

Tarnóczy Tibor

Szoftverágensek és a döntéstámogatás

A döntéstámogató és gazdaságmodellező számítógépes rendszerekben egyre nagyobb szerep jut a mesterségesintelligencia-kutatás eredményeinek, így a szoftverágenseknek is, amelyek alkalmazása ma már az információtechnológia szinte minden területén megtalálható. Az ágens olyan autonóm „részszoftver”, ami érzékelői által érzékeli környezetét és beavatkozó szervei által megváltoztatja azt, képes alkalmazkodni a környezetéhez, képes tanulni cselekedetei hatásából, illetve a környezet változásaiból, és e tudást felhasználva képes változtatni tervein, cselekedetein annak érdekében, hogy azok megfelelőbbek legyenek. Az ágensek alkalmazása átütő eredményt hozhat a gazdasági modellezésben is. A tanulmány bemutatja a döntéshozatal azon jellemzőit, amelyek fontosak az ágensek megítélésében, valamint a szoftver ágensek főbb jellemzőit és gazdasági alkalmazási lehetőségeit.

A racionalitás hosszú időn keresztül a kulcsfeltételezések egyike volt a közgazdasági elmélet modelljeiben, amelynek az volt a lényege, hogy az emberek, ha az adott gazdasági körülmények változatlanok maradnak, a legjobb tevékenységeket választják. A racionalitás lényege megtalálni az emberi viselkedés racionális feltételezésének és a valóságos viselkedésének a legjobb kombinációját. A gazdasági cselekvéssel kapcsolatban okkal vetődik fel a kérdés: Mi tekinthető tulajdonképpen racionálisnak? Lehetséges-e, hogy a döntéshozók racionálisan válasszanak?

Az egyik lehetséges definíció *Max Weber (1967)* cselekvéseméleti kategóriáibanól meríthető: az emberi viselkedés csak abban az esetben minősül cselekvésnek, ha a cselekvő szándékolt értelmet kapcsol hozzá. Társadalmi cselekvés esetében viszont a szándékolt értelmén túl megkívánja, hogy mások viselkedésére legyen vonatkoztatva, és menetében mások viselkedéséhez igazodjék. Amennyiben ezek a kritériumok teljesülnek, mind az egyéni szintű, mind a társas cselekvés kielégítheti a racionalitás követelményét.

Amikor a gazdasági-üzleti élettel foglalkozunk, akkor a gazdasági irányultságú társadalmi cselekvés, a gazdasági cselekvés foglal el központi helyet. *Weber (1967:81)* szerint „gazdasághoz igazodónak kell nevezni a cselekvést, amennyiben szándékolt értelmé szerint arra irányul, hogy bizonyos hasznos szolgálatok iránti kereslet kielégítéséről gondoskodjék”. Más szerzők a célracionális cselekvést mintegy a gazdasági cselekvéssel azonosítják (*Bertalan 2001*). Még egyértelműbb a célracionális cselekvésnek a gazdasági cselekvéssel való megfeleltetése, ha az önérdeket tekintjük vonatkoztatási pontnak. Ebben az esetben a gazdasági cselekvés célracionális számításon alapszik, amelyet a megismerési folyamat szempontjából az érdekek, a szabályok, a törvények, az előírások mint intézményszerűült szociokulturális cselekvési minták határolják be. (*Bertalan 2001*)

Simon (1955) szerint a racionális választás különböző modelljei elsődlegesen abból a speciális megközelítésből származnak, hogy be kell vezetni bizonyos korlátokat (feltételeket), amelyek keretében a racionális megközelítés végbemehet. Ismert, hogy létezik pozitív

és normatív közgazdasági elmélet. *Simon* munkái alapján (1955; 1960; 1979; 1986; 1991) úgy tűnik, hogy ő főleg a pozitív elméletet alkalmazza. Ugyanakkor nem mindig tiszta, hogy egy bizonyos modellt az eredményezi, ami az ágens választása vagy amit választania kellene. *Simon* a probléma megoldását a választás emberi mechanizmusának részletes tanulmányozása által próbálta megkeresni. Okfejtésében a számítógép-programozásban az emberi gondolkodást követő algoritmushoz nagyon hasonlóval találkozhatunk.

March (2000) szerint a racionális választás elmélete azt feltételezi, hogy a döntési folyamatok következtetések és preferenciákra épülnek. Következtetések abban az értelemben, hogy a cselekvés függ a folyamatban lévő cselekvés jövőbeli hatásainak előrejelzéseitől, és preferenciákra épülnek abban az értelemben, ahogyan a következményeket a személyes preferenciák tükrében értékeli. Racionálisnak azt az eljárást tartja, amely a következményeket logikusan értékeli. A standard racionális modell jellemzője az egyik szituációból a másik szituációba történő elmozdulás, és ezáltal a modell előrejelzési lehetőségének a kihasználása. Ugyanakkor a szituáció változása, az információ korlátozottsága és feldolgozhatósága szükségessé teheti az elérendő szint megváltoztatását, és ha ilyen körülmények között a racionális szabályoktól eltérünk, hibát követhetünk el.

A döntéshozatal racionális modelljét követők felismerték a racionalitás feltevéseinek széles körű használatát és az ilyen modellek sikerességét. Hamar észrevették az elfogadottság és a hasznosság jeleit, ami hatásos támogatást adott ezeknek a modelleknek. A racionális választás tiszta elméletének leggyakoribb és legjobban megalapozott változata az, amely felismeri a jelenbeli tevékenységek jövőbeli következményeivel kapcsolatos bizonytalanságot. Azt feltételezik, hogy a döntéshozók a várható következmények alapján választanak az alternatívák között, de a következményeket nem ismerik teljes biztonsággal (*March* 2000).

A haszonmaximalizálásra, illetőleg az optimalizálásra tekintettel a racionalitás fogalma módosul: a gyakorlatban nehéz, szinte lehetetlen a cél eléréséhez a legjobb eszközt meghatározni, illetőleg a cselekvő – ahhoz, hogy viselkedése a szó szűkebb értelmében racionális legyen – általában nem fér minden információhoz hozzá, ezért elfogadható kompromisszumot kell kötnie. A boudoni cselekvés-racionalitás (*Boudon* 2003) Weberre visszavezethető viselkedési elemeket is tartalmaz, különösen a szubjektív racionalitás, illetőleg az a priori feltételek bevezetése által, s ez összekötő kapcsolatot képez egyrészt a szubjektivitás és objektivitás, másrészt a viselkedést meghatározó „szándékolt értelem” és normakövető magatartás között.

Megállapítható, hogy a racionális magatartás négy elemből tevődik össze: a preferenciákból (a célokból), a meggyőződésekből, a lehetőségekből és a tevékenységekből. A preferenciák a szereplő céljait és az azok közötti rangsort tartalmazzák. Racionálisan csak az viselkedhet, akinek következtetés célrendszere van, azaz tudja, mit akar elérni. A meggyőzések a cselekvések várható eredményeinek ismeretét jelenti: olyan hipotézisek, amelyek az egyes cselekvések várható kimenetére vonatkoznak.

A racionális magatartás csak akkor valósulhat meg, ha ismerik az egyes döntések várható következményeit is. A lehetőségek a célok megvalósításához rendelkezésre álló erőforrásokat és egyéb korlátozó tényezőket foglalják magukba. A racionális magatartás végül is a döntésben és az annak alapján megvalósuló választásban testesül meg: a preferenciák, a meggyőzések és a lehetőségek alapján a racionális egyén kiválasztja a számára legkedvezőbb „alternatívát és azt fogja megvalósítani.

A racionalitás kizárólag az ágens külső tevékenységei alapján ítéltethető meg. Amikor korlátozott erőforrásokkal rendelkezünk, nem tudjuk reálisan megítélni, hogy ha az ágens hibázik a legjobb lépés megtételében, az irracionális cselekvés-e. Ebből következően a racionalitásnak egyfajta „gyengébb” változatára van szükségünk, amely az ágens nemcsak az általa kiválasztott tevékenység, hanem az általa használt következtetés és az elérhető erőforrások által ítéli meg (Parkes 2001).

Jones (1999) azt írja, hogy a racionális döntéshozó viselkedését teljes mértékben a feladatkörnyezet határozza meg. Ez azt jelenti, hogy ha ismerjük a döntéshozó környezetét és céljait, akkor levezethetjük a döntéshozó tevékenységeit is. Ugyanakkor ha a döntéshozó racionális akar lenni, de hibázhat, akkor ismernünk kell a döntéshozó kognitív és emocionális felépítését. A döntéshozatal ezen koncepciója két fontos hipotézishez vezet:

- 1) A viszonylag fix feladatkörnyezetben a szereplők viselkedése két egymást kölcsönösen kizáró kategóriára osztható: racionális célélérés és korlátozott racionalitás.
- 2) Bizonytalan, kétes kimenetelű vagy ellentmondó feladatkörnyezetben a viselkedés a célok, a feldolgozási korlátok és a döntéshozó problémateréje és a feladatkörnyezet közötti kapcsolat függvénye. Ebben a nagyon komplex helyzetben a problémateréje beállítások nemlineáris kapcsolatban lehetnek a célokkal és a feldolgozási korlátokkal.

Az előzőek azt jelentik, hogy egy viszonylag fix feladatkörnyezetben megkülönböztethetünk adaptív, célorientált döntéshozói viselkedést, és olyan viselkedést, amelyik a feldolgozási korlát következménye, amikor is lehetőségünk lehet az eltérés mérésére is. Ez nagyon fontos lehet az ágens „teljesítményének” a megítéléséhez, fontos szerepet játszhat az ágens „tanulásában”.

A gyakorlati döntéshozatallal foglalkozó kutatások (Simon 1979)¹ azt sugallják, hogy az adott időpontban nem minden alternatíva ismert, nem minden következményt lehet figyelembe venni, és nem minden preferencia egyértelmű (korlátozott racionalitás). Úgy tűnik, hogy a valós-világi döntéshozatallal alkalmazott szabályok különböznek a döntéshozatallal foglalkozó elméletekben találhatóaktól. A döntéshozók megelégszenek egy „elég jó” (kielégítő) cselekvéssel a „lehetséges legjobb” helyett. (March 2000) Ehhez kapcsolódóan Simon két fontos – a mai döntéstámogató rendszerek szempontjából is komoly jelentőséggel bíró – fogalmat vezetett be: a *kielégítő megoldást* és az *alkalmazkodó célszintet*. Simon szerint a tudás és az elemzési lehetőség elkerülhetetlen korlátai arra kényszerítik az embereket, hogy az első lehetőséget válasszák, amelyik elfogadhatóan jó nekik. A teljesen racionális megoldás esetén azt feltételezzük, hogy ismerjük az összes lehetséges választási lehetőséget, és egyszerűen ki kell választani az adott szituációban a legjobbat. A kielégítő megoldás elve pedig azon a feltételezésen alapszik, hogy a szituáción belül rendelkezünk valamilyen minimális célszinttel, és az alternatívák közötti keresés ezen szint elérésekor fog megállni. A célszint dinamikus, és azt a mindenkori körülményekhez igazítjuk.

A kielégítő megoldás nem feltétlenül jelent optimális megoldást is. Az optimális megoldás keresésekor azt feltételezzük, hogy azt a lehetőséget választjuk, amelynél a maximális hasznosságot tudjuk elérni. Korlátozott tudás és információ rendelkezésre állása

¹ Simon azt írja, hogy a korlátozott racionalitás a klasszikus mindenttudó racionalitás alternatívájaként került megalkotásra, és számtalan gyakorlati tanulmány mutatta be, hogy a korlátozott racionalitás, szemben a teljes racionalitással, nagymértékben konform az üzleti döntéshozatallal. Bevezette az adminisztratív ember (administrative man) fogalmát, amit a korlátozottan racionális ágens szinonímájaként használt.

esetén ez általában nem lehetséges, de ilyen körülmények között is nagy valószínűséggel is kiválaszthatunk egy szuboptimális megoldást, amely a megadott feltételnek megfelel (Conlisk 1996).

Szubjektívnek nevezi Boudon (2003) a racionalitást, ha a cselekvő mindent mozgósít, ami lehetővé teszi számára, hogy értelmet adjon annak a helyzetnek, amelybe került, célokat tűzzön maga elé és megválassza az eszközöket. Számos esetben viszont a cél-eszköz megkülönböztetés elmosódik. Éppen ezek a lehetőségek indokolják, hogy kilépve az objektív racionalitás keretei közül, szubjektív racionalitásról beszéljünk. A szubjektív racionalitás eltávolodik attól a kiindulóponttól, hogy a cselekvést a lehetséges megoldások közötti optimum – vagy legalább kielégítő – választásnak tekintse, mivel úgy véli, hogy a cselekvő több, szerinte az adott problémára megoldást kínáló elv közül bármelyik mellett dönthet. Az ilyen típusú racionalitás kognitív racionalitásnak tekinthető. A lényeg az, hogy a cselekvés különböző (pozitív vagy normatív) természetű a priori feltételekre támaszkodhat.

A kognitív döntésemélet a döntést mint döntéshozatali folyamatot a problémamegoldási folyamathoz nagyon hasonlóknak tekinti, és úgy tekint rá, mint egy speciális és időigényes információfeldolgozási folyamatra. Az emberi döntéshozót a számítógépes rendszer egyfajta analógiájának tekinti, és a végrehajtott információfeldolgozás típusát a végeredmény határozza meg.

Gabaix-Laibson (2000) megjegyzi, hogy a kognitív (megismerési) források úgy oszthatók el, mint minden más szűkösen rendelkezésre álló forrás.² Ez a probléma szinte minden döntési feladatnál jelentkezik. Az összes ilyen probléma esetében a döntéshozó nem ragaszkodhat a lehetséges legjobb kiválasztás meghatározásához, hanem inkább az ismeretein alapuló döntést kell hoznia és bizakodni a legjobbakban. A döntéshozóknak gyakran kell az ismereteiken alapuló becsléseket tenniük, amikor az időkorlát eleve kizárja a lehetséges alternatívák komplex elemzését.

A racionális viselkedés valójában helyzeti maximalizálást tesz lehetővé, ami azt jelenti, hogy csak adott helyzetben érvényes, mivel minden helyzet egyedi. A szabálykövető viselkedés koncepciója szerint a cselekvők nem egyedi történéseknek, eseményeknek számító különleges helyzetekre válaszolnak, hanem szituációs kategóriákat, „helyzet-típusokat” próbálnak kialakítani, amelyekhez adott szituációban viselkedésüket hasonlítani tudják. Így adott személy viselkedését a múltbeli tapasztalat is befolyásolja, tehát a szabálykövető perspektíva logikája szerint a megfigyelt viselkedési szabályszerűségek általános diszpozícióinak tulajdoníthatók, amelyek lehetővé teszik, hogy a cselekvő bizonyos helyzet-típusokba kerülve eligazodjék és szabályos viselkedést tanúsítson (Balogh-Kovács 2001).

A szabályok megkönnyítik a döntéshozatalt, ha bonyolult helyzetbe kerülünk. Egyszerűsítik a számításba veendő körülményeket, és ezáltal korlátozzák választási lehetőségeink terjedelmét. A szabály az adott helyzetben szabályozó funkciót tölt be, ami által könnyebbé válik a helyes alternatíva kiválasztása.

A kielégítő megoldás jelentős szerepet játszik a heurisztikus módszerekben. Az alternatívák ismeretének hiánya arra készteti a döntéshozót, hogy keresésbe kezdjen; így a keresés a döntéshozatali folyamat mechanizmusának szerves részévé válik. A másik kulcsfogalom a korlátozott racionalitás feltételezése mellett hozott döntések esetében a megelégedésre

² Cikkünkben leírják a korlátozottan racionális választás moduláris modelljét, amelyben formalizálják azt az elvet, hogy a megismerési források korlátozottan állnak rendelkezésre és takarékosan kell bánni velük.

törekvés, vagyis a hasznosság maximalizálása helyett a döntéshozó feltevésekkel rendelkezik az elfogadható alternatívákra vonatkozóan. Ha a keresés közben a döntéshozó olyan alternatívát talál, amely feltevéseinek megfelel, akkor abbahagyja a keresést, és a megtalált alternatívát választja. A heurisztikus modell abban az értelemben racionális, hogy az intuícióhoz fordul és elkerüli a „mérlegelési” költséget, de korlátozottan racionális abban az értelemben, hogy gyakran vezet „elfogult,” választáshoz.

Az utóbbi időben a játékelmélettel foglalkozók is lelkesedéssel fordultak a korlátozott racionalitás felé, amelyre gyakran hivatkoznak. Például a fejlődési játék modellekben a „játékot” ismétlődően játsszák és a játékos a megszerzett „tapasztalata” alapján ciklusról-ciklusra módosítja a stratégiáját (ld. később az ágenseknél leírtakat) (Conlisk 1996.) A korlátozott racionalitás több formában is megjelenhet ezekben a játékokban. Egyrészt korlátozott a rendelkezésre álló stratégiák száma, a játékosok a tevékenységek csak megadott halmazát képesek végrehajtani, és korlátozottak az általuk megjeleníthető következtetési típusok is.

Néhány szót kell ejteni a többtényezős (több szempontú) döntéshozatalról is, amely nagymértékben a racionális döntéshozatalra támaszkodik, amely magában foglalja a többtényezős hasznosság elméletet³, a játékelméletet és a döntéshozatal magatartási aspektusait is. Az ilyen problémák a következő közös jellemzőkkel rendelkeznek (Poh 1998):

- Lehetővé teszik az előre meghatározott alternatívák közül a legjobb alternatíva kiválasztását.
- Többféle kritérium felhasználását igénylik az értékelési folyamatban.
- Feltételezik a jellemzők (attribútumok) közötti konfliktusok feloldását.
- Tartalmazhatnak egymással közvetlenül nem összehasonlítható tényezőket.

A kiválasztás lényegében az „érintettektől” származó több tényezőt tartalmazó függvény alapján történő optimalizálást (többnyire maximálizálást) jelent. Az ilyen típusú problémák megoldása nem egyszerű feladat, mert sokszor egymással ellentmondásban vannak, és nem mindig lehet olyan megoldást találni, amely az összes nézőpont szempontjából a legjobb.

Mivel az esetek nagy részében a döntés minősége nagyon fontos, ezért az emberi döntéshozatal tökéletlenségeinek csökkentésére, illetve kiküszöbölésére az elmúlt évtizedek során kiemelt figyelmet fordítottak. A statisztika és az operációkutatás keretében számtalan módszert dolgoztak ki a racionális választás támogatására. Napjainkban ezek a módszerek megerősítésre kerülnek az informatikából, a kognitív pszichológiából és a mesterséges intelligenciából származó technikákkal, amelyek komplex módon kerülnek integrálásra különböző számítógépes programokban (Druzdel-Flynn, 2000).

A mesterséges intelligencia kutatása

Napjaink számítógépei csodálatos dolgokra képesek. A számítógépek ma már nemcsak a műveletek milliárdjainak nagyon gyors végrehajtására alkalmasak, hanem sokféle „emberi” tevékenységet is képesek pótolni. Mára már elmondhatjuk, hogy a számítógépek bizonyos tevékenységükben sok mindent átvettek az emberi intelligenciából. Ebből következően egyre többet hallani a számítógépes, illetve ma már az üzleti intelligenciáról is,

³ Multiattribute Utility Theory: Dyer-Edmunds-Butler-Jia (1997) tanulmányukban bemutatnak egy többtényezős hasznossági modellt.

amelynek a kifejlődése, a mai szintre jutása nagyon sokat köszönhet az 1950-es években megindult mesterségesintelligencia-kutatásoknak.

McCharty (2002) szerint a mesterséges intelligencia intelligens gépek készítésének tudománya és technikája, különösképpen vonatkozik ez az intelligens számítógépes programokra, és ahhoz a feladathoz kapcsolódik, amelynek célja a számítógépnek az emberi intelligencia megértésére történő felhasználása, de nem korlátozza módszereit a biológiailag megfigyelhető dolgokra.

Fekete-Gregorics-Nagy (1990) szerint olyan feladatok tartoznak a mesterséges intelligencia vizsgálati körébe, amelyek nem rendelkeznek minden részletében tisztázott, fix megoldó mechanizmussal, hanem szerepet kap a próbálkozás és az azt irányító emberi szakértelem és intuíció.

Russel és Norvig (2000) a mesterséges intelligenciát nem mint az ember alapvető jellemzőjét vizsgálja, hanem azt, hogyan lehet létrehozni és megérteni az intelligenciát mint a rendszerek általános tulajdonságát. Szerintük az intelligencia a sikeres viselkedés képességéhez kapcsolódik.⁴ A szerzők megjegyzik, hogy a különböző kategóriák képviselői között komoly viták dűlnak, és véleményük szerint az intelligencia lényegében a racionális cselekvéssel kapcsolatos. A racionális cselekvésen nem globális racionalitást értünk, hanem a szervezet, a vállalat által elérhető információmennyiség, annak feldolgozásához rendelkezésre álló technológia és a környezet támogatásával biztosítható racionális viselkedést, a korlátozott racionalitást. A korlátozott racionalitás feltételei szerinti választási mechanizmusnak alapvetően két jellemzője van: a keresés és a megfelelő kiválasztása. A korlátozott racionalitás elméletének magában kell foglalnia a keresés-elméletet (*Simon 1979*).

Továbbra is fontos kérdés, hogy mikor intelligens egy számítógépes rendszer? Talán ma még ezt a kérdést a legnehezebb megválaszolni. *McCharty (2002)* szerint az intelligencia a célélérési képesség számítógépes része, és a világon az intelligencia változatos formái és fokozatai fordulnak elő az embereknél, többféle állatnál és bizonyos gépek esetében. Úgy gondolom, hogy ez nagyon tömör megfogalmazás és nem igazán ad arra választ, mi is az intelligencia.

Russel és Wefald (1991) tanulmányukban azt írják, hogy az intelligenciának olyan általános meghatározására van szükség, amelynek közvetlen kapcsolatban kell lennie a rendszer inputjával és outputjával. Véleményük szerint az intelligencia szoros kapcsolatban van az eredményes viselkedés képességével, amit a mesterséges intelligencia ágens-alapú⁵ nézőpontjának neveznek. Az intelligenciát egy ágenshez rendelik és azt mondják, hogy intelligens az az ágens, amely érzékeli a környezetét és arra megfelelő választ ad. Ez a meghatározás is nyitva hagyja a kérdést, mert ahhoz meg kellene határozni, hogy mit értünk megfelelő válasz alatt?

Russel és Norvig (2000) az intelligenciát racionalitásnak nevezi, és megkülönböztet teljes racionalitást (maximálisan eredményes viselkedés az elérhető információk alapján), kalkulatív (számolási) racionalitást (teljesen racionális döntés a kezdetben rendelkezésre

⁴ Ez a mesterséges intelligencia úgynevezett ágensalapú szemléletmódja, amely egész könyvükre alapvetően jellemző.

⁵ *Futó et al. (1999)* szerint az ágensnek, az intelligenciához hasonlóan nincsen általánosan elfogadott jelentése. Ágens lehet bármi, ami bizonyos fokú önállósággal, autonómiával bír és amely reaktív (pl.: program, robot, gép, műszer, stb.)

álló információk alapján), metaszintű racionalitást (a megadott feltételek szerinti számítás és tevékenység optimális kombinációjának kiválasztása, ahol a tevékenység kiválasztása a számítási eljárással történik) és korlátozott optimalitást (a maximálisan eredményes viselkedés elérése az elérhető információ és a számítógépes erőforrások segítségével).

Az előzőekből is láthatjuk, nem könnyű meghatározni, hogy mit értsünk intelligencia alatt. Ahhoz, hogy a döntéstámogató rendszert intelligensnek nevezhessük, legalább erre az esetre konkretizálni kellene, hogy mikor tekinthetünk egy számítógépes programot intelligensnek. Úgy gondolom, hogy az emberi módra gondolkodó és cselekvő rendszerektől a mindennapi munkában még elég messze vagyunk, még ha annak bizonyos kezdeményei már meg is jelentek. A közgazdasági és a mérnöki tudományokra jellemző a racionalitásközpontú megközelítés. Ennek megfelelően a kalkulatív racionalitás és korlátozott optimalitás lehet az, amelyet az intelligens gazdasági rendszereknek tartalmazniuk kellene.

A fentiekből következően kicsit leegyszerűsítve azt is lehetne mondani, hogy az intelligencia nem más, mint célok elérésének képessége. Amelyik rendszer gyorsabban és könnyebben képes a céljait elérni, az intelligensebb. Ez az utóbbi az intelligencia növelése érdekében magában foglalja a tanulás képességét is. Minden intelligens rendszernek rendelkeznie kell valamilyen fő céllal. Más intelligens rendszerekkel való kapcsolattartás érdekében megfelelő kommunikációs rendszerrel is rendelkeznie kell. Ezekből következően azt is lehetne mondani, hogy az intelligencia adekvát válaszadási képesség a rendszert érő „ingerekre”.

Az intelligens ágens

A mesterséges intelligencia meghatározásából is következik, hogy az intelligencia lényegében a racionális cselekvéshez kapcsolódik, amely cselekvés végrehajtói lehetnek az ágensok, vagy pontosabban fogalmazva a racionális ágensok. Az elmúlt közel két évtizedben az ágens kifejezés meghonosodott mind a mesterséges intelligenciában, mind a számítógép-tudományban. Napjainkban egyre sűrűbben hallhatjuk az ágens szót a software engineeringben, az adatfeldolgozásban, a konkurens rendszerek kutatásában, valamint a robotikában (Jennings-Wooldridge 1995; Nwana 1996.) Azt lehetne mondani, hogy igazán intelligens számítógépes rendszert ágensok alkalmazása nélkül nem lehet létrehozni, de semmiképpen sem szabad a mindenáron való alkalmazásukra törekedni. Sok egyetemen és kutatóintézetben folytatnak vizsgálatokat az intelligens ágenseknek a társadalom és a gazdaság különböző területein történő felhasználásával kapcsolatban. Megjelentek az intelligens ágensok a heurisztikus módszereket alkalmazó döntéstámogató rendszerekben is.⁶

Az ágensalapú modellek alkalmazása jelentős áttörést hozott a programfejlesztésben is⁷ (Wooldridge-Jennings 1995)⁸. Russel-Norvig (2000) szerint *ágens bármi lehet, ami érzékelői által érzékeli környezetét és beavatkozó szervei által megváltoztatja azt. (1. ábra) Az ágens képes alkalmazkodni a környezetéhez, képes tanulni cselekedetei hatásából, illetve*

⁶ Pl.: Intelligent Systems Laboratory of Swedish Institute of Computer Science - <http://www.sics.se/isl/>

⁷ szoftverágensek vagy szoftverrobotok (softbots - software robots) (Russel-Norvig 2000; Wooldridge-Jennings 1995)

⁸ Ez a tanulmány egy viszonylag széleskörű áttekintését adja az ágensok elméletével foglalkozó szakirodalomnak is.

a környezet változásaiból, és e tudást felhasználva képes változtatni tervein, cselekedetein annak érdekében, hogy azok megfelelőbbek legyenek.

Az ágensekhez többek között két jelzőt szoktak kapcsolni: *racionális és autonóm*. A *racionális ágens* helyesen cselekszik, ahol a helyes cselekedet mértéke az elvárt siker kifejezése a megfigyelések tükrében, egyszerűbben fogalmazva az, hogy az ágens milyen mértékben képes a céljait elérni. A tanulás nagymértékben növeli az ágens problémamegoldó képességét, s így az autonómiáját is. Az *autonómia* tehát a beépített tudáshoz kapcsolódik, amelynek a hatása az ágens működésének autonómia-fokát határozza meg. Azt is mondhatjuk, hogy az autonómia foka az jelenti, hogy mennyire képes az ágens közvetlen emberi beavatkozás nélkül működni.

Az ágens képes környezetéből új információkat gyűjteni, tárolni azokat, összefüggéseket feltárni közöttük, illetve felhasználni azokat. Az ágens képes alkalmazkodni a környezetéhez. Képes tanulni cselekedetei hatásából, illetve a környezet változásaiból. E tudást felhasználva képes változtatni tervein, cselekedetein annak érdekében, hogy azok optimálisak legyenek.

Futó, et al. (1999) szerint az ágenseknek minimálisan a következő négy tulajdonsággal kell rendelkezniük:

- beágyazottság (az ágens a környezetébe ágyazott),
- reaktivitás (környezetérzékelés és valós idejű reagálás),
- autonómia (önálló működés),
- helyzetfüggőség (helyzethez és szerephez kötődés).

Az *AgentBuilder*⁹ rendszer leírásában az intelligencia foka szerint az 1. táblázatban szereplő háromféle típust különböztetik meg (*AgentBuilder, 1999*).

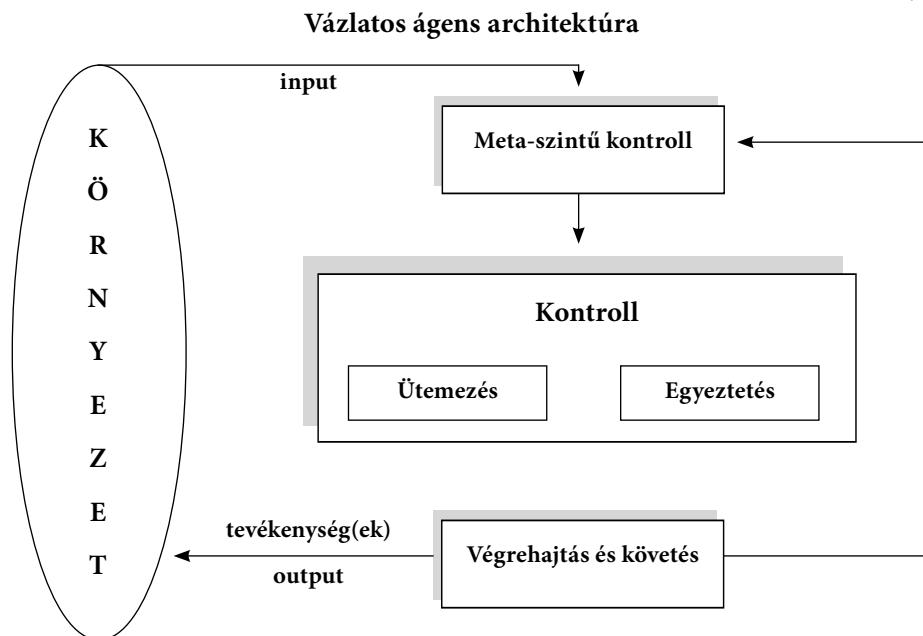
1. táblázat

Egy intelligens ágens jellemzői

Ágens típusa	Ágens jellemzői
Ágens	autonóm módon működik
	más ágensekkel és a felhasználóval kommunikál
	figyeli működési környezetének az állapotát
Intelligens ágens	képes szimbólumokat és absztrakciókat használni
	a tárgyhoz tartozó tudás jelentős mennyiségét képes hasznosítani
	adaptív célorientált viselkedésre képes
Valóságosan intelligens ágens	képes a környezetből tanulni
	toleráns a hibás, a váratlan és a rossz inputok iránt
	valós idejű tevékenységekre képes
	természetes nyelv felhasználásával képes kommunikálni

⁹ integrált eszköztár intelligens software ágensek létrehozására

1. ábra



Tekintsük át az ágensek főbb jellemzőit, különös tekintettel a software ágensekre, amelyek a döntéstámogató- és a szakértői rendszerek fejlesztése szempontjából is lényegesek.

Az ágensekkel foglalkozó kutatók két nagy táborra oszthatók: a klasszikus szimbolikus logikai és a viselkedési táborra (Wooldridge-Jennings 1995). A klasszikus megközelítés az ágenseket szimbolikus tudással ruházza fel, amely az ágens körülvevő környezethez kapcsolódik. Ez a tudás szabályokból, keretekből (*frames*), szemantikai hálóból¹⁰ (semantic network), vagy általánosabban fogalmazva bizonyos logikai nyelvi formulákból áll. Az ágens ezen tudás „manipulálásával” új tudást állít elő. Ez a „manipuláció” általában a tételbizonyítást jelenti. Ebből következően a klasszikus tábor képviselői a szimbolikus mesterségesintelligencia-kutatás két fő területével foglalkoznak, a tudás reprezentációs technikákkal és az automatikus tételbizonyítással (Genesereth-Nilsson 1987; Steels 1990; Connah-Wavish, 1990; Brooks 1991).

A Russel-Norvig (2000) féle meghatározás kissé általános, a döntéstámogató rendszerek vonatkozásában is értelmezhető a racionalitás és az autonómia is, de esetükben inkább olyan magasszintű programrendszerekről van szó, amelyek szimbolikus reprezentációt alkalmaznak, a klasszikus táborhoz állnak közelebb, és megismerési jellegű (*cognitive-like*) (Shoham 1990) funkciót töltenek be¹¹. Az ilyen típusú rendszereket Genesereth és Nilsson (1987) „mérlegelő” ágenseknek (*deliberate agent*) nevezték, amely ágensek a következő tulajdonságokkal rendelkeznek:

¹⁰ Gráfokra épülő ismeretrepresentálási módszer, amely fogalmakat vagy objektumokat, illetve azok attribútumait képviselő csomópontokból, valamint a köztük fennálló relációkat reprezentáló élekből áll (Futó, et. al. 1999)

¹¹ explicit logikai vagy pseudo-logikai következtetés

1. Tudása reprezentációjaként formulák explicit módon megadott adatbázisával rendelkezik valamelyik logikai nyelven;
2. A következő folyamatos ciklusban működik: észlelés – mérlegelés – tevékenység ...;
3. A következtetés (*reasoning*) és a tevékenység eldöntése a logikai következtetés (*logical inference*) valamilyen formáján alapszik.

Ahhoz, hogy egy ilyen ágens működni tudjon, két funkciót kell magában foglalnia (Wooldridge-Jennings 1995):

- *Átalakító funkció*: a valós világnak egy pontos és szabatos, szimbolikusan leírt világba történő transzformálása.
- *Következtetési funkció*: tevékenységek végzése a szimbolikus reprezentáción.

Az ilyen rendszerekben központi szerepet játszik a „tervező”, aki központi eleme a mesterségesen intelligens ágens-strukturának. A „tervező” adja meg a „világ/környezet” aktuális állapotát és azt a célt, amit az ágensnek el kell érnie (Georgeff 1987). Az ágens a környezetét üzenetek által érzékeli, illetve ezek által tart kapcsolatot más ágensekkel.

Ez az ágenstípus nagyon szorosan kapcsolódik a logikai programozáshoz, mert az ágens jellemzőiként leírtak akár magát a logikai programnyelvek fejlettebb implementációit is jellemezhetnék. A leírtakhoz hasonló elven működik az LPA Prolog++, illetve az LPA Prolog Flex szakértői shell. Az LPA Prolog++ ötvözi a logikai programozás és az objektumorientált programozást, és jól alkalmazható szimulációs programok írására, s bizonyos mértékig megfelel az ágensprogramozás követelményeinek is. A bizonyos mérték azt jelenti, hogy nem képes a megfelelő autonómia biztosítására.

Mit nevezünk ágensprogramoknak¹², vagy inkább a napjainkban elterjedt kifejezést használva, szoftverágenseknek? Azt tudjuk és a felhasználók is tapasztalják, hogy nehéz kiváló minőségű szoftvereket készíteni a komplex valós világ problémáinak megoldására. Az ágensorientált megközelítés képes megnövelni modellezési, tervezési és komplex szoftverrendszer építési képességünket. (Jennings 2000) Ugyanakkor az ágensorientált rendszer elkészítése nagyobb előkészítő és tervező munkát igényel, mint a hagyományos rendszerek. Russel és Norvig (2000) az ágensek, a gépi architektúrák és a programok közötti összefüggést a következőképpen adja meg:

Ágens = Gépi architektúra + Program

Könyvükben négyféle ágensprogramot különböztetnek meg:

- egyszerű reflexszerű ágens (feltétel-cselekvés típusú ágens),
- ágensek, amelyek nyomon követik a világot (belső állapottal rendelkező reflexszerű ágens),
- célorientált ágensek (a jelenlegi helyzet leírása mellett megjelenik a célinformáció, amelynek függvénye a cselekvés),
- hasznosságorientált ágensek (a cselekvés hasznosságát is vizsgálja; rendelkezik egy hasznosságfüggvénnyel).

A felsorolásból láthatjuk, hogy az egyes ágensek hogyan oldják meg a problémákat. A magas szintű döntéstámogató rendszerek egyes komponensei cél- vagy hasznosság ori-

¹² Program, amelyik kölcsönhatásban van a felhasználóval, más alkalmazásokkal, más ágensekkel és együttműködik a felhasználóval.

entált ágensekként foghatók fel, vagy másként problémamegoldó ágenseknek is nevezhetők. Annak megfelelően, ahogy *Russel–Norvig (2000)* írja, hogy az ágens képes célt maga elé tűzni, és megpróbálja azt elérni. Azt szokták mondani, hogy *minden szoftver ágens program, de nem minden program ágens*. Felvetődhet a kérdés, hogy mikor tekinthető egy program ágensnek?

Wooldridge (1999) azt írja, hogy az ágens olyan „tevékenység repertoárral” rendelkezik, amely az ágens tényleges „kapacitását” jelenti, azt a képességét, amellyel a környezetét módosítani képes. A kulcsprobléma, amellyel szemben találjuk magunkat, annak eldöntése, hogy milyen tevékenységek végrehajtására kell az ágensnek képesnek lennie annak érdekében, hogy a tervezett célt minél jobban teljesíteni tudja. Ezt a „képességet” az *ágens architektúra* tartalmazza, ami valójában egy *szoftverarchitektúra* a környezetbe beágyazott döntéshozó rendszer számára (*Jennings 2001*). A döntéshozatali folyamat komplexitását a környezeti tulajdonságok határozzák meg.

Az objektumorientált programozással foglalkozók sokszor egyenlőséget tesznek az objektum és az ágens között. Ennek az az alapja, hogy ténylegesen nagyon sok hasonlóság mutatható ki közöttük, de vannak különbségek is. Az objektum számítógépes entitásként definiálható, amely bizonyos állapotokat zár magába, képes tevékenységeket végrehajtani, és üzenetek küldésével kommunikál. Az objektum és az ágens közötti egyik különbség az autonómiában jelentkezik. Az objektum általában véve nem gyakorol felügyeletet a viselkedése felett. Objektumorientált esetben a döntés annál az objektumnál van, amelyik azt meghívja, míg az ágensek esetében a döntés annál az ágensnél van, amely a kérést megkapja, illetve az ágens saját tudása alapján önállóan (meghívása nélkül) is kezdeményezhet valamilyen tevékenységet¹³ (*Wooldridge 1999; Wooldridge–Ciancarini 2001*). Mindez nem jelenti azt, hogy ágenseket nem lehet objektumorientált technikával létrehozni, de az objektumok számára erősebb autonómiát kell biztosítanunk, mert az ágensek esetében leírt autonómia nem része az alapvető objektummodelleknek. *Wooldridge (1999)* a második fő különbségnek azt tekinti, hogy az ágens rugalmas (reaktív, proaktív) autonóm viselkedéssel rendelkezik, míg harmadikként azt, hogy az ágens a kontroll saját „szálával” (*thread*¹⁴) rendelkezik. Ez utóbbi különbség megoldása ma már több objektumorientált és logikai programnyelvben is megjelenik. Fel kell azonban hívni a figyelmet, hogy a konkurens programozás magában nem oldja meg a problémát (ld. aktív és passzív objektum). Egy aktív objektum lényegében autonóm, de nem szükségszerűen rendelkezik rugalmas autonóm viselkedéssel.

Az ágensrendszerek fejlesztésében a fő probléma megtalálni az ésszerű egyensúlyt a között, ahogy az ágens a környezet változásaira reagál, és a között, ahogy céljai elérése érdekében tevékenykedik. Könnyű olyan ágens létrehozni, amelyik csak a környezet változásaira reagál, és olyat is, amelyik csak a céljai elérése érdekében tevékenykedik, de olyat amelyik a két dolog közötti egyensúlyt is képes megteremteni már sokkal nehezebb. Azo-

¹³ *Wooldridge (1999)* azt írja, hogy az objektum és az ágens közötti különbség megvilágítására a következő - általa érdekesnek tartott - szlogent hallotta: Az objektumok ingyen dolgoznak, az ágensek pénzért. (*Objects do it for free; agents do it for money.*)

¹⁴ A „szál” (*thread*) a deklaratív konkurens programozáshoz tartozó fogalom, amely a „kérés által irányított” (*demand-driven*) programozást támogatja, amely például jól alkalmazható ütemezési problémák megoldásának programozásában. Egy adott „thread” futtásra kész program, amelynek a működése mindaddig felfüggesztődik, amíg a működéséhez szükséges összes információt meg nem kapta. (*Van Roy–Haridi 2001*) A „szál” egy utasítás halmaz egy programon belül, amely ütemezhető és végrehajtható (*Berg–Fritzing 1999*).

kat az ágenseket, amelyek erre képesek, „tapasztalati következtető rendszereknek” nevezük (Wooldridge 1998/a). Az ilyen rendszerek olyan környezetben is jól működnek, ahol a hagyományos szoftver engineering módszerekkel nem lehet sikereket elérni.

Meg kell jegyezni, hogy általában nem az egész szoftverrendszer (pl. a döntéstámogató rendszer) az ágens, hanem léteznek olyan komponensei, amelyek ágens funkciókat hajtanak végre, vagyis a rendszer egyes feladatait ágensek oldják meg. Ebből következően egy nagyobb rendszerben több ágens működik együtt, azonos vagy különböző tevékenységeket végezve. A több ágenssel is rendelkező rendszereket multi-ágens rendszereknek nevezük, amely rendszerekhez több „szál” is kapcsolódhat (*multi-thread systems*), és ilyen rendszerek esetében minden ágens rendelkezik legalább egy „szállal”.

Az ágensek gazdasági alkalmazási lehetőségei

Az ágens alapú számítógépes rendszerek gazdasági alkalmazásának számos példájával találkozhatunk (Jennings–Wooldrige 1998). Először is néhány szót szólni kell egy újonnan kifejlődött területről, az ágens alapú számítógépes közgazdaságtanról (ACE¹⁵), amely terület kutatóinak fő célja megérteni a gazdasági folyamatokban lévő globális szabályszerűség látszólag spontán alakulását, úgy mint a decentralizált piacgazdaság kereskedelmi tevékenységeinek nem tervezett koordinációját, amit a közgazdászok Adam Smith „lát-hatatlan kezével” társítanak. Az ACE egyfajta keveréke az evolúciós közgazdaságtanból, a kognitív tudományból és a számítógép-tudományból származó fogalmaknak és eszközöknek. Az ACE keretrendszer ágensei jellemzően olyan heterogén entitásokat modelleznek, amelyek a környezettel és más ágensekkel való kölcsönhatásaikat belső adatok és viselkedési szabályok alapján határozzák meg¹⁶.

Az ACE kutatók egyik fő területe annak megértése, hogy miért az adott globális szabályok figyelhetők meg a felülről lefelé tervezés és a kontroll hiányában kifejlődő és fennmaradó decentralizált piaci gazdaságokban: például a kereskedelmi hálózatok, a szociálisan elfogadott jólét, piaci protokollok, gazdasági ciklusok és a technológiai innováció általános elfogadása. A megoldásra váró feladat az, hogy bemutatásra kerüljön, hogyan változhatnak ezek a globális szabályok az autonóm ágensek ismétlődő egymásra hatásai által.

A kereskedelmi hálózati játék (TNG) egy moduláris felépítésű, ágens alapú keretrendszer a különbözőképpen specifikált piaci protokollok alatt működő stratégiaileg kölcsönösen egymásra ható kereskedők (vásárlók, eladók és ügynökök) közötti kereskedelmi hálózatok kialakulásának és fejlődésének tanulmányozására (Tesfatsion 2000; Tesfatsion 2001). Az erőforrás-korlátozott kereskedők egymás után következő generációi a folyamatosan korrigált elvárt hasznosság alapján kiválasztják vagy elutasítják a kereskedelmi partnereiket, kétszemélyes játékként modellezett kereskedelmi együttműködést folytatva, amelynek során a kereskedelmi stratégiájukat folyamatosan fejlesztik. A TNG által biztosított fejlődési eredmények négy különböző szinten tanulmányozhatók:

¹⁵ Agent-based Computational Economics

¹⁶ lásd Trade Network Game (kereskedelmi hálózati játék) – TNG

1. kereskedői jellemzők (a személyiségtípusok endogén fejlődése),
2. a kereskedelmi hálózatok kialakulása (ki kivel kereskedik és milyen szabályok szerint),
3. a megjelenített kereskedői magatartás (együtműködés, „megszerzés”, ...),
4. egyedi és társadalmi boldogulás (piaci hatékonyság, piaci részesedés, egyéni hasznossági szint, ...)

A TNG program struktúráját a 2. ábra, az ágens felépítését pedig a 3. ábra mutatja be. A TNG statikus struktúrája 3 alapvető osztály definícióin és kapcsolatain keresztül mutatható be (*McFadzean–Tsfatsion 1999*):

- *tngSimulation*, amelyik a teljes szimulációt menedzseli,
- *tngPopulation*, amelyik a kereskedők fejlődését menedzseli,
- *tngTradeBot*, amelyik egy egyszerű kereskedőt szimulál (vevő, eladó, ügynök).

2. ábra

A kereskedelmi hálózati játék (TNG) pseudo-kódja¹⁷

```
int main() {
    Init(); // Véletlen kereskedői stratégiával elkészíti a
           // kezdő kereskedő „generációt”
    For (G = 1, ..., GMax) { // A generáció ciklus
        InitGen(); // A kereskedők konfigurálása a felhasználó
                 // által megadott paraméterekkel
        For (I= 1, ..., IMax) { // Kereskedői ciklus
            MatchTraders(); // A kereskedő partnerek meghatározása,
                           // megadva az elvárt hasznosságot, felsorolva
                           // a munkakeresést és az inaktivitás költségeit
            Trade(); // A kereskedések végrehajtása és a kifizeté-
                   // sek nyilvántartása
            UpdateExp(); // Az elvárt hasznosság aktualizálása, felhasz-
                       // nálva az újonnan nyilvántartásba vett költsé-
                       // geket és kifizetéseket
        }
        // Környezeti fázis
        AssessFitness(); // A kereskedő megfelelési pontjának megha-
                       // tározása
        Output(); // Kereskedői információ output
                // Fejlődési fázis
        EvolveGen(); // Új kereskedői generáció létrehozása
    }
    Return 0;
}
```

¹⁷ Felhasználva *Tsfatsion (2000) tanulmányában szereplő kódleírást. A bemutatott kód csak szimbolizálja a folyamatot, a program tényleges működtetése ugyanezeket a lépéseket tartalmazza, de ott el kell végezni a konkrét paraméterezéseket, definiálni kell a kereskedőket, mint szoftver ágenseket (tradebot – kereskedőrobot).*

A modellben a saját tevékenységének értékelése útján tanulásra és fejlődésre képes ágens definiálható. Ezek az ágensok autonóm, endogén, egymással kommunikáló szoftverágensok (*softbots*), amelyek belső viselkedési függvénnyel és belsőleg tárolt információkkal rendelkeznek, amely az azonosítót is tartalmazza más ágensok számára. Az előzőek alapján az ágensok képesek meghatározni kereskedelmi partnereiket és kialakítani kereskedői magatartásukat. A modell rövid leírása megtalálható *McFadzean és Tesfatsion (1997)* cikkének 2. fejezetében, illetve *McFadzean és Tesfatsion (1999)* cikkének 4.3. fejezetében.

A TNG kereskedelmi hálózati játék viszonylag egyszerű program, de arra nagyon jól alkalmas, hogy be lehessen mutatni vele a kereskedők működését és tevékenységük fejlődését a decentralizált piacokon. A programnak az egyik fontos eleme a SymBioSys modul, amelynek leírása megtalálható *McFadzean és Tesfatsion (1999)* cikkében.

A TNG-platform az elemzés 3 szintjén teszi lehetővé eredményeinek kísérleti tanulmányozását. Először az egyedi kereskedők (*tradebot*) attribútumait tudjuk közvetlenül felhasználni a kereskedői típusok fejlődésének becsléséhez. Másodsor a „ki kereskedik kivel” kérdéshez kapcsolódó részletes információkat összegyűjthetve követhetjük nyomon a kereskedelmi hálózatok endogén kialakulását és fejlődését. Ehhez segítséget nyújt, hogy minden kereskedői ciklus végén az AssessFitness függvény arra „ösztönzi” a kereskedőket, hogy adják meg a potenciális kereskedelmi partnereikről tárolt adatokat, amelyek a következőket tartalmazzák: a kereskedelmi partnerektől kapott kifizetések összege és száma, az aktuálisan várható kifizetések száma, az aktuális kereskedések száma, a kereskedelmi partnerek tevékenységeinek (együttműködés és elpártolás) teljes rendezett listája. Végül valamennyi egymás után következő generáció kereskedőihez rendelt illeszkedési pontok felhasználhatók a társadalmi jólét becslésére, vagyis különböző statisztikai jellemzőket (minimum, maximum, átlag) számolhatunk az egymás utáni kereskedői generációk vonatkozásában. (*McFadzean–Tefatsion 1999*)

A 3. ábra bemutatja, hogy milyen információkat kell megadni egy kereskedőágensre vonatkozóan. A szabályok megadására valamilyen ágens leíró nyelvet használnak (pl: LALO, AgentBuilder, Jess). A megadott vázból láthatjuk, hogy milyen információkat igényel egy ágens, és azt is, hogy a működésére vonatkozó szabályokat előre meg kell tervezni. Az autonómiájából és az intelligenciájából következően a többi ágenstől és a környezettől kapott információk alapján működését az általunk meghatározott szabályok figyelembevételével képes meghatározni. Ha a szabályrendszert jól fogalmazzuk meg, akkor képes lesz „fejlődni” azáltal, hogy a korábbi tevékenységeiből „tanul”.

3. ábra

A kereskedelmi hálózati játék (TNG) ágensének leírása¹⁸

```

class TradeBot
{
    Belső társadalmi normák:
        Piaci protocollok más kereskedőkkel történő kommunikációhoz;
        Piaci protocollok a kereskedelmi partnerek kereséséhez és összepárosításához
        Piaci protocollok a kereskedelmi együttműködésre.
    Belsőleg tárolt állapot információk:
        Jellemzőim;
        Dotációk;
        Tudásaim és preferenciáim;
        Címek magam és más kereskedők számára;
        Pótlólagos adatok más kereskedőkről.
    Belső viselkedési szabályok:
        Új információk összegyűjtésére és feldolgozására vonatkozó szabályaim;
        Kereskedői viselkedésem meghatározására vonatkozó szabályok;
        Tudásaim és preferenciáim aktualizálására vonatkozó szabályok;
        Hasznossági (megfelelési) szintem mérésére vonatkozó szabályok;
        Szabályaim módosítására vonatkozó szabályok.
};

```

Hasonló modellek a gazdaság más területeire is kialakíthatók és alkalmazhatók. Például a pénzügyi piacok konvencionális modelljei a racionális választás és a piaci hatékonyság feltevésén alapulnak (*Chan–LeBaron–Lo–Poggio 1999; Parkes–Huberman 2001*). Az elmúlt években az ACE modellek kutatói ez utóbbi terület számára próbálnak hasonló típusú ágenseket létrehozni. A számítógépes ágens alapú modellek kiemelik az árak és a piaci információk közötti viszonyokból tanuló és nagyon különböző magatartást tanúsító kereskedői csoportok kölcsönhatását és a tanulási dinamikáját. Egy ilyen rendszerben a megadott szabályok figyelembevételével az ágensek döntik el, hogy mennyi kockázatos eszközt vásárolnak. Az ágensek folyamatosan fejlődnek saját és környezetük döntéseinek elemzése útján. A tanuláshoz genetikai algoritmust¹⁹ használnak fel (*LeBaron 2000*).

A modellnek nemcsak makroökonómiai és pénzügyi alkalmazásai képzelhetők el, hanem felhasználható szervezetek (vállalatok) viselkedésének modellezésére is. Az ACE modellek felhasználásával tesztelni lehet integrált elméleteket, valamint azt, hogy az eltérő mikrostrukturák mennyire képesek támogatni az adott makroszintű szabályozottságot.

A szakirodalomban találkozhatunk speciális vállalati alkalmazásokkal is (*Jennings et al. 1996²⁰; Jennings–Wooldridge 1998; Bussmann–Jennings–Wooldridge 2001*), mint például

¹⁸ Felhasználva *Tesfatsion (2000)* tanulmányában szereplő ágens leírást.

¹⁹ A genetikai algoritmus egyedek halmazából kiindulva – szelekciós és reprodukciós műveletek felhasználásával – egy sikeres egyed „evolúcióját” végzi el. A sikerességet az alkalmassági függvény (fitness function) méri. Az alkalmassági függvény valamilyen teljesítménymérték vagy jutalomfüggvény. Az „evolúciós” folyamat felfogható a megerősítéses tanulás egyik formájaként. Megerősítéses tanulásnak (Reinforcement learning) azt nevezzük, amikor az ágens az általa végrehajtott tevékenység jutalmazásával kap visszajelzést, de nem közlik vele, hogy mi a helyes tevékenység, hanem az ágensnek kell meghatároznia, hogy melyik tevékenység volt a hibás. (*Russel–Norvig, 2000*)

²⁰ Ágens alapú üzleti folyamat-menedzsment

az ágens alapú elektromos áramelosztó rendszer (CIDIM), vagy az ágens alapú egészségügyi ellátás menedzsmentrendszere (Jennings 1995).

A különböző típusú szoftver ágensek optimális támogatást nyújthatnak a döntési problémák megoldásának különböző szakaszaiban. Ilyenek lehetnek például információk szűrése, adatkonverziók, kapcsolattartás a felhasználóval, adatok tárolása és előkeresése, problémaelemzés, speciális megoldás-keresési eljárások, megosztott problémamegoldás, eredmények ellenőrzése, feladatmonitorozás, távoli elérések kezelése, grafikus lehetőségek biztosítása, fejlődési algoritmusok használata stb (Bui-Lee 1999). Fontos megjegyezni, hogy a felsorolt feladatok nagy része egyszerű eljárásként is megírható. Ha ezeket ágenseknek adjuk feladatul, akkor más gondolkodásmóddal kell a DSS felépítéséhez hozzáállnunk, és nem szabad elfeledkezni az ágensek autonóm viselkedésének pontos megtervezéséről.

A Mozart-Oz programrendszer és az ágensek

A Mozart programozási rendszer, amely több mint egy évtizedes, a programozási nyelvek tervezése és implementálása, a korlát-alapú következtetés, az osztott programozás és az ember-számítógép kapcsolat területén végzett kutatás eredménye, egyedülálló fejlett fejlesztési környezet Windows és Unix platformú intelligens, osztott alkalmazások számára. A Mozart az Oz programnyelven alapul, amely mint egységes egész támogatja a deklaratív, az objektumorientált, a korlátos és a konkurens programozást. A Mozart ezen tulajdonságai által egyedülálló lehetőséget biztosít multiágens rendszerek fejlesztéséhez.

Az Oz kiváló konkurens programozási nyelv, amely a Mozart rendszerbe implementált „szálakhoz” kapcsolódó kezdeményező ütemezés segítségével könnyen kezelhető megoldások létrehozását magas szinten támogatja. Más programnyelvektől eltérően, amelyek egyetlen programozási paradigmába szorítják a programozót, az Oz koherens keretet biztosít több programozási paradigma egyidejű használatához is (pl: konkurenskorlátos logikai programozás). Az egyes szálak – akár az interneten keresztül is – képesek egymással kommunikálni (Van Roy-Haridi-Brand 2000).

Az előzőek figyelembe vételével a Mozart-Oz programozási rendszer kiváló felületet biztosít intelligens multiágens, illetve mobil ágens rendszerek létrehozására. A korlátozott terjedelem miatt csak egy egyszerű aszinkron termelői-fogyasztói szimulációs példát mutatok be az ágensek működésére a Mozart-Oz rendszer alkalmazásával, a stream (adatfolyam) felhasználásával.

Egy termelő (*Termelo*) termékeket állít elő, amelyeket elhelyez a stream-re²¹ és a fogyasztó (*Fogyasztó*) a stream-ről leemeli a terméket. A fogyasztó csak azután tud lemenni terméket a stream-ről, ha a termelő elhelyezte ott. Persze felmerülhet az a kérdés is, hogyan kezelhető az a helyzet, amikor a termelő sokkal gyorsabban állítja elő a termékeket, mint azt a fogyasztó igényli. Ez a fogyasztó által küldött információkkal (üzenetekkel) kezelhető, amit kereslet-vezérelt kontrollnak is nevezhetünk, azaz a termelő csak akkor állít elő terméket, ha a fogyasztó igényli azt. Úgy is fogalmazhatnánk, hogy a termelő csak a fogyasztó által igényelt mennyiséget termeli.

Az elkészült egyszerű programot a 4. ábrán mutatom be. A példában a fogyasztó 100 000 termék előállítását kéri a termelőtől. A termelő a termék előállítását a ‘termék’ szónak a stream-re történő elhelyezésével jelzi, és mindaddig folytatja a termelést, amíg

²¹ A stream tekinthető egy virtuális piacnak.

a fogyasztótól jelzés nem érkezik a termelés befejezésére (*nil*), és akkor kiírja a ‘Termelés vége.’ szöveget. A fogyasztó egy ciklusban várja a termék elkészültét, mindaddig amíg az igényelt termék darabszáma le nem csökken nullára. Ha van elkészült termék (megjelenik a stream-en a ‘termék’ szó), akkor az elfogyasztásra kerül, ha nincsen kész termék, akkor a fogyasztó vár annak elkészültére. A fogyasztó 1000 termékenként kiírja a ‘1000 termék elfogyasztva.’ szöveget. A programot a

```
{Fogyasztó 100000 thread {Termelő $} end}
```

utasítás indítja el, összekötve a termelőt és a fogyasztót.

4. ábra

A termelő-fogyasztó szimuláció programja

```
local
  proc {Termelő Xs}
    case Xs of X|Xr then X = termék {Termelő Xr}
    [] nil then {Browse 'Termelés vége.'}
    end
  end
end

proc {Fogyasztó N Xs}
  if N=<0 then Xs=nil
  else X|Xr = Xs in
    if X == termék then
      if N mod 1000 == 0 then
        {Browse '1000 termék elfogyasztva.'}
      end
      {Fogyasztó N-1 Xr}
    else
      {Fogyasztó N Xr}
    end
  end
end

in
  {Fogyasztó 100000 thread {Termelő $} end}
end
```

A 4. ábrán közölt program egy rendkívül leegyszerűsített megoldása az ágensnek, de ez a kicsi program is érzékelteti azt a lehetőséget, amely egy bonyolultabb, a Mozart-Oz további lehetőségeit is kihasználó programmal elérhető. A bemutatott program lényegében csak a szál (*thread*) és az adatfolyam (*stream*) biztosította lehetőségekre épül, amelyek csak egy kis részét képezik a programrendszer nyújtotta lehetőségeknek.

Annak ellenére, hogy napjainkban az ágens technológia fontos szerepet játszik az élenjáró szoftveralkalmazások fejlesztésében, nem szabad túlértékelni. Az elmúlt időszakok fejlesztéseinek egy részét, amelyek ágens technológiát alkalmaznak, meg lehetett volna valósítani ágensek felhasználása nélkül is. Nem szabad azt hinnünk, hogy az ágensek

mindenre megoldást jelentenek, és minden esetben csak azokkal lehet a megfelelő megoldást elérni. Azt sem szabad ugyanakkor gondolni, hogy az ágensekkel való foglalkozás csak divat vagy hóbort, mert bizonyos esetekben, ha a feladatnak magas szinten megfelelő szoftvert akarunk készíteni, elkerülhetetlen az alkalmazásuk. Az ágensek alkalmazásánál is meg kell találnunk az „arany középutat”. *Jennings és Wooldridge (1998)* szerint arról sem szabad elfeledkezni, hogy az ágensek alkalmazása problémákat is okozhat:

1. Nem megfelelő azon problémák számára, ahol globális feltételek, valós idejű válaszok alapján történik a vezérlés.
2. Az ágensek viselkedése lokális, ezért nem lehetséges a globális „tudás” kezelése általuk. Az ágensek a rendszer egészének szempontjából szuboptimális megoldások előállítására képesek.
3. Ahhoz, hogy bizonyos feladatokat az ágensekhez delegáljunk, először bízni kell bennük, ami azt jelenti, hogy nagyon alaposan ki kell dolgoznunk az ágensekre bízott feladatokat, az ágensek reakcióit, azok „viselkedését”. (Az ágensek „tudnia” kell a korlátait.)

Felvetődik a kérdés: vajon a számítógépes programok, nevezetesen a döntéstámogató rendszerek lehetővé teszik-e, hogy az emberek racionálisabban gondolkodjanak, racionálisabb döntést hozzanak. Ehhez hasonló kérdést tett fel a *Post-Gazette* tudományos szerkesztője, *Byron Spice* Herbert A. Simon-nak is, aki a következőt válaszolta: „*Minden, ami több ismeretet nyújt a számunkra, lehetőséget biztosít arra, hogy racionálisabbak legyünk.*”²²

Hivatkozások

- AgentBuilder. (Integrált toolkit intelligenságensek létrehozására.) Reticular Systems, Inc., San Diego, February 1999. [<http://www.agentbuilder.com/>]
- Balogh, G.-Kovács, M.: *A gazdaság társadalmi dimenziói*, Osiris-BIP, Budapest, 2001.
- Bertalan, L.: Előadások Max Weberről: *Max Weber cselekvésselmélete és cselekvéstipológiája*. Szociológiai Szemle, 2001/3. pp. 113-136.
- Berg, D.J.-Fritzinger, J.S.: *JAVA felsőfokon*. JAVA 2 platform. Kiskapu Kft., Budapest, 1999.
- Boudon, R.: *Beyond Rational Choice Theory*. Annual Review of Sociology, Prefatory chapter of the volume 29, 2003.
- Brooks, R.A.: *Intelligence without Representation*. Artificial Intelligence, No 47, 1991, pp 139-159.
- Bui, T.-Lee, J.: *An Agent-based Framework for Building Decision Support Systems*. Decision Support Systems, 25, 1999, pp. 225-237.
- Bussmann, S.-Jennings, N.R.-Wooldridge, M.J.: *On the Identification of Agents in the Design of Production Control Systems*. in: Agent Oriented Software Engineering (eds. P. Ciancarini and M. J. Wooldridge), Springer Verlag, 2001, pp. 141-162.
- Chan, N.-LeBaron, B.-Lo, A.-Poggio, T.: *Agent-Based Models of Financial Markets: A Comparison with Experimental Markets*. MIT Artificial Markets Project, Paper No. 124, September 1999,

²² *CMU's Simon reflects on how computers will continue to shape the world. Monday, October 16, 2000 By Byron Spice, Post-Gazette Science Editor: Can computers help people act more rationally?*

- Conlisk, J.: *Why Bounded Rationality?* Journal of Economic Literature, Vol. XXXIV. (June 1996), pp. 669-700.
- Connah, D.-Wavish, P.: *An Experiment in Cooperation*. (Ed. Demazeau Y.-Müller J.P.), Decentralized AI-Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89), Elsevier Science Publishers B.V.: Amsterdam, 1990, pp 197-214.
- Druzdzal, M.J.-Flynn, R.R.: *Decision Support Systems*. In.: Encyclopedia of Library and Information Science, (Ed. Allen Kent), Marcel Dekker, Inc., New York, 2000.
- Dyer J.S., Edmunds T., Butler J., Jia, J.: *A Multiattribute Utility Analysis of Alternatives for the Disposition of Surplus Weapons-grade Plutonium*. ANRCP and U.S. Department of Energy (DE-FC04-95AL85832), Lawrence Livermore National Laboratory (W-7405-Eng-48), May 23 1997. [http://www.bus.utexas.edu/Faculty/Jim.Dyer/pu_or.PDF]
- Fekete, I.-Gregorics, T.-Nagy, S.: *Bevezetés a mesterséges intelligenciába*. LSI Oktatóközpont, Budapest, 1990.
- Futó, I. (szerk.) et al.: *Mesterséges intelligencia*. Aula Kiadó, Budapest, 1999.
- Gabaix, X.-Laibson, D.: *Bounded Rationality and Directed Cognition*. World Conference Econometric Society, Seattle, August 2000.
- Genesereth, M.R.-Nilsson N.: *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers: San Mateo, CA, 1987.
- Georgeff, M.P.: *Planning*. Annual Review of Computer Science, No 2, 1987, pp 359-400.
- Jennings, N.R.-Wooldridge, M.: *Applying Agent Technology*. Journal of Applied Artificial Intelligence, special issue on Intelligent Agents and Multi-Agent Systems, 1995.
- Jennings, N.R.: *Agent Software*. Proc. UNICOM Seminar on Agent Software, London, UK, 1995, pp. 12-27.
- Jennings, N.R.-Faratin, P.-Johnson, M.J.-Norman, T.J.-O'Brien, P.-Wiegand, M.E.: *Agent-based Business Process Management*. International Journal of Cooperative Information Systems, 5 (2&3), 1996, pp. 105-130.
- Jennings, N.R.-Wooldridge, M.: *Applications of Intelligent Agents*. in: Agent Technology Foundations, Applications, and Markets (eds. N.R. Jennings and M. Wooldridge), 1998, pp. 3-28. [<http://www.ecs.soton.ac.uk/~nrj/download-files/agt-technology.pdf>]
- Jennings, N.R.: *On Agent-based Software Engineering*. Artificial Intelligence, 117 (2), 2000, pp 277-296.
- Jennings, N.R.: *An Agent-based Approach for Building Complex Software Systems*. Communications of the ACM, Vol 44. No 4, April 2001, pp 35-41.
- Jones, B.D.: *Bounded Rationality*. Annual Review of Political Science, 1999, 2., pp. 297-321.
- LeBaron, B.: *Agent Based Computational Finance: Suggested Readings and Early Research*. Journal of Economic Dynamics and Control, 24, 2000, pp 679-702.
- March, J.G.: *Bevezetés a döntéshozatalba. Hogyan születnek a döntések*. Panem Kiadó, Budapest, 2000.
- McCarthy, J.: *What is Artificial Intelligence?* Computer Science Department, Stanford University, Stanford, CA 94305, 2002, [<http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>]
- McFadzean, D.-Tefatsion, L.: *An Agent-Based Computational Model for the Evolution of Trade Networks*, in Angeline, P.-Reynolds, R.-McDonnell, J.-Eberhart, R. (eds.):

- Evolutionary Programming VI, Proceedings of the Sixth International Conference on Evolutionary Programming, Springer-Verlag, Berlin, 1997, pp. 73-83.
- McFadzean, D.-Tsfatsion, L.: *A C++ Platform for the Evolution of Trade Networks*, Computational Economics 14, 1999, pp. 109-134.
- Nwana, H.S.: *Software Agents: an overview*. Knowledge Engineering Review, Vol 11, No 3, October/November 1996, pp 205-244.
- Parkes, D.C.: *Bounded Rationality*. [<http://www.cis.upenn.edu/~dparke/bounded.ps>; <http://citeseer.nj.nec.com/83317.html>]
- Parkes, D.C.-Huberman, B.A.: *Multiagent Cooperative Search for Portfolio Selection*. Games and Economic Behavior, 35, 2001, pp. 124-165, Special issue on Artificial Intelligence and Economics.
- Poh, K.: *Knowledge-based Guidance System for Multi-attribute Decision Making*. Artificial Intelligence in Engineering, vol 12., no 3., 1998, pp. 315-326.
- Russel, S.J.-Wefald, E.H.: *Do the Right Thing: Studies in Limited Rationality*. MIT Press, Cambridge, Massachusetts, 1991.
- Russel, S.J.-Norvig, P.: *Mesterséges intelligencia modern megközelítésben*, Panem-Prentice-Hall, Budapest, 2000.
- Shoham, Y.: *Agent-oriented Programming*. Technical Report STAN-CS-1335-90, Computer Science Department, Stanford University, Stanford, CA 94305, 1990.
- Simon, H.A.: *A Behavioral Model of Rational Choice*. The Quarterly Journal of Economics, Vol. LXIX., February, 1955., pp. 99-118.
- Simon, H.A.: *The New Science of Management Decision*. Harper & Row, New York, 1960., pp. 40-43.
- Simon, H.A.: *Rational Decision Making in Business Organizations*. The American Economic Review, Vol. 69., Issue 4 (September), 1979., pp. 493-513.
- Simon, H.A.: *Decision Making and Problem Solving. Report of the Research Briefing Panel on Decision Making and Problem Solving*. National Academy Press, Washington DC, 1986.
- Simon, H.A.: *Bounded Rationality and Organizational Learning*. Organization Science, Vol. 2, Num. 1, February 1991.
- Steels, L.: *Cooperation between Distributed Agents through Self Organization*. (Ed. Demazeau, Y.-Müller, J.P.), Decentralized AI-Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89), Elsevier Science Publishers, Amsterdam, 1990, pp 175-196.
- Tsfatsion, L.: *Agent-based Computational Economics: a Brief Guide to the Literature*. Iowa State University, Ames, Iowa, January 2000. [<http://www.econ.iastate.edu/tesfatsi/ace.htm>]
- Tsfatsion, L.: *Agent-based Computational Economics: Growing Economies from the Bottom up*. Iowa State University, Ames, Iowa, Economics Working Paper No 1, December 2001. [<http://www.econ.iastate.edu/tesfatsi/>]
- Van Roy, P.-Haridi, S.-Brand, P.: *Distributed Programming in Mozart – A Tutorial Introduction*, 2000, [<http://www.mozart-oz.org/download/mozart-ftp/store/1.2.5-2003-01-28-print/tutorial/DistributedProgramming.pdf.gz>]
- Van Roy, P.-Haridi, S.: *Concepts, Techniques, and Models of Computer Programming with Practical Applications in Distributed Computing and Intelligent Agent*. Draft, 2001, [<http://www.mozart-oz.org>]

-
- Weber, M.: *Gazdaság és társadalom. A megértő szociológia alapvonalai*. 1. Szociológiai kategórián. Közgazdasági és Jogi Könyvkiadó, Budapest, 1967. pp. 37-80.
- Wooldridge M.-Jennings N.R.: *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review 10(2), 1995. [<http://www.csc.liv.ac.uk/~mjw/pubs/ker95.ps.gz>]
- Wooldridge, M.: *Agents and Software Engineering*. AI*IA Notizie, Vol XI, No 3, September 1998, pp 31-37.
- Wooldridge, M.: Intelligent Agents. Ed.: Weiss, G.: *Multiagent Systems*, The MIT Press, April 1999.
- Wooldridge, M.-Ciancarini, P.: *Agent-oriented Software Engineering: The State of the Art*. Ed.: Ciancarini, P.- Wooldridge, M.: *Agent-Oriented Software Engineering*. Springer-Verlag Lecture Notes in AI Volume 1957, January 2001.