

HY 58301104 Tietojenkäsittelytieteen historia 2 ov
Seminaariesitelmä
Tekoälyn historia

Mikko Apiola, mvapiola@cc.helsinki.fi

10. helmikuuta 2005

Sisältö

1	Tekoäly	1
2	Neuroverkot	2
3	Tekoälyn synty	3
4	Suuret odotukset	4
5	Asiantuntijajärjestelmien synty	5
	5.1 Einsteinin arvoitus prolog-kielillä	6
6	1980-luvun kehitystä	9
7	Nykypäivä	9
8	Yhteenveto	10

1 Tekoäly

Tekoälyn ensyklopedia [HKS93] määrittelee tekoälyn (artificial intelligence) tieteenalaksi, jonka tavoitteena on analysoida älykästä toimintaa ja rakentaa älykkäästi toimivia järjestelmiä. Siitä, mitä älykkyys tai älykäs toiminta tarkoittaa, on monia mielipiteitä ja näkemyksiä, ja määrittely-yritykset ovatkin osa loputonta filosofista pohdiskelua. Tekoälytutkimukseen kuuluu älykkyyden luonteen ja siihen liittyvän teorian, sekä älykkäästi toimivien järjestelmien kehittäminen. Tekoälyterminä saattaa luoda suuria odotuksia ja siihen voi liittyä mielikuvia tai pelkoja esimerkiksi "eloon heräävistä" tietokoneista, jotka ottavat vallan käsiinsä.

Tekoälytutkimus on luonteeltaan monitieteinen alue, johon olennaisesti kuuluvia tieteenaloja ovat tietojenkäsittelytiede, älyllisyyttä tutkivat tieteet, kuten kognitiotieteet ja psykologia sekä kielen ja ajattelun suhdetta sekä luonnollisen kielen ymmärtämistä tutkivat kielitieteet.

Yleinen jako etenkin kognitiotieteiden piirissä tehtävässä tutkimuksessa on jakoa ns. heikkoon ja vahvaan tekoälyyn. Vahvan tekoälyn mukaan älykkyys on "toteutusalueesta" (aivot vs. tietokone) riippumatonta, ja tekoälytutkimuksen tehtävä on löytää älykkään toiminnan yleiset lait ja toteuttaa ne tietokoneella. Heikolla tekoälyllä tarkoitetaan ohjelmia, jotka ovat "älyllisen toiminnan hyviä malleja".

Usein älykkääksi toiminnaksi mielletään sellainen toiminta, johon tietokone ei vielä kykene. Esimerkiksi vielä 60-luvulla todistettiin matemaattisesti, ettei tietokone tule ikinä voittamaan ihmistä shakkipelissä. Tuolloin shakkia pelaavaa konetta olisi varmasti pidetty hyvinkin älykkäänä, mutta nykypäivänä se saatetaan mieltää vain mekaaniseksi toiminnaksi, johon "tyhmä tietokonekin kykenee".

Eräs kanta tietokoneiden älykkyyttä pohdiskelevassa filosofisessa keskustelussa on John Searlen paljon kritisoitu Kiinalaisen huoneen argumentti [Sea80]. Argumentti on seuraavanlainen: on olemassa huone, joka sisältää asioista mitään ymmärtämättömän ihmisen sekä suuren joukon "ohjekirjoja". Huoneen ulkopuolelta syötetään kysymyksiä huoneessa olevalle henkilölle, joka etsii kysymyksiin vastauksen ohjekirjoista. Argumentin sanoma on, että vaikka huoneen ulkopuolta katsoen toiminta on älykästä, niin huoneessa oleva ihminen ei ole älykäs. Tähän vedoten Searle toteaa, että myöskään tietokone ei voi ymmärtää käsittelemistään sisällöistä mitään.

Ensimmäisen kokonaisvision tekoälystä loi Alan Turing artikkelissaan *Computing Machinery and Intelligence* [Tur50], jossa hän esitteli Turingin testin, koneoppimisen, geneettiset algoritmit sekä vahvistusoppimisen (reinforcement learning). Turingin testi on ollut merkittävä tekoälyn filosofista pohdintaa innoittanut ajatusleikki. Sen mukaan mikä tahansa järjestelmä, joka “näyttää älykkäältä” on älykäs. Testi voidaan järjestää esimerkiksi siten, että ihminen keskustelee tietokoneen kanssa. Järjestelmä voidaan katsoa älykkääksi, jos ihminen ei havaitse, että keskustelukumppani on tietokone. Turingin testin läpäisemiseksi ei tähän päivään mennessä olla kovinkaan pitkällä, joskin hyviä yrityksiä on paljon.

Tekoälyn taustat ovat filosofiassa, matematiikassa, taloustieteissä, neurotieteissä (miten aivot käsittelevät tietoa), kognitiotieteissä (miten ajattelu toimii), tietokonetekniikassa (miten rakennetaan tehokas tietokone), kielitieteessä (kielen ja ajattelun suhde sekä luonnollisen kielen ymmärtäminen). Historiallisesti tekoälyn voidaan jakaa esihistoriaan, joka käsittää ajan ennen tietokoneiden keksimistä, sekä moderniin historiaan, joka käsittää ajan tietokoneiden keksimisestä nykypäivään (noin vuodesta 1943 nykypäivään saakka. Tässä esitelmässä keskitytään lähinnä tekoälyn moderniin historiaan.

2 Neuroverkot

Ensimmäinen yleisesti tekoälyksi tunnistettu työ on Warren McCulloch:n ja Walter Pitts:n 1943 neuroverkkoihin liittyvä työ [RN03]. Tietosanakirja¹ määrittelee neuroverkot seuraavasti: “Neuroverkot ovat informaation käsittelyn matematiikan tai laskennan malleja, jotka perustuvat yhdistävään laskentaan. Niiden toiminta perustuu fyysisten aivojen neuronien verkottuneen rakenteen jäljittelyyn, eli neuroverkot pyrkivät tekemään tietojen käsittelyä kuten aivojen neuronit”.

Neuroverkoissa ideana onkin pyrkiä jäljittelemään ihmisen aivojen toimintaa. Neuroverkko koostuu tyypillisesti joukosta hermosoluja simuloivia solmuja, joiden liitokset simuloivat hermosoluja liittäviä synapseja. Tietoa käsitellään verkossa liitoksien välityksellä aktivaation leviämisenä solulta toiselle. Neuroverkko oppii sille annettujen esimerkkien perusteella, eli sitä ei suoraan ohjelmoida vastaamaan tiettyllä tavalla [AHJ96]. Neuroverkkotutkimus on luonteeltaan vahvaa tekoälyä, eli

¹<http://fi.wikipedia.org/>

siinä pyritään nimenomaan löytämään älykkään toiminnan yleiset lait ja toteuttamaan ne tietokoneella. Neuroverkot eivät alkuaikoinaan saavuttaneet suurtakaan suosiota.

Vuonna 1950 Turing julkaisi merkittävän artikkelinsa *Computing Machinery and Intelligence* [Tur50].

3 Tekoälyn synty

Vuonna 1956 John McCarthy järjesti Dathmouthissa kesäkoulun kaikille lasketavuuden teoriasta, neuraaliverkoista ja älykkydestä kiinnostuneille tutkijoille. Kouluun osallistuivat henkilöitä, kuten Trenchard More, Arthur Samuel (IBM) sekä Ray Solomonoff sekä Oliver Selfridge (MIT). Työryhmän toiminnasta ei sinänsä koitunut mitään mullistavaa, paitsi että he tulivat keksineeksi nimen tieteenalalle: tekoäly.

Noihin aikoihin myös kaksi Carnage Tech:n (nykyinen Carnegie Mellon University) tutkijaa Allen Newell ja Herbert Simon esittelivät laatimansa päättelykoneen *The Logic Theorist*. Monet pitävät *Logic Theoristia* ensimmäisenä tekoälyä käyttävänä järjestelmänä.

Newell:n ja Simonin kehittivät edelleen *General Problem Solver (GPS)* niminen järjestelmä. Toisin kuin *Logic Theorist*, se oli suunniteltu imitoimaan ihmisen ongelmanratkaisuprosessia. GPS:n ajateltiin olevan ensimmäinen ohjelma, joka kykeni "ajattelemaan ihmismäisesti".

Vuonna 1959 Herbert Gelernter laati *The Geometry Theorem Prover*-nimisen järjestelmän, joka kykeni todistamaan joidenkin opiskelijoiden mielestä vaikeita matemaattisia teoreemoja. Vuodesta 1952 lähtien Arthur Samuel kirjoitti joukon tammipeliä pelaavia oppivia ohjelmia. Ohjelmia demonstroitiin televisiossa helmikuussa 1956, ja ne tekivät yleisöön suuren vaikutuksen [RN03].

Vuonna 1958 McCarthy määrittely korkean tason kielen LISP ², josta tuli dominoiva tekoälykieli. LISP (*LISt Processing language*) on omalaatuinen, rekursiivisiin funktiomäärittelyihin perustuva ohjelmointikieli. Kielen puhtaassa muodossa

²<http://www.lisp.org>

ei ole muuttujia tai sijoitusoperaatioita, vaan kaikki ohjelman tilannetieto sisältyy sen tekemien sisäkkäisten funktiokutsujen argumentteihin. LISP on suhteellisen helppo käyttää, ja se on tehokas työkalu monimutkaisten ongelmien ratkaisemisessa. LISP on nykypäivänäkin tekoälyohjelmoinnin ykköskieli. Esimerkiksi EMACS-editori on kirjoitettu LISP:llä, ja jopa sisältää LISP-tulkin [Orp98]. McCarthyta pidetään yleisesti tekoälyn isänä.

4 Suuret odotukset

1950-luvulla tietokoneet ja ohjelmointivälineet olivat kovin alkeellisia. Ei ollut pitkä aika siitä, kun tietokoneiden ajateltiin olevan kyvykkäitä vain ja ainoastaan aritmeettisiin laskutoimituksiin. Noina aikoina pidettiin hyvin ihmeellisenä, jos tietokone osasi mitään vähänkin älykkäältä vaikuttavaa. Muutamien yksinkertaisten järjestelmien, kuten Logic Theorist:n ja GPS:n parissa tehtyjen onnistuneiden kokeilujen perusteella tekoälyn tulevaisuus nähtiin hyvin optimistiseksi. Tekoälytutkijat eivät ujostelleet tehdessään tulevaisuuden saavutuksia koskevia oletuksia. Seuraava Herbert Simonin 1957 tekemä lausunto on usein siteerattu:

Tarkoitukseni ei ole yllättää tai järkyttää teitä - haluan vain sanoa, että tällä hetkellä maailmassa on olemassa koneita jotka ajattelevat, oppivat ja luovat uutta. Lisäksi näiden koneiden kyky ajatella, oppia ja luoda tulee kasvamaan huimaa vauhtia kunnes - näkyvässä olevassa tulevaisuudessa - niiden ratkaisemien ongelmien joukko tulee olemaan verrattavissa siihen ongelmien joukkoon, joiden ratkaisemiseen nyt käytetään ihmisen mieltä". - Herbert Simon 1957.

Simon ennusti myös, että 10 vuoden sisällä tietokone olisi shakkimestari, ja että tietokone todistaisi jonkin merkittävän matemaattisen teoreeman. Simonin ylioptimistisuus johtui tuon ajan järjestelmien lupaavasta suoriutumisesta yksinkertaisissa esimerkkitapauksissa.

Kuitenkin nämä järjestelmät osoittautuivat käyttökelvottomiksi yritettäessä soveltaa niitä laajempaan ja vaikeampien ongelmien joukkoon. Ongelmat johtuivat siitä, että tuolloin laadituilla ohjelmilla ei ollut käytössään riittävästi aihepiiriin liittyvää tietämystä. Esimerkiksi kielenkääntämisen suhteen oli ollut suuria odotuksia esim. U.S. National Research Council:n rahoittamassa projektissa, jossa ta-

voitteena oli pystyä automaattisesti kääntämään venäjänkielistä tieteellistä tekstiä. Myöhemmin havaittu yleinen totuus on, että kääntäminen vaatii yleistä alueeseen liittyvää arkitietämystä, jonka opettaminen tietokoneelle on osoittautunut hyvin vaikeaksi. Seuraavassa on kielenkääntämisen epäonnistumisia havainnoivia englanti-venäjä-englanti -uudelleenkäännöksiä:

The Spirit is willing, but the Flesh is weak \Rightarrow *The vodka is good but the Meat is rotten.* [RN03]

Out of sight, out of mind. \Rightarrow *Blind, Mad.*

Kielenkääntäminen ei tähän päivään mennessä ole kokenut merkittävää edistymistä. Seuraava esimerkki on BabelFish-käännöskoneella³ 2.2.2005 tuotettu englanti-espanja-englanti -käännös.

The spirit is willing, but the Flesh is weak. \Rightarrow *The alcohol is arranged, but the meat is weak.*

Toiseksi ongelmat johtuivat ratkottujen ongelmien vaikeudesta. Aiemmat järjestelmät olivat toimineet, koska niiden käsittelemä joukko tietämystä oli hyvin rajallinen. Tuolloin uskottiin yleisesti, että järjestelmien skaalaamisessa (scaling up) ratkaisemaan laajempia ongelmia oli yksinkertaisesti kysymys tehokkaamman laitteiston ja isomman muistikapasiteetin hankinnasta.

Vuonna 1966 ilmestyi arviointikomitean raportti, jossa todettiin että minkäänlaista edistystä kääntämisen suhteen ei ollut tapahtunut. Tämän seurauksena kaikki yhdysvaltojen hallituksen myöntämä rahoitus käännösprojekteille katkaistiin. Nykyään konekääntämistä käytetään, vaikkakin se on monin osin puutteellista [RN03]. Myös esimerkiksi Brittihallitus katkaisi rahoituksen lähes kaikkiin tekoälyä tutkiviin yliopistoihin.

5 Asiantuntijajärjestelmien synty

Epäonnistumisten kautta tekoälyn mahdollisuudet alkoivat hahmottua. Aiemmat järjestelmät toimivat ainoastaan hyvin yksinkertaisissa ympäristöissä. Ryhdyttiin tekemään kokeiluja, joissa pyrkimyksenä oli tehdä järjestelmiä ratkaisemaan tiet-

³<http://babelfish.altavista.com>

tyjä alakohtaisia ongelmia siten, että järjestelmälle opetettiin alakohtaista tietämystä.

DENDRAL- ohjelma vuonna 1969 oli esimerkki tästä lähestymistavasta. DENDRAL kehitettiin Stanfordissa, jossa Ed Feigenbaum, Bruce Buchanan, Joshua Lederberg kokoontuivat yhteen ratkaisemaan molekyyli-rakenteisiin liittyvää ongelmaa. DENDRAL oli ensimmäinen onnistunut järjestelmä, joka käytti hyväkseen laajaa alakohtaista sääntöjen joukkoa.

Myös lääketieteellisten diagnoosien tekemiseen tarkoitettuja järjestelmiä, kuten MYCIN kehitettiin. MYCIN:n tietämuskanta sisälsi noin 450 sääntöä, ja sen toimintakyky oli joidenkin arvioiden mukaan ihmislääkärin tasolla.

Alakohtaisen tietämyksen tärkeys ole ilmeinen myös luonnollisen kielen tunnistuksessa. Winograd:n SHRDLU-järjestelmä kykeni selviytymään sanojen moniselitteisyydestä ja ymmärtämään viittauksia pronomineihin, mutta tämä johtui pääasiassa siitä, että se oli suunniteltu toimimaan ainoastaan hyvin rajatussa ympäristössä.

Nämä niinsanotut asiantuntijajärjestelmät saavuttivat laajan suosion. Suuri joukko erilaisia sääntö- ja päättelykieliä kehitettiin. Jotkut näistä perustuivat logiikkaan. Esimerkiksi Prolog-kieli (PROgramming with LOGic) tuli suosituksi Euroopassa, ja PLANNER-perhe yhdysvalloissa.

5.1 Einsteinin arvoitus prolog-kielillä

Prolog on logiikkaohjelmointikieli, joka sopii erityisesti symboliseen ja ei-numeeriseen laskentaan. Se on yleisesti käytetty kieli tekoälysovelluksissa, joissa käsitellään symboleja ja tehdään niiden perusteella päättelyä. Prolog-ohjelma koostuu joukosta faktoja ja sääntöjä (facts and rules). Ohjelma suoritetaan siten, että tietämuskantaan esitetään kysely (query), ja katsotaan voidaanko se ratkaista tietämuskannassa olevien faktatietojen ja sääntöjen perusteella.

Albert Einstein kirjoitti seuraavan arvoituksen 1900-luvun alkupuolella ja arveli, että 98% maapallon väestöstä ei kykene sitä ratkaisemaan.

Albert Einsteinin arvoitus:

1. Samalla kadulla on 5 taloa. Jokainen talo on eri värinen. Kunkin talon omistaja on eri kansallisuutta.
2. Omistajista kullakin on eri lempijuoma, kukin polttaa erimerkkisiä savukkeita ja omistaa eri lemmikkieläimen.
3. Kenelläkään ei siis ole samaa lemmikkiä, kukaan ei juo samaa juomaa, polta samaa savukemerkkiä eikä ole samaa kansallisuutta kuin toinen.

Kysymys kuuluu: “Kenen lemmikkieläin on kala?”

VINKIT

1. Britti asuu punaisessa talossa.
2. Ruotsalaisen lemmikki on koira.
3. Tanskalainen juo teetä.
4. Vihreä talo on valkoisen talon vasemmalla puolella (vieressä).
5. Vihreän talon omistaja juo kahvia.
6. Henkilö, joka polttaa Pall Mallia kasvattaa lintuja.
7. Keltaisen talon omistaja polttaa Dunhill:a.
8. Keskimmäisessä talossa asuva juo maitoa.
9. Norjalainen asuu ensimmäisessä talossa.
10. Henkilö joka polttaa Blendiä asuu kissan omistajan naapurissa.
11. Henkilö, jonka lemmikki on Hevonen, asuu Dunhillia polttavan naapurissa.
12. Henkilö, joka polttaa Blue Masteria juo olutta.
13. Saksalainen polttaa Princea.
14. Norjalainen asuu sinisen talon naapurissa.
15. Henkilöllä, joka polttaa Blendiä on naapuri joka juo vettä.

Seuraavassa (kaava 1) on esitetty Prolog-tietämuskanta, joka muotoilee edellisen Einsteinin ongelman.

```
onvieressa(X, Y, List) :- onoikealla(X, Y, List).
```

```
onvieressa(X, Y, List) :- onoikealla(Y, X, List).
```

```
onoikealla(L, R, [L | [R | _]]).
```

```
onoikealla(L, R, [_ | Rest]) :- onoikealla(L, R, Rest).
```

```
einstein(Talot, KalanOmistaja) :-
```

```
=(Talot, [[talo, norjalainen, _, _, _, _], _, [talo, _, _, _, maito, _], _, _]),
member([talo, britti, _, _, _, punainen], Talot),
member([talo, ruotsalainen, koira, _, _, _], Talot),
member([talo, tanskalainen, _, _, tee, _], Talot),
onoikealla([talo, _, _, _, vihrea], [talo, _, _, _, _, valkoinen], Talot),
member([talo, _, _, _, kahvi, vihrea], Talot),
member([talo, _, lintu, pallmall, _, _], Talot),
member([talo, _, _, dunhill, _, keltainen], Talot),
onvieressa([talo, _, _, dunhill, _, _], [talo, _, hevonen, _, _, _], Talot),
member([talo, _, _, _, maito, _], Talot),
onvieressa([talo, _, _, blend, _, _], [talo, _, kissa, _, _, _], Talot),
onvieressa([talo, _, _, blend, _, _], [talo, _, _, _, vesi, _], Talot),
member([talo, _, _, bluemaster, kalja, _], Talot),
member([talo, saksalainen, _, prince, _, _], Talot),
onvieressa([talo, norjalainen, _, _, _, _], [talo, _, _, _, _, sininen], Talot),
member([talo, KalanOmistaja, kala, _, _, _], Talot).
```

Kaava 1: Prolog tietämuskanta

Seuraavassa (kaava 2) tehdään Prologille kyseiseen tietämuskantaan kysely, jonka vastauksena saadaan talojen oikea järjestys, sekä kalanomistajan kansalaisuus.

```
?- einstein(Talot,KalanOmistaja).
```

```
Talot = [[talo, norjalainen, kissa, dunhill, vesi, keltainen],
[talo, tanskalainen, hevonen, marlboro, tee, sininen],
[talo, britti, lintu, pallmall, maito, punainen],
[talo, saksalainen, kala, rothmans, kahvi|...],
[talo, ruotsalainen, koira, winfield|...]]
```

```
KalanOmistaja = saksalainen
```

Kaava 2: Kysely Prolog-tietämuskantaan

Tämän esimerkin oli tarkoitus demonstroida Prolog-kielen toimintaa ja logiikkaohjelmoinnin vahvuuksia verrattuna perinteisiin ohjelmointikieliin. Tarkoituksena oli myös antaa konkreettinen esimerkki asiantuntijajärjestelmä-tyyppisestä tekoälysovelluksesta.

6 1980-luvun kehitystä

Ensimmäinen kaupallinen asiantuntijajärjestelmä R1 aloitti toimintansa Digital Equipment Corporationilla [RN03]. Järjestelmä auttoi uusien tietokonetilausten käsittelyssä. Paljon muita järjestelmiä kehitettiin esim. DEC:llä.

Vuonna 1981 Japanissa käynnistettiin “Fifth Generation” projekti, jonka tavoite oli rakentaa älykkäitä järjestelmiä, jotka toimisivat Prolog-pohjaisesti. Vastauksena tähän Yhdysvalloissa käynnistyi Microelectronics ja Computer Technology Corporation (MCC). Molempien tavoitteena oli rakentaa tekoälypohjaisia tietokonejärjestelmiä. Kuitenkaan kummankaan yhtiön koneet eivät saavuttaneet menestystä.

Yleisesti tekoälyteollisuus kukoisti, mutta pian tämän jälkeen tuli tekoälyn takatalveksi (AI winter) kutsuttu ajanjakso, jossa monet yritykset kärsivät tappioita, koska eivät pystyneet toteuttamaan huimia lupauksiaan.

1970-luvulta lähtien tietojenkäsittelytieteen alalla ei juuri tutkittu neuroverkkoja, mutta muilla tieteenaloilla kylläkin. Varsinainen sysäys neuroverkkojen paluulle lähti, kun ainakin neljä eri ryhmää uudelleenkeksi takaisinkytkentäalgoritmin (back-propagation). Algoritmia sovellettiin moniin oppimisongelmiin tietojenkäsittelytieteessä ja psykologiassa, ja se saavutti laajan suosion.

7 Nykypäivä

Nykyään monilla aloilla tehdään tekoälyyn liittyvää tutkimusta. Esimerkkejä toimivista järjestelmistä ovat NASAN avaruudessa toimiva suunnitteluohjelma, joka suunnitteli avaruusaluksen toimintojen suoritusjärjestystä maasta lähetettävien ohjeiden perusteella [RN03]. Peleistä mainittakoon IBM:n Deep Blue [DEEPB], jonka voitto maailmanmestari Gasparovista shakissa oli eräänlainen “tekoälyn unel-

man” toteutuminen. Voiton ansiosta IBM:n osakkeiden arvo nousi USD 18 biljoonaa. Esimerkki neuroverkkoja hyödyntävästä järjestelmästä on ALVINN-järjestelmä [Pom89], jonka saavutuksena on kyseisellä järjestelmällä varustettu auto, joka kykeni ajamaan USA:n halki 2850 mailia ilman kuljettajaa - tarkemmin ottaen 98% ajasta. Lääketieteellisten diagnoosien tekemiseen tarkoitettuja järjestelmiä ovat myös yleistyneet.

Persianlahden kriisin aikoihin vuonna 1991 USA:lainen DART-järjestelmä teki automaattisesti tehokkaat logistiikkasuunnitelmat ajoneuvojen, lastin, ihmisten yms. siirtämiseksi [RN03]. Tekoälyä hyödynnetään myös robotiikan alueella leikkauksissa, sekä erilaisissa automaattitehtaissa. Myös pienempiä viihdekäyttöön, kuten esimerkiksi ristisanatehtävien ratkaisuun tarkoitettuja järjestelmiä on kehitetty jonkin verran.

8 Yhteenveto

Tekoäly on monitieteinen tieteenala, joka tutkii älykkääksi katsottua toimintaa, ja tuottaa älykkäitä tietojärjestelmiä. Tekoälyn moderni historia voidaan katsoa alkaneeksi samoihin aikoihin, kuin tietokone on keksitty. Tekoälyn alkuaikojen luonnehtii kokeilut neuroverkkotutkimuksen ja erilaisten ongelmanratkaisuohjelmien, kuten Logic Theorist ja GPS parissa, sekä ylioptimistiset visiot esim. ihmisen kaltaiseen älykkyyteen, tai sujuvaan kielenkääntämiseen piankin kykenevistä koneista.

Järjestelmät eivät kuitenkaan skaalautuneet ratkaisemaan isompia ongelmia, ja vakavia ongelmia havaittiin esimerkiksi kielenkääntämisen alalla. Tekoälyn mahdollisuudet alkoivatkin tarkentua, ja monilla tahoilla laadittiin onnistuneita asian tuntijajärjestelmä-tyyppisiä jonkin kapean sovellusalueen ongelmia ratkovia järjestelmiä. Esimerkkejä näistä ovat erilaiset lääketieteelliset diagnoosijärjestelmät.

Esimerkki pelistä, jossa tietokone ei vielä kykene antamaan ihmiselle kunnollista vastusta on GO-peli. Nähtäväksi jää, mihin tietokoneet tulevaisuudessa kykenevät.

Viitteet

- [AHJ96] Alander, J., Honkela, T., Jakobsson, M., Neuroverkot: Johdatus moderniin tekoalyyn. *Proceedings of STeP 96*, Publications of the Finnish Artificial Intelligence Society, sivut 1–3, 1996. URL: <http://www.uwasa.fi/stes/step96/step96/honkela2/> (2.2.2005).
- [DEEPB] IBM Research: Deep Blue-project. URL: <http://www.research.ibm.com/deepblue/> (10.2.2005).
- [HKS93] Hyvonen, E., Karanta, I., Syrjanen, M., Tekoälyn ensyklopedia. ISBN: 951-662-559-2, Gaudeamus 1993.
- [Orp98] Orponen, P. Tietotekniikan perusteet, luentokalvot: Tekoaly. Teknillinen korkeakoulu. 1998. URL: <http://www.tcs.hut.fi/orponen/opetus/ttp/981124/luento/index.htm> (10.2.2005).
- [Pom89] Pomerlau, D., ALVINN: An autonomous Land Vehicle In a Neural network: Advances in Neural Information Processing Systems 1, Morgan Kaufmann, 1989.
- [RN03] Russell, S., Norvig, P., Artificial Intelligence: A Modern Approach, 2. painos, ISBN: 0-13-080302-2, Prentice-Hall, 2003.
- [Sea80] Searle, J., Mind, Brains and Programs. *Behavioral and Brain Sciences* 3, sivut 417-424, 1980.
- [Tur50] Turing, A. M., Computing machinery and Intelligence, *Mind* 59, sivut 443–460, 1950. URL: <http://www.loebner.net/Prizef/TuringArticle.html> (2.2.2005).