

Sisteme Expert (Sisteme bazate pe cunostinte)

Curs: Boldur E. Barbat

1. Locul inteligentei artificiale in informatica. Sistemele expert-caz particular al sistemelor bazate pe cunostinte.
2. Elemente de programare in medii vizuale. Tehnici de programare pentru inteligenta artificiala in VC++. Simularea mecanismelor de inferenta (deductie, abductie).
3. Notiuni de logica matematica, algebra booleana, structuri arborescente.
4. Concepte si paliere in inteligenta artificiala. Palierul simbolic (ipoteza Newell-Simon), subsimbolic (ipoteza conectionista), atitudinea intentionala: legitimare si utilitate.
5. Agentul inteligent. Proprietati slabe si tari. Agentitate.
6. Abordari, modele, metode in sistemele bazate pe cunostinte. Sisteme expert si retele neurale artificiale (comparatie intre ele), rationament bazat pe precedente (Case Base Reasoning), strategii evolutive, algoritmi genetici (Science American - July 1992), inteligenta artificiala distribuita in retele tip Internet.
7. Arhitecturile sistemelor bazate pe cunostinte. Arhitecturi cu model simbolic, reactive, hibride, tipologie.
8. Elemente de reprezentare si prelucrare a cunostintelor. Metode: slabe, tari, motoare de inferenta.
9. Incertitudinea in sistemele bazate pe cunostinte. Perspective: antropocentrica, tehnologica, logico-matematica, logicile incertitudinii, modele de cuantificare euristica, abductia in sistemele de diagnoza.
10. Limbaje. Programarea logica sau functionala? Prolog si Lisp. Agentitate si comunicare. Ontolimbaje (KIF), comunicare interagent (KQML), programare orientata spre agent.
11. Tipuri de aplicatii ale sistemelor bazate cunostinte. Exemple: diagnoza medicala, diagnoza industriala, ingineria programarii bazata pe cunostinte (KBSE).
12. Elemente de dezvoltare a sistemelor bazate pe cunostinte. Paradigme de programare moderne, sinergia lor, calculul maleabil (soft computing).

Anul 3 semestrul I

Casian Stefan.

1997-Informatica industriala. Ingeria programarii in timp real. Ed. Tehnica

1999-Informatica industriala. Noi paradigme si aplicatii. Ed Tehnica.

1998-Principii de logica si programare logica. Metakides,G.A. Nerode Ed. Tehnica.

1999-Aplicatii JAVA inteligente pentru internet si intraneturi. Mark Watson Editura All

1993-Artificial Intelligence. Structures and Strategies for Complex Problem Solving.

Luger,G.F.,W.A. Stubblefield

si altele.

Viitoare aparitii la editura albastra:

Principiile inteligentei artificiale

Algoritmi genetici si strategii evolutive

In UNDU - Articole MSAgent.

“Cea mai cautata aplicatie pentru inteligenta artificiala, din urmatoarea decada, va fi dezvoltarea de agenti inteligenti pentru aplicatii de tipul jocurilor si pentru aplicatiile practice de procesare a informatiilor.” Watson, M.

www.markwatson.com

Despre inteligenta artificiala (AI)

Istoria cercetarii in domeniul AI incepe chiar inaintea aparitiei computerelor digitale cand cercetatorul britanic Alan Turing a dezvoltat o platforma filosofica pentru AI. Stiinta AI inspira numeroase discipline: informatica, psihologia, stiinta cognitiva, biofizica, teoria sistemelor si neurologia.

Sfera cercetarii in domeniul AI:

Translatarea automata a limbajelor naturale.

Diagnosticarea prin sisteme medicale expert.

Descoperirea de noi teoreme matematice si de noi teorii in alte domenii stiintifice.

Sisteme automate de asistenta (help desk) care raspund problemelor ridicate de clienti fara interventia umana (agenti (exemplu: Office Assistant, BunzyBudy)

Recunoasterea diferitelor obiecte vazute printr-o camera de luat vederi

Domenii in cercetarea AI: sisteme de procesare a simbolurilor si retele neurale.

“Agentii: sisteme software complexe care se adapteaza in medii dinamice si schimbatoare. Problema fundamentala consta in modul in care folosesc agentii experienta anterioara si informatiile senzoriale primite din mediu pentru a decide urmatoarele miscari, modul in care sa le abordeze si cum sa colaboreze cu alti agenti.” .Watson, M.

Cateva lucruri despre agentii de interfata (care m-ar interesa):

Se ocupa cu gasirea de informatii necesare utilizatorilor.

Au trei scopuri:

Sunt proactivi-pornire automata. Iau initiativa in determinarea metodelor de acordarea asistentei pentru utilizatori.

Din miscarile utilizatorilor invata cum sa se adapteze astfel incat sa imbunatateasca actiunile ce se repeta in timp.

Se personalizeaza automat pentru fiecare utilizator.

(Vezi BunzyBuddy: Peedy)

Creativitatea si flexibilitatea sistemelor inteligente nu va putea fi atinsa deca atunci cand sistemele vor putea sa-si creeze singure structurile de date care sa le reprezinte mediul.

Multe sisteme inteligente folosesc modele de simulare a lumii reale. Un agent creativ , flexibil, va avea capacitatea da a-si schimba modelele interne pentru a se adapta la informatiile percepute din lumea reala.

Problema dezvoltarii de interfete universale pentru agenti AI interactivi este mult mai dificila deoarece aceste interfete nu vor cere numai cunostinte despre formate, dar si despre definirea de formate de mesaje pentru alti agenti. O analogie ar fi diferenta dintre HTML si SGML (Standard Graphics Markup Language). SGML este un meta-limbaj, documentele SGML pot contine noi tipuri de tag-uri si informatii de formate de document. Cunostintele despre modul de interpretare a acestor informatii formate sunt stocate in document. HTML pe de alta parte foloseste o pura sintaxa statica.

Abilitatea agentilor de a negocia schimbari la interfata cu alti agenti necesita cunostinte despre urmatoarele concepte:

Actiunile pe care le executa agentul si ceilalti agenti.

Abilitati de a descrie, folosind un limbaj universal, interfete existente.

Abilitati de a cunoaste si explica cerintele pentru schimbarea unei interfete.

Prima generatie de agenti software au fost proiectati astfel incat sa lucreze pe un singur computer (si de obicei monoproces) si sa rezolve problemele dintr-un domeniu bine specificat. Urmatoarea generatie de sisteme inteligente va fi produsul efortului mai multor oameni aflati in locuri fizic diferite. In viitor, cea mai ieftina metoda de a dezvolta o aplicatie din domeniul AI va fi integrarea agentilor software specifici aplicatiilor dintr-o multime de surse cum ar fi mediile universitare, comerciale sau surse interne de programare. Aceste surse vor contine cod procedural si date. Clientii acestor furnizori de sisteme vor primi simpli agenti de interfata pe care il vor putea executa in propriul mediu computerizat pentru a receptiona cunostinte de la sursele distante. In acest mod clientii pot plati furnizorilor de surse de informatii o taxa de folosire care ar putea acoperi cheltuielile de dezvoltare a acestor informatii. Acest model de afacere va permite companiilor sa investeasca resursele necesare pentru dezvoltarea de componente inteligente reutilizabile.

Agentii software automatizeaza lucruri pe care, altfel, ar trebui sa le facem noi insine (un Alter Ego, sau avatar chiar- din curs). Ce anume ii deosebeste pe agentii software de alte programe software utilitare este capacitatea de a lucra in medii distribuite de prelucrare a datelor si posibilitatea de-a furniza unele cunostinte pentru automatizarea sarcinilor pentru utilizatori.

Ex: agenti cooperanti distribuiti si agenti ce ruleaza pe o masina locala si care colecteaza date si informatii de interes, in mod automat, din serviciile publice cum ar fi WWW si email-ul.

Exista o metodologie standard industriala pentru partajarea obiectelor in sistemele distribuite. OMG (Object Management Group) – consortiu nonprofit 1989. Se promoveaza teoria si practicarea tehnologiei obiectelor in medii distribuite de prelucrare a datelor. OMA (OMArchitecture) este un plan de nivel inalt pentru construirea mediilor distribuite. Cea mai importanta componenta OMA este ORB (Object Request Broker). Un ORB controleaza toate comunicatiile dintre componentele sistemului distribuit. Si, cel mai important, ORB permite diferitelor programe sa ruleze, in mediile largi distribuite, pe diferite tipuri de computere. Un ORB permite de asemenea metodelor sa se execute (RMI) in obiectele distante de date.

Agentii distribuiti sunt programe cooperante care se pot executa pe orice computer conectat in retea.

Majoritatea limbajelor de inteligenta artificiala furnizeaza un mod flexibil de manipulare a structurilor de date complexe: LISP-furnizeaza liste, Prolog-furnizeaza clauze.

Retele neurale

Utilizate in majoritate pentru recunoastere de forme.

Fiecare neuron dintr-o retea neurala are asociata o valoare numerica reprezentand o energie de activare. Retelele pot avea mai multe seturi de neuroni: intrare, hidden, iesire-acesa retea cu tri nivele este capabila sa rezolve majoritatea problemelor de recunoastere de forme si control.

Una dintre dificultatile intampinate in folosirea retelelor neurale o constituie pregatirea datelor de antrenare. Uzual datele de antrenare au nevoie de o ajustare manuala prin eliminarea cazurilor de antrenare conflictuale.

Algoritmi genetici

Constituie o metoda eficienta pentru cautarea in baze mari de date folosind principiile populatiei biologice. Spatiul de cautare este definit prin reprezentarea aleasa pentru cromozomi. Informatia stocata in cromozomi este reprezentata prin gene. In simularea algoritmilor genetici o gena este reprezentata printr-un bit de informatie (True/False). La manipularea genelor din cromozomi, folosim termenul de locus pentru specificarea indexului de localizare a unei gene specifice in cromozom. Pentru simulare de algoritmi genetici, se lucreaza cu un set de cromozomi, numiti populatie, care contin acelasi numar de gene.

Pentru gasirea unei solutii, pe baza de algoritmi genetici, a unei probleme, avem nevoie de o functie numerica de conformitate (fitness function), care ia un singur argument, un cromozom si intoarce ca rezultat o valoare numerica de conformitate. Valori mari de conformitate, indica cromozomi care reprezinta cele mai bune solutii. Un cromozom reprezinta un singur punct in spatiul de cautare al solutei problemei.

O mutatie intr-un cromozom reprezinta o comutare aleatoare a unei singure valori (True/False) reprezentand o gena.

Ex: T T F F F =>mutatie T T T F F

Incrucisare (crossover) pentru a schimba gene intre cromozomi.

Ex: TTT|FF =>incrucisare (incepand cu pozitia 2 marcata |) TTT|FF
TTT|TT TTT|FF

Articole de pe Internet

A Stronger Notion of Agency

For some researchers - particularly those working in AI - the term 'agent' has a stronger and more specific meaning than that sketched out above. These researchers generally mean an agent to be a computer system that, in addition to having the properties identified above, is either conceptualised or implemented using concepts that are more usually applied to humans. For example, it is quite common in AI to characterise an agent using mentalistic notions, such as knowledge, belief, intention, and obligation [Shoham, 1993]. Some AI researchers have gone further, and considered emotional agents [Bates, 1994][Bates et al., 1992a]. (Lest the reader suppose that this is just pointless anthropomorphism, it should be noted that there are good arguments in favour of designing and building agents in terms of human-like mental states - see section 2.) Another way of giving agents human-like attributes is to represent them visually, perhaps by using a cartoon-like graphical icon or an animated face [Maes, 1994a] - for obvious reasons, such agents are of particular importance to those interested in human-computer interfaces.

A Weak Notion of Agency

Perhaps the most general way in which the term agent is used is to denote a hardware or (more usually) software-based computer system that enjoys the following properties:

- autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language [Genesereth and Ketchpel, 1994];
- reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
- pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

A simple way of conceptualising an agent is thus as a kind of UNIX-like software process, that exhibits the properties listed above. This weak notion of agency has found currency with a surprisingly wide range of researchers. For example, in mainstream computer science, the notion of an agent as a self-contained, concurrently executing software process, that encapsulates some state and is able to communicate with other agents via message passing, is seen as a natural development of the object-based concurrent programming paradigm [[Agha et al., 1993](#)][[Agha, 1986](#)].

This weak notion of agency is also that used in the emerging discipline of agent-based software engineering:

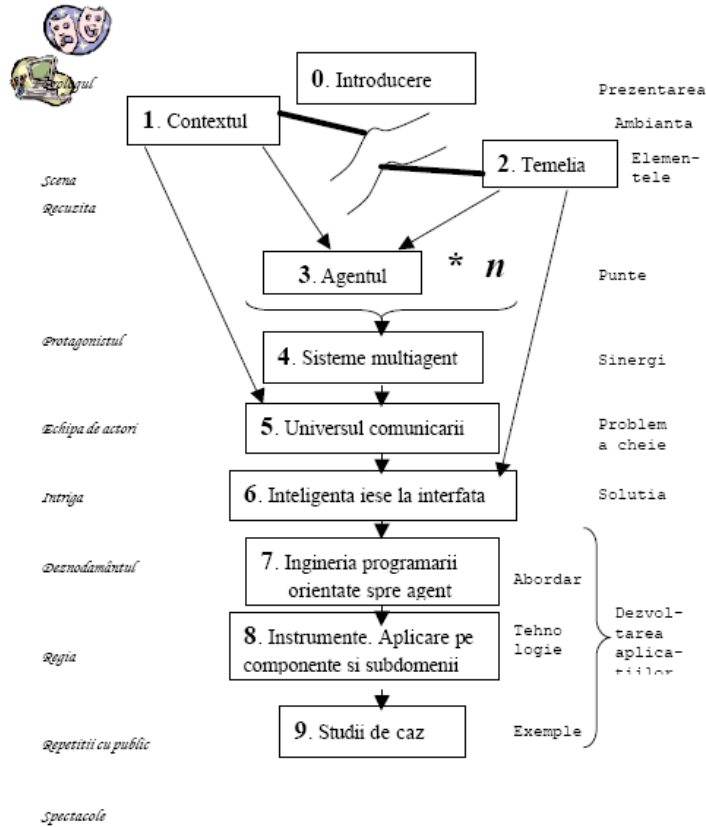
`[Agents] communicate with their peers by exchanging messages in an expressive agent communication language. While agents can be as simple as subroutines, typically they are larger entities with some sort of persistent control.' [[Genesereth and Ketchpel, 1994](#)]

A softbot (software robot) is a kind of agent:

`A softbot is an agent that interacts with a software environment by issuing commands and interpreting the environment's feedback. A softbot's effectors are commands (e.g., UNIX shell commands such as mv or compress) meant to change the external environment's state. A softbot's sensors are commands (e.g., pwd or ls in UNIX) meant to provide ... information.' [[Etzioni et al., 1994](#)]

SISTEME INTELIGENTE ORIENTATE SPRE AGENT

CUPRINS



Pe lângă contribuția de fond la cercetările prezentate în capitolul 8, colegii din Artificial Intelligence Research Group (AIRG), în frunte cu directorul ing. Ciprian Cândeș (îndeosebi Marius Stăicu, Mihai Luca și Gabriela Iozon) precum și cei din Intelligent Systems Workgroup (ISW), în frunte cu directorul ing. Ștefan Coarfa (îndeosebi Cristian Bichis, Dumitru Pipernea, Casian Ștefan) m-au ajutat cu mult entuziasm (pentru că sunt tineri) și cu multă competență (deși sunt tineri), dovedind un rar spirit de echipă.

...

AN EXCERPTS DOCUMENT ONLY.
 FROM THE AI COURSE BY B.B. and various notes by C. Ștefan.
 PERIOD CONCLUDED WITH GRADUATION PAPER IN AI AND ACCEPTED PAPERS.
 DOCUMENT TO SERVE ONLY FOR HISTORICAL PURPOSE.
 NO FURTHER USAGE.