiti naše (slovensko) internetno tveganje in spretnost uporabnikov glede na ostale uporabnike interneta po svetu. Smiselno bi se bilo pohvaliti, koliko »naprednejši« smo od držav v razvoju in »kako malo« za razvitimi državami. In ravno to smo naredili. Vzeli smo primer 3 držav in njihove značilnosti glede uporabe interneta in njihovih uporabnikov. Kot primer smo določili državo z visoko razvitim gospodarstvom – Nemčijo, državo v razvoju, kot je Tajski, ter našo deželo (Slovenijo) kot razvito državo, ki pa ni svetovna gospodarska velesila. Upoštevali smo 3 grupe parametrov in zaradi poenostavitve omejili »vrednosti stopenj« na: nizko, srednje, visoko, saj je tukaj bolj pomemben odnos vrednosti med vhodnimi podatki – državami kot pa točne vrednosti posameznih parametrov. Določene vrednosti grup parametrov so predstavljene v Tabeli 11.

Tabela 11: Odnos parametrov po vhodnih podatkih – državah

	Zaščita pri ponudnikih	Spretnost uporabnikov	Faktor novih storitev
Nemčija	visoka	visoka	nizka
Tajska	nizka	nizka	visoka
Slovenija	visoka	srednja	srednja

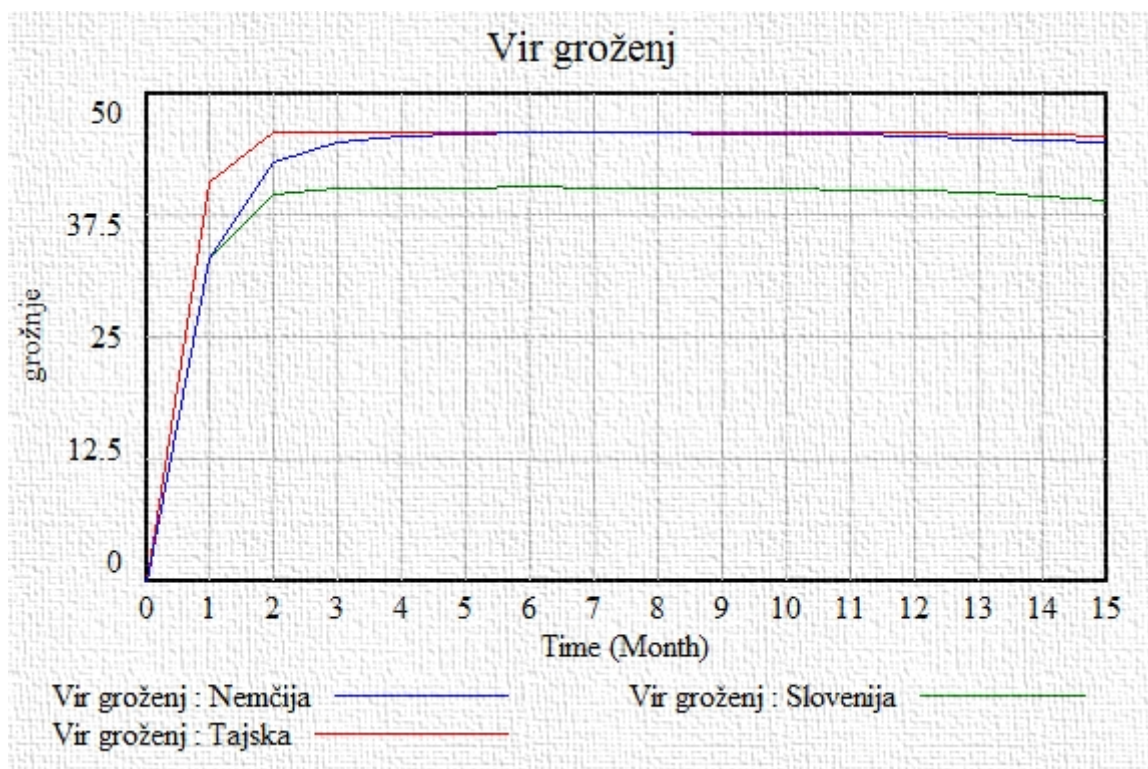
Naredili smo simulacijo z manjšim prilagajanjem vrednosti vhodnih parametrov in dobili grafe ključnih spremenljivk. Za časovno obdobje smo določilo 15 mesecev, to pa zato, ker da je v njem reprezentativno dovolj iteracij ter je dovolj kratko (1,25 leta), da ne zaidemo v nerealne futuristične napovedi.

Če opišemo rezultate, lahko ugotovimo, da razumevanje groženj vseskozi narašča, torej se uporabniki učijo in pridobivajo izkušnje, a rast vsekakor ni enakomerna (Slika 20). Rast groženj, ki jih je uporabnik dojel kot grožnje in se nanje odzval, je v začetnih iteracijah simulacije manjša, to pa zato, ker je njegova osebna ogroženost nizka (Slika 22), nasprotno pa je v iteracijah ob koncu simulacije rast nizka, ker je ogroženost nizka. Vmes je razumevanje groženj večje, kar ponazarja naklon grafa. Kar lahko iz tega povzamemo, je, da se ob večji osebni ogroženosti uporabniki več naučijo in ukrepajo, saj so v to prisiljeni.



Slika 20: Rezultat izvedbe – Razumevanje groženj

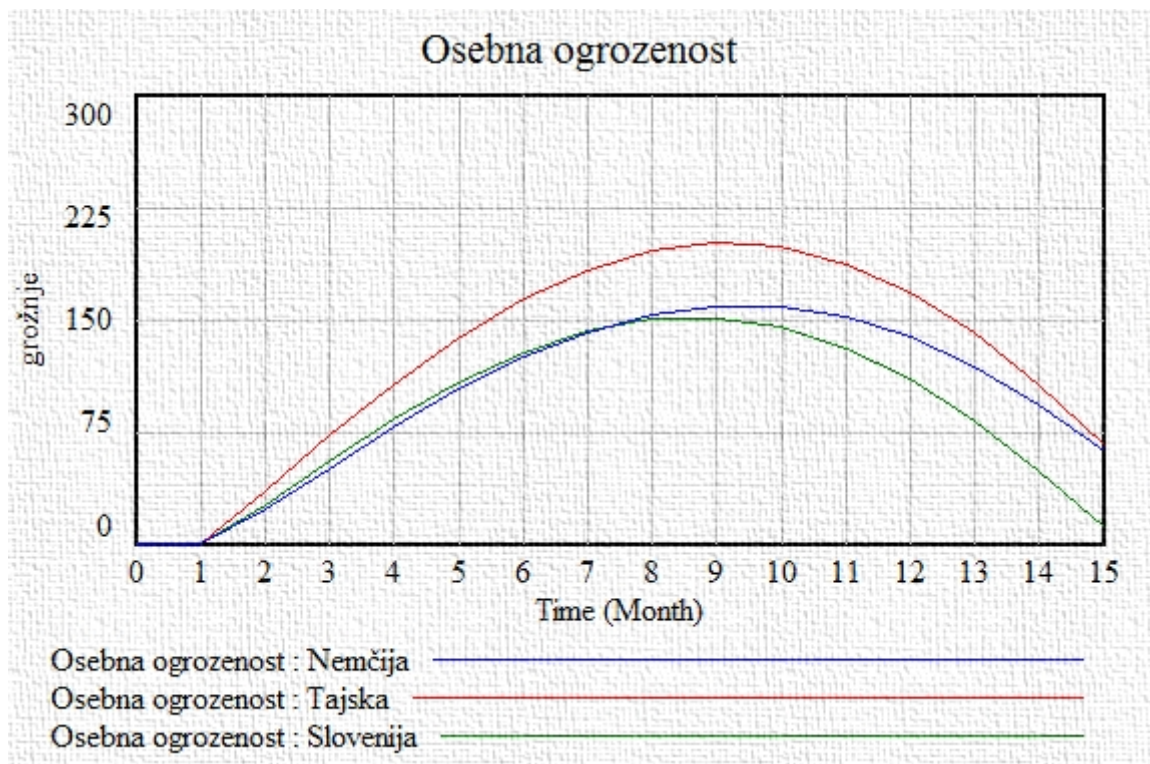
Grafi gibanja vira groženj niso posebej zanimivi, vir groženj na začetku hitro zraste do neke optimalne vrednosti in je od tedaj konstanten (Slika 21). Zanimivo pa je, da je povprečno število novo nastalih groženj v Sloveniji manjše kot v ostalih dveh primerjanih državah. Eden od smiselnih razlogov bi lahko bil ekonomske narave – potencialni dobiček uspešno izvedenih groženj je v višje razviti državi večji kot pri manj razviti, tako je kriminalni interes večji in potemtakem je groženj več.



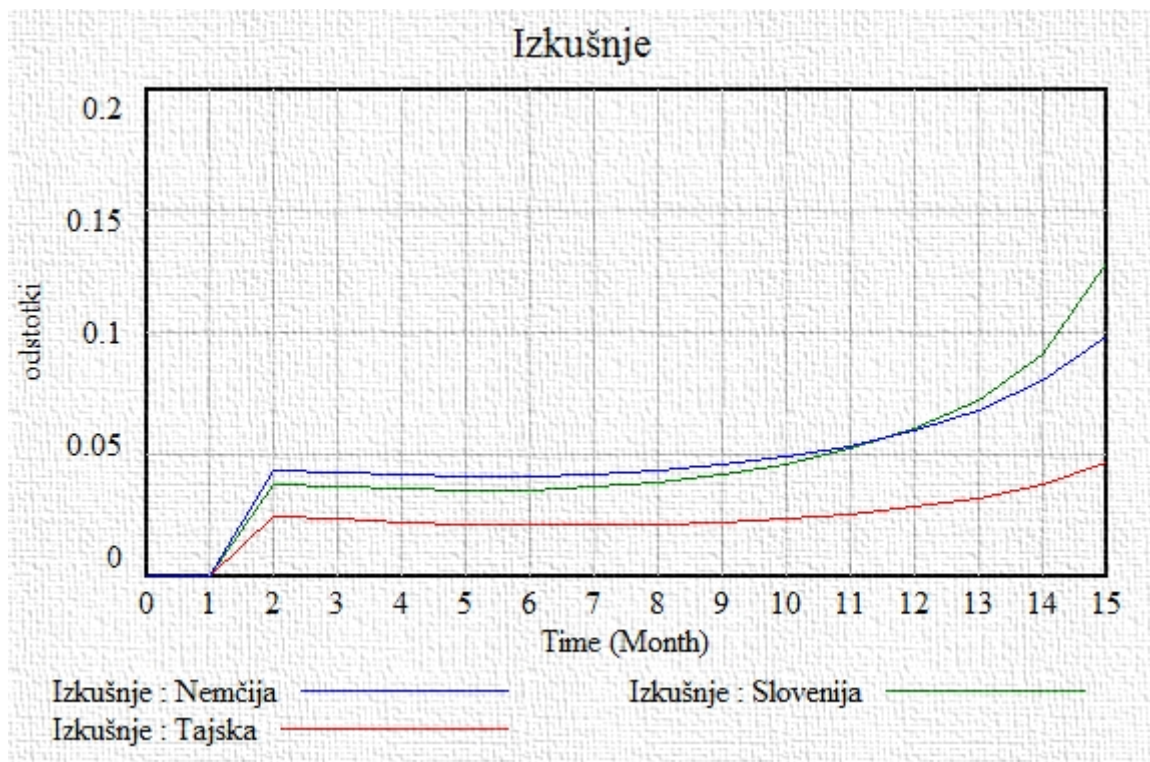
Slika 21: Rezultat izvedbe – Vir groženj

Osebna ogroženost (Slika 22) je glavni pokazatelj internetnih groženj, ki pretijo nad povprečnim uporabnikom interneta. Ta graf je najbolj nazoren od treh grafov ključnih spremenljivk, morda tudi utopičen, ker je konča osebna ogroženost povprečnega slovenskega uporabnika v zadnji iteraciji skoraj na ničli. Grafi vseh treh držav se gibajo podobno: začnejo v ničli, do nekje polovice rastejo, dalje pa padajo (ampak z različno hitrostjo). Tako gibanje je možno razložiti na sledeči način: iz vira groženj prihajajo grožnje, del teh sploh ne pride do uporabnika, saj se preprečijo že na nivoju internetnega ponudnika, ostale se kopičijo v osebno ogroženost uporabnika. Ker je za odpravo groženj potreben nek čas, grožnje pa vseskozi nastajajo, se le-te kopičijo in povečujejo osebno ogroženost uporabnikov. Večja osebna ogroženost uporabnikov zahteva večjo udeležbo uporabnikov pri odpravi groženj, tako se več njihovo znanje in pridobivajo izkušnje. Z več znanjem in izkušnjami pa so sposobni določiti in odpraviti več groženj, stvar se spiralno povečuje, več enega, več drugega, več tretjega in zaradi tega je več prvega itd. To se najbolj opazi pri grafu za Slovenijo – če pogledamo izkušnje slovenskih uporabnikov, rastejo hitreje kot nemške (Slika 23), kar pomeni, da so uporabniki bolj vešč, torej so se zmožno bolje in v večji meri spopasti z grožnjami. In ravno zaradi tega je graf za Slovenijo na Sliki 22 padel nižje kot graf za Nemčijo. Graf za Tajsko je pa tak, kot smo ga pričakovali – nizka stopnja zaščite pri ponudniku internetnih storitev, nižja stopnja spretnosti uporabnikov (znanje, izkušnje) od ostalih dveh in enak ali večji vir groženj, torej je graf najvišje (Slika 22 in Slika 23). So pa uporabniki zaradi težkega okolja primorani znati več in ko dosežejo dovolj visoko mejo izkušenosti, da odpravijo več groženj, kot jih

priteka, je samo vprašanje časa, kdaj bodo konvergirali k konstantni osebni ogroženosti razvitih držav.



Slika 22: Rezultat izvedbe – Osebna ogroženost



Slika 23: Rezultat izvedbe – dodaten graf – Izkušnje uporabnikov

## 5 ZAKLJUČEK

Ekspertni sistemi postajajo zaradi svoje uporabnosti vedno bolj pomembni. Dandanes je skoraj 74 odstotkov [24] IT tehnikov frustriranih s časom in energijo, ki jih porabijo za redna vsakodnevna opravila (iskanje napak, javljanje na telefon itd.). To ni samo nestimulativno, ampak tudi velika potrata znanja v podjetju.

Če bi vsakodnevna nizkostopenjska opravila opravljal ekspertni sistem, bi zaposleni v IT oddelku lahko uporabljali svojo ekspertno znanje in talente ter opravljali delo na svojem področju. Tako bi se izboljšala IT infrastruktura, dvignila produktivnost ter bi se pripomoglo k rasti podjetja. S takim način se lahko danes v podjetjih reši osnovni problem – povečati kvaliteto in zanesljivost IT infrastrukture z manjšim proračunom.

Na začetku zaključnega dela smo se zavezali, da bomo preučili delovanje ekspertnih sistemov in izgradili prototip ekspertnega sistema, ker smo tudi storili. Po krajšem uvodu smo v drugem in tretjem poglavju preučili delovanje ekspertnih sistemov ter predstavili njihovo uporabo v realnem svetu. V četrtem pa smo zasnovali in izgradili prototip ekspertnega sistema. Predstavili smo ga v vseh fazah razvoja. Od začetne ideje, ki smo jo risali šele na papir in dopolnjevali, preko modeliranja v programskem paketu Vensim do temeljite razlage logike in razvoja prototipa ter do končne faze – primera delovanja. V poglavju 4.2 smo opredelili razloge za izbiro programskega paketa Vensim, v poglavju 4.4 pa razloge za izbiro ogroženosti posameznika na internetu kot problemskega področja tega prototipa. Poglavje 4.3 razlaga podlago dogajanja v prototipu – problemsko področje in s katerimi aspekti se srečujemo ter katere bomo upoštevali v tem prototipu, torej kateri aspekti so relevantni in katere niso. Vensimov model, ki je jedro tega prototipa, je obširno predstavljen v poglavju 4.5. Tam so predstavljene vse količine, ki jih upoštevamo, vplivi med njimi in kaj predstavljajo. Predstavljen je tudi prikaz celotnega modela. Zadnje poglavje, poglavje 4.6, predstavlja njegovo delovanje – simulacije.

Potrebno je poudariti, da bi lahko sistem razčlenili v mnogo več podrobnostih, to je namreč le prototip, saj je bil naš glavni cilj prikazati postopek izdelave ekspertnega sistema na tak način, da bralec ob koncu branja dojame novodobno paradigmo programiranja in se zaveda prednosti in slabosti tega načina. To pomeni, da čeprav je izgradnja dobrega ekspertnega sistema zahtevna naloga (tehnično in finančno), določen tip problemov zahteva ravno tak pristop, in to tam, kjer z običajnimi pristopi ne pridemo daleč.

## 6 LITERATURA

- [1] A. Abraham, B. de Baets, M. Köppen, B. Nickolay, *Applied Soft Computing Technologies: The Challenge of Complexity (Advances in Intelligent and Soft Computing)*. Springer, Berlin, 2006.
- [2] K. Ae, M. Su, H. Dong, Compliance Risk Assessment Measures of Financial Information. (*International Journal of Security and Its Applications*) 6, št. 4 2012.
- [3] O. Barnett, C. James, J. Hupp in E. Hoffer, DXplain: an Evolving Diagnostic Decision-Support System. *Journal of the American Medical Association*, 1987: 67–74.
- [4] I. Berković, B. Radulović in P. Hotomski, Extensions of Deductive Concept in Logic Programming and Some Applications, *Proceeding of the International Multiconference on Computer Science and Information Technolog*, Wisła, 2007. 1071–1080.
- [5] W. Clancey in E. Shortliffe, Introduction: Medical Artificial Intelligence Programs, *W*, 1984: 1–17.
- [6] J. Fuchs, I. Heller, M. Topilsky in M. Inbar, CaDet, a Computer-Based Clinical Decision Support System for Early Cancer Detection and Prevention, *Cancer Detect Prevention*, 1999: 78–87.
- [7] C. Giarratano in G. Riley, *Expert Systems: Principles and Programming, Fourth Edition*. Course Technology, Boston, 2004.
- [8] M. Grabisch, H. Nguyen in E. Walker, *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference (Theory and Decision Library B)*, Springer, Berlin, 2010.
- [9] I. Hatzilygeroudis in J. Prentzas, HYMES: A HYbrid Modular Expert System with Efficient Inference and Explanation, *Proceedings of the 8th Panhellenic Conference on Informatics*. Nicosia, Cyprus: GSRT of Greece, 2001, 422–431.
- [10] C. Hempel in P. Oppenheim, Studies in the Logic of Explanation, *Philosophy of Science* 15, št. 2 (1948): 135–175.
- [11] A. Hopgood, *Intelligent Systems for Engineers and Scientists, Third Edition*, CRC Press, Boca Raton, 2011.
- [12] M. Kahn, S. Steib, V. Fraser, in W. Dunagan, An expert system for culture-based infection control surveillance, *Proceedings of the Annual Symposium on Computer Application in Medical Care*. Washington: Division of Medical Informatics, Washington University School of Medicine, 1993, 171–175.



- [13] R. Kalter in A. Skidmore, Dairypert: An Expert Systems Approach for Improving Dairy Farm Management Practices and Assisting Technology Transfer, *Research Bulletins*, Cornell University, Department of Applied Economics and Management, 1991.
- [14] R. Ledley in L. Lusted, Reasoning Foundations of Medical Diagnosis, *Science*, 1959: 9–21.
- [15] S. Lewandowsky in M. Coltheart, Cognitive modeling ‘versus’ cognitive neuroscience: Competing approaches or complementary levels of explanations?, (The Australian Psychological Society) 64, št. 1 (2012): 1–3.
- [16] M. Madden in K. Zickuhr, *65% of online adults use social*, Washington, D.C.: Pew Research Center, 2011.
- [17] I. Maglogiannis, K. Karpouzis, M. Wallace in J. Soldatos, Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval ... in Artificial Intelligence and Applications. IOS Press, Lansdale, 2007.
- [18] H. Markham, An internet-based expert system for teaching introductory data structures. *Journal of Computing Sciences in Colleges*, 2001: 155–165.
- [19] Z. Nedic, V. Nedic in J. Machotka, Intelligent Tutoring System for teaching 1st year engineering. *World Transactions on Engineering and Technology Education* 1.2, 2002.
- [20] M. Negnevitsky, „A knowledge based tutoring system for teaching fault analysis.“ *IEEE Transactions on Power Systems*, 1998: 40–45.
- [21] S. Nwigbo in C. Okechuku, Expert System: A Catalyst in Educational Development in Nigeria, *Proceedings of the 1st International Technology, Education and Environment Conference*. Omoku: Human Resource Management Academic Research Society, 2011.
- [22] G. Prasad in V. Babu, A study on various expert systems in agriculture, *Georgian Electronic Scientific Journal: Computer Science and Telecommunications* (Georgian Electronic Scientific Journal: Computer Science and Telecommunications), 2006: 81–86.
- [23] R. Ribeiro in E. Fabri’cio, *A sociologically inspired heuristic for optimization algorithms: A case study on ant systems*. 8. October 2012.
- [24] L. Russel in J. Johnston, *Average IT engineer is over-qualified and under-utilised according to IPsoft research*. London: realwire, 2012.
- [25] S. Russell in P. Norvig, *Artificial Intelligence: A Modern Approach* (2nd ed.), Prentice Hall, Upper Saddle River 2003.

- 
- [26] R. Schalkoff, *Intelligent Systems: Principles, Paradigms and Pragmatics*. Jones & Barlett Publishers, Burlington, 2009.
- [27] S. Shearer in drugi, Use of the decision support system for agriculture (DSS4Ag) for corn fertilization in Kentucky, *Proceeding of the 5th International Conference on Precision Agriculture*. Bloomington, Minnesota: American Society of Agronomy, 2000. 1–16.
- [28] S. Sihazalwa, *Development of an expert system prototype for corn plant disease and pest attack diagnosis*, Doctoral dissertation, Universiti teknologi MARA (UiTM), 2006.
- [29] M. Starek, T. Mukesh, K. Bhaskar, in M. Garcia, An expert system for mineral identification. *Journal of Computing Sciences in Colleges*, 2002: 193–197.
- [30] E. Turban, J. Aronson in L. Ting-Peng, *Decision support systems and intelligent systems*. Prentice-Hall, Upper Saddle River, 2005.
- [31] A. Vivacqua in H. Lieberman, Agent to Assist in Finding Help, *Proceedings of the SIGCHI conference on Human factors in computing system*, 2000: 65–72.
- [32] D. Woodin, Design and implementation of GungaWeb: an application of classical expert system technology to the production of Web-based commercial systems, *Proceedings of the 8th international conference on Artificial intelligence and law*, 2001: 104–108.
- [33] B. Zadeh, The Approaches to Community Development in Iranian Rural Community, *AGREX 08: Agriculture Sustainability Trought Participative Global Extension*, 2008.