

MOLNÁR ISTVÁN

**Mikroszimulációs modellfejlesztési környezetek**

A tanulmány egy mikroszimulációs kutatás néhány eredményét írja le, s olyan új szoftvertechnológiákat elemez (közte az internet alapú alkalmazásfejlesztést), amelyeket mikroszimulációs modellek fejlesztéséhez lehet alkalmazni. Elsőként az alkalmazási területet, annak néhány jellemzőjét és követelményét mutatja be. A mikroszimulációs modellek rövid ismertetését és klasszifikációját követően a modellalkalmazások alapvető technológiai megközelítését vázolja fel. Majd két technológiát – az adatbázis-orientáltat, illetve az internetorientáltat – elemez részletesen. Az internetszolgáltatások realizációjának két jelentős megoldását a J2EE és a .NET hasonlítja össze, továbbá röviden felvázolja ezeknek a mikroszimulációra és a kapcsolódó területekre gyakorolt hatásait.\*

Journal of Economic Literature (JEL) kód: L68, L86.

A *mikroszimuláció* olyan módszer, amely képes komplex társadalmi-gazdasági rendszereket kezelni azáltal, hogy olyan *modellt alkot* és tanulmányoz, amely a vizsgált *modell-elemek* statisztikai adatainak intenzív használatán alapul. Ezek a modellelemek a társadalom vagy a gazdaság úgynevezett *mikroegységei*, a *személyek*, a *családok* vagy a *háztartások*. A mikroszimulációs modellekben azzal a céllal használják a szimulációs eljárásokat, hogy a mikroegységek időbeli viselkedését vizsgálják (*Orcutt és szerzőtársai* [1961]).

A mikroszimuláció a döntéshozók által általánosan elfogadott olyan módszer, amelyet széleskörűen használnak a politikai döntések előkészítése során Ausztráliában, Kanadában, az EU több tagállamában és az Egyesült Államokban (lásd *O'Donoghue* [2001]). Nemcsak a fejlett országok, de az átmeneti gazdaságok előtt is számos olyan megoldandó probléma áll, amely a demográfiai helyzettel, a nyugdíjrendszerrel, az egészségügyi ellátással és az adórendszerrel függ össze. A mikroszimuláció igen hasznos eszköz lehet a kapcsolódó problémák és a lehetséges megoldások modelleken alapuló tanulmányozására.

Két különböző mikroszimulációs modellosztályt fejlesztettek ki valósághű modellek kialakítására: az egyik az adatalapú, a másik az ágensalapú modellosztály. Az eltérő modellezési megközelítés ellenére mindkét modellosztály hasonlóan kezeli a modelladatokat és módszereket: mindkét esetben jelentős mennyiségű adatot kell elemezni és feldolgozni.

A mikroszimulációs modellek alkalmazásának egyik legfontosabb technikai problémája a mikroszimulációs modellek számára elérhető különféle adatforrások integrálása. Történelmileg három megközelítést dolgoztak ki:

\* A tanulmány a Nemzetközi Számítógépes Szimulációs Társaság (*The Society for Computer Simulation International [SCSI]*) 16. Európai Szimulációs Szimpóziumon (*European Simulation Symposium [ESS2004]*) (Budapest 2004. október 17–20.) elhangzott előadás írott változata.

- fájlfeldolgozós megközelítés (például *Heike és szerzőtársai* [1994]);
- adatbázis-orientált megközelítés (például *Sauerbier* [2002]);
- ágensalapú megközelítés (például *Pryor és szerzőtársai* [1996]).

Ezek a megközelítések hagyományos központi számítógépes vagy személyi számítógépes technológiát használnak, és ezek nem voltak hordozhatók, környezetfüggetlenek. A kilencvenes években új számítógéphálózat-orientált alkalmazásokat fejlesztettek ki azzal a céllal, hogy heterogén hardver- és/vagy szoftverplatformokat használó olyan alkalmazásokat is támogassanak, mint a modellalapú elemzések. Napjainkban a hálózatos multiplatform mikroszimulációs alkalmazások fejlesztése nem csupán szükséges, de egyben technikailag is lehetséges: a hálózatos adathozzáférés mellett az osztott feldolgozás (több, esetleg különböző helyen található, processzoron elvégzett számítás) is lehetséges.

Magyarországon az európai uniós csatlakozást követően egyre nagyobb igény mutatkozik a makrogazdasági elemzést és előrejelzést támogató eszközök iránt. Ez a tendencia párosul azzal, hogy mind több költségvetési forrás is rendelkezésre áll erre a célra. Sürgetően szükséges lenne megtalálni és/vagy kifejleszteni olyan fejlett mikroszimulációs szoftvereket, amelyek a következőkben felsorolt célok elérését szolgálják:

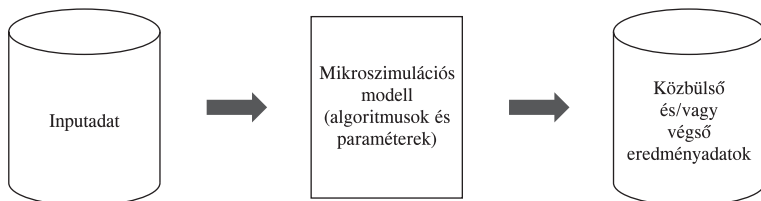
- hálózatorientált adat- és modellhozzáférés,
- osztott feldolgozás (modellfejlesztés és számítások),
- multiplatform hardver- és szoftvermegoldások,
- nyílt standardra épülő szoftvermegoldások,
- adat- és hálózatbiztonság,
- felhasználóbarát megoldások,
- hatékonyság a teljes szoftveréletciklus alatt.

### Mikroszimuláció

A mikroszimulációs modelleknek különböző adattartalmuk van: a *kiindulási*, a *közbülső* és/vagy a *végső szimulációs adat*. Ezeket az adatokat további elemzések céljából tároljuk. A mikroszimulációs modellek *viselkedését* (a modellállapotok időbeli változásait) algoritmusok segítségével írjuk le; az algoritmusok megjelenítik a mikroegységek környezetét és időbeli viselkedését. Külön figyelmet kell fordítani az *adatelemezésre*, továbbá a szimulációs *modellparaméterek* becslésére. A mikroszimulációs modell *kísérleti környezetben* működik, miután az a célja, hogy a változásoknak a modell mikroegységeire gyakorolt hatásait vizsgálja.

Az inputadatokat, a közbülső és/vagy végső eredményadatokat figyelmesen kell elemezni. Különleges technikákat fejlesztettek ki az adatminőség javítására (például *imputing*, *merging*, szintetikus adatok kezelése stb.). A modellek ellenőrzésre és érvényesítésére szintén különféle eljárások, módszerek állnak rendelkezésre (lásd *Rubin* [2004], *Little–Rubin* [2002], *Schofield–Polette* [1998]).

1. ábra  
Mikroszimulációs modell



## Technológiai lehetőségek

Figyelembe véve a rendelkezésre álló hálózatorientált technológiákat, azt látjuk, hogy a hálózati indíttatású mikroszimulációs modelleket különböző technológiák felhasználásával is ki tudjuk fejleszteni. Az információtechnológiai terület főáramlatait tekintve két alapvető megközelítést lehet megkülönböztetni:

- adatbázis-orientált megközelítés,
- internetszolgáltatáson alapuló megközelítés.

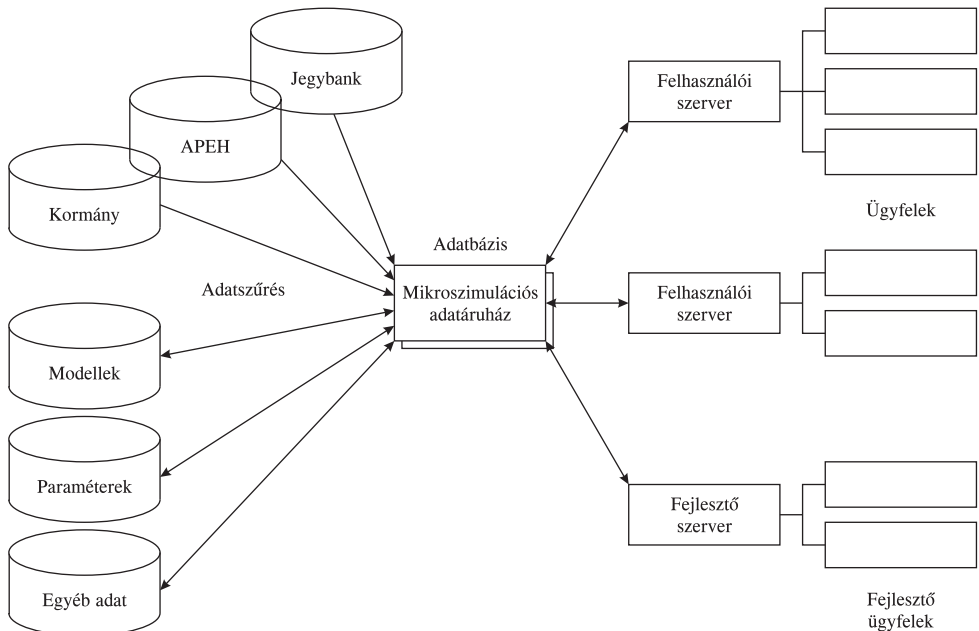
### Adatbázis-orientált megközelítés

Az adatbázis-orientált megközelítés alapja a relációs adatbázis-kezelési rendszerek (RDBMS) technológiája: a modelleket relációs adatbázis környezetbe helyezzük, és ebben a környezetben használjuk azokat a különféle analitikus eszközök és technológiai standardok segítségével (matematikai és/vagy statisztikai programcsomagokkal, fejlett felhasználói kontrollal és interfésszel, valamint internettechnológiákkal).

A hálózatorientált RDBMS lehetőséget kínálnak fejlett mikroszimulációs alkalmazások kifejlesztésére a 2. ábrán ábrázolt felépítés szerint. Ez a felépítés az adatkezelésre helyezi a hangsúlyt, és a hálózaton keresztül multiplatformos hozzáférést tesz lehetővé a mikroszimulációs adatokhoz. Ezen adatok felhasználása nem „különleges”, inkább „megszokott” adatbázis-felhasználásnak minősül.

Számos olyan szoftvertermék van kereskedelmi forgalomban, amely képes átvállalni a hálózati adatbázis-kezelés feladatát (például az *Oracle Application Server* [2003]). A Java segítségével ki lehet fejleszteni mikroszimulációs algoritmusokat; különleges matematikai és statisztikai szoftvereszközökkel pedig el lehet készíteni az adatelemzést és

2. ábra  
Adatbázis-orientált felépítés



-paraméterbecslést (például SAS, SPSS). Ennek a technológiának az egyes elemei széles körben hozzáférhetőek, igen elterjedtek, sokszorosan kipróbáltak, ipari standardokat használnak, és forgalmazósemleges, platformfüggetlen megoldások.

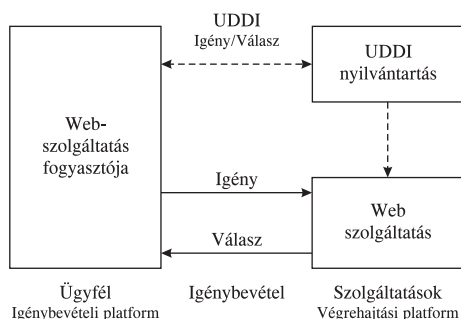
### *Internetszolgáltatáson alapuló megközelítés*

Az internetszolgáltatásokon alapuló modell laza felépítésű; különféle, együtt futó szoftverösszetevőkből áll. Az internetszolgáltatás igénybevétele előtt a fogyasztónak először meg kell találnia a szükséges szolgáltatást kínáló alkalmazást, fel kell lelnie az interfészt és olyan módon kell konfigurálnia a szoftvert, hogy az együtt tudjon működni az interfész segítségével az internetszolgáltatással.

Az internetszolgáltatás átmenet a szolgáltatásorientált, komponensalapú osztott alkalmazások felé. Az internetszolgáltatások olyan alkalmazások, amelyek jól meghatározott interfészekkel ellátott, speciálisan kialakított interneten alapuló elemeket használnak, amelyek bizonyos funkcionalitást kínálnak az ügyfeleknek az interneten keresztül. Telepítésük után a fogyasztók (ügyfelek, más szolgáltatók vagy felhasználók) az internetszolgáltatásokat felfedezik, használják és újra használják, mégpedig blokkszerűen építkezve a nyílt ipari standard protokollok, forgalmazótól független specifikációk alkalmazásával. A szolgáltatásokat bármely programozási nyelven fel lehet használni, majd a felhasználói rendszert bármely operációs rendszerrel vagy szoftverplatformmal működtetni lehet. Az internetszolgáltatások szoftver felépítését a 3. ábra mutatja be.

3. ábra

Az internetszolgáltatás felépítése



Az internetszolgáltatások felhasználása nyílt standardokon alapul, amelyeket egy sokszereplős konzorcium irányít. Az általános leírás, feltárás és integrációs (*UDDI, Universal Description Discovery and Integration*) protokoll segítségével történik az internetszolgáltatások kiadása, fellelése és az, hogy azokat összekapcsolják az alkalmazói szoftverekkel. A fogyasztó által igényelt szolgáltatást a szolgáltató és a szolgáltatást igénybe vevő ügyfél közötti szerződés határozza meg. A szerződést a WSDL web-szolgáltatást leíró nyelv (*Web Service Description Language*) alapján lehet megszövegezni. Az egyszerű objektumelérést biztosító protokoll (*SOAP, Simple Object Access Protocol*) és a hiperszöveg átalakító protokoll (*HTTP, Hypertext Transfer Protocol*) használatával a felek egy közös üzenetben és protokollban állapodnak meg. A kommunikációk során alkalmazott adatsere-formátumot is standardizálták, a használt nyelv az XML (*Extensible Markup Language*).

Jelenleg két jelentős platformot használnak az internetszolgáltatások fejlesztéséhez, a Microsoft .NET-et és a J2EE-t.

Ezek a technológiai fejlesztések alapvetően megkérdőjelezik és/vagy kiterjesztik a korábbi hálózatorientált szimulációs technológiákat (például *Miller és szerzőtársai* [1998], [2001]). A HLA (*High Level Architecture*) alapú megoldások (például *Lantzsch és szerzőtársai* [1999]) tovább tágítják az internetszolgáltatások kínálta lehetőségeket, és egyben lehetővé teszik a felhasználók számára nagy, multiplatform, hálózatorientált mikroszimulációs modellek kifejlesztését. Sajnálatos módon azonban jelenleg még nincs elegendő tapasztalat ezen a téren. Az egyértelmű előnyök bizonyára újabb vonzerőt jelentenek majd a jövőben a szimulációs szoftverfejlesztők számára, és ennek nyomán az internetszolgáltatások egy, nemcsak lehetséges, de igen előnyös alternatívát jelentenek majd a mikroszimulációs szoftverfejlesztők számára is.

**A Microsoft .Net technológia.** A Microsoft .Net egy új Windows-platform az internetszolgáltatások fejlesztésére és telepítésére. A platform új és régebbi eszközök és megközelítések kombinációja. A .Net-et az XML-alapú internetszolgáltatások köré tervezték és így az XML-optimalizált. A .Net a platformon található valamennyi programozási nyelvre kiterjesztette a Visual Basic paradigmát; bármely nyelv felhasználható a fejlesztési környezetben, s ezek kölcsönösen is működnek. A platform magában foglal egy futtatásiidő-környezetet, amit *közös nyelvi futtatási időnek* (*Common Language Runtime, CLR*) neveznek. Ez a környezet igen hasonlatos a Java virtuális géphez, a Java VM-hez, és úgy utalnak rá, mint egy „kezelt végrehajtoi környezetre”, ami lehetővé teszi az összetett programok bármely platformon történő futtatását. A keretrendszerhez tartoznak még más internetszolgáltatásokkal összefüggő elemek, mint az ASP.NET és a Microsoft internet információs szolgáltatásai (*Microsoft Internet Information Services, IIS*).

**A Java technológia.** A Java technológiák közé tartozik a Java programozó nyelv, a futtatásiidő-környezet, a platformváltozatok és az alkalmazásprogramozási interfész (*Application Programming Interface, API*). A jelenlegi platform változatok a következők: J2ME (mikro), J2SE (standard), J2EE (vállalati), amelyek mindegyike tartalmazza a Java nyelvet, a hordozható bájtkóddal, valamint a magsoftvert. A J2EE applikációs szerver vezető forgalmazói az IBM, a Sun, a Borland, az Oracle és a Macromedia. A J2EE számos platformon fut, de csak a Java nyelvet támogatja.

A J2EE-alapú internetszolgáltatási környezetben az eltérő forgalmazók által kínált különféle szoftverelemeknek együtt kell működniük, ami nehezítheti a programfejlesztést. Éppen ezért a fejlesztéshez nagyobb professzionális figyelemre és fegyelemre van szükség, mivel a folyamat- és a kódgeneráció nem igazán automatikus folyamat, mint például a .NET esetében. A programozói hatékonyság növelése érdekében erőteljesen javasolt egy olyan integrált szoftverfejlesztői környezet (*Integrated Development Environment, IDE*) használata, mint például az IBM Web-Sphere vagy a Sun One.

**A .NET- és a Java-technológiák összehasonlítása.** A döntéshozók általában a következő kritériumok alapján hasonlítják össze és értékelik az alkalmazásprogramozási szoftvereket: TCO, szoftverteljesítmény, fejlesztési és egyéb lehetőségek (*CGI* [2002]). A szoftverfejlesztő szempontjából az érvek és a vélemények is nagyon fontosak (lásd *Benchmark comparison* [2001] és *Middleware* [2003]).

A .NET-et az XML internetszolgáltatások platformjaként tervezték, de az internetszolgáltatási technológiákat még nem standardizálták a J2EE-ben. Mindezen tények ellenére a .NET és a J2EE tud együttlétezni, sőt, kell is nekik, és együtt fognak létezni a jövőben is. Azok a vállalkozások vagy kormányzati intézmények, amelyek el-

utasítják az egyedüli forgalmazót vagy a magas szintű megbízhatóságot, a biztonságot és a stabilitást tartják kizárólagos szempontnak, elkerülhetik a .NET-et, de ebben az esetben elveszítik a .NET platform által kínált előnyöket. A döntések következményei hosszú távúak, s pénzügyi-gazdasági hatásai nagyok.

**Az internetszolgáltatások előnyei és hátrányai.** A mikroszimulációs szoftverek fejlesztésének szempontjából a hangsúly a számítógépes modellfejlesztésen van. A mikroszimulációs algoritmusokat magas szintű programnyelveken (például Java) fejleszthetjük, míg az adat-elemzést és paramétermeghatározást matematikai-statisztikai programcsomagok segítségével végezhetjük (például SAS, SPSS). Az internetszolgáltatások mindkét fő komponens használatát támogatják. Hálózati környezetben működő relációs adatbázis-kezelő rendszerek által nyújtott támogatásokat az internetszolgáltatások keretrendszeréből magas szintű programnyelvek segítségével is fel lehet használni.

Ennek a megközelítésnek jelentős előnye, hogy a technológiának az egyes elemei platformfüggetlenek és széles körben hozzáférhetők.

Az internetszolgáltatások használatának *előnyei*:

- régi rendszerek is integrálhatók, a már meglévő programkód újrafelhasználható,
- a szoftverfejlesztési, -karbantartási és -működtetési költségek csökkenthetők,
- új üzleti modellek alakíthatók ki, új bevételt lehet elérni, miközben az üzleti partnerekkel és az ügyfelekkel tovább lehet javítani a kapcsolatot.

Az internetszolgáltatások használatának *hátrányai*:

- a hálózat teljesítménye nem garantálható,
- néhány standard hiányzik, illetve kidolgozásán jelenleg még dolgoznak (például biztonsági, azonosítási, számlázási és szerződési standardok),
- a koncepció bonyolult, nehéz elsajátítani.

**Szervizorientált architektúrák.** A szervizorientált architektúrák (*Service Oriented Architecture, SOA*) az olyan lazán kapcsolt szoftverkomponensek alapelveire épülnek, amelyeket szolgáltatásként vesznek igénybe (azaz a szoftverkomponensek szervizt nyújtanak). A szervizorientált architektúrák háromfajta szerepet tartalmaznak (szerviz-igénybevevő, szervizszolgáltató és szerviz-nyilvántartás), amelyek mindegyike megvalósítható egy számítógépes hálózat egy csomópontja, illetve programja által (egy egyszerű program akár több szerepet is betölthet!). Valamennyi szoftverfunkciót olyan szervizként valósítanak meg, amelyet fel lehet használni az interneten keresztül. Az internetszolgáltatások architektúrája egyfajta szervizorientált architektúra.

Szervizorientált architektúrák esetében a szoftverfejlesztés során a fő hangsúly a jól definiált interfészekkel ellátott szoftverkomponensek segítségével megvalósított alkalmazásfejlesztésen van, aminek következtében a szoftverrendszer integrációja az interfészek szintjén megvalósítható, s nem kell azt a komponensek megvalósításának szintjén elvégezni. A szervizorientált architektúrák alkalmazása megváltoztatja a fejlesztők „világnézetét”, és egyben a megvalósítható alkalmazások számát is megnöveli, tovább növelve ezzel a szoftverfejlesztés hatékonyságát. Mivel a régi fejlesztésű rendszereket is át lehet alakítani szolgáltatásorientálttá, rugalmas intézményi szoftver-infrastruktúra alakítható ki.

### Következtetések

A mikroszimuláció iránti növekvő érdeklődés feltétlenül szükségessé teszi Magyarországon a jelenlegi gyakorlat felülvizsgálatát, és az új projektekhez alkalmazható új technológiák megismerését, alkalmazását (Molnár [2004]). Azokat a szoftvermegoldásokat cél-

szerű előnyben részesíteni, amelyek támogatják a felhasználói együttműködést, nemcsak a programfejlesztés, de az erőforrások (számítástechnikai, adat- és emberi) használata területén is. A tudományos eredmények nyújtotta előnyök teljes kihasználása csakis úgy lehetséges, ha a közgazdasági modellezés kellő figyelmet fordít a kiszámíthatóságra, valamint a technikai megvalósításra, és ezeken a területeken is a legújabb tudományos eredményeket használja fel; a tudás ugyanis a végtermékben testesül meg, s annak létrehozása és felhasználása során jön létre.

A hagyományos mikroszimulációs modellek internetalapúra konvertálása eredményeként lehetővé válik azok hatékony használata a modern integrált információs rendszerekben. Az új technológiák felhasználóbarát módon kínálnak kiváló eszközöket az új modellfejlesztések számára; mind az oszttmodell-fejlesztés, mind az oszttmodell-számítás, mind pedig a közös adathozzáférési problémák megoldhatók (Molnár [2005]).

Az internet-indíttatású mikroszimulációs modelleket olyan interneten alapuló modellbankok fejlesztésére lehet használni, amelyek különböző típusú mikroszimulációs modellekből állnak és interneten alapuló döntést támogató rendszereket (DSS) eredményeznek. Ezek a rendszerek az államigazgatásban és a tudományos kutatásban egyaránt jól használhatók. A felvázolt szoftverkörnyezetek általában elérhetők a mai információtechnológiai környezetben.

### Hivatkozások

- BENCHMARK COMPARISON [2001]: Benchmark comparison. Building XML-based Web Services in Microsoft .NET vs. IBM WebSphere 4.0, Revision 2.0., december.
- CGI [2002]: Microsoft .NET or Java 2 Enterprise Edition: Is it just a question of platforms and languages? White Paper, CGI Group Inc., szeptember.
- HEIKE H.-D.–BECKMANN K.–KAUFMANN A.–SAUERBIER T. [1994]: Der Darmstädter Mikro-Makro-Simulator–Modellierung, Software Architektur und Optimierung. Megjelent: *Faulbaum, F.* (szerk.): *SoftStar'93 – Advances in Statistical Software 4*. Fischer, Stuttgart–New York, 161–169 o.
- LANTZSCH, G.–STRASSBURGER, S.–URBAN, C. [1999]: HLA-basierte Kopplung der Simulationssysteme Simplex3 und SLX, Megjelent: *Proceedings der Tagung Simulation und Visualisierung '99 der Otto-von-Guericke-Universität Magdeburg* (Kiadja: *Deussen, O.–Hinz, V.–Lorenz, P.*), Institut für Simulation und Graphik, március 4–5., Magdeburg.
- LITTLE, R. J. A.–RUBIN, D. B. [2002]: *Statistical analysis with missing data*. Wiley-Interscience, New York.
- MIDDLEWARE [2003]: J2EE and .NET. Yet Another Performance Case Study. The Middleware Company Application Server Baseline Specification, <http://www.middleware-company.com/casestudy/>, The Middleware Company Case Study Team, június.
- MILLER, J.–FISHWICK, P.–TAYLOR S.–BENJAMIN, P.–SZYMANSKI, B. [2001]: Research and Commercial Opportunities in Web-Based Simulation. *Simulation Practice and Theory*. Special Issue on Web-Based Simulation. Vol. 9, No. 1–2. október, Elsevier Science, 55–72. o.
- MILLER, J.–GE, Y.–TAO, J. [1998]: Component-Based Simulation Environments: JSIM as a Case Study Using Java Beans. Megjelent: *Proceedings of the 1998 Winter Simulation Conference*, Washington DC., december, 373–381. o.
- MOLNÁR ISTVÁN [2004]: A mikroszimulációs modellek használatának új hazai lehetőségei. *Statisztikai Szemle*, 82. évf. 5. sz. 462–477. o.
- MOLNÁR ISTVÁN [2005]: Microsimulation Model Development Environments. *International Journal of Simulation, Systems Science and Technology*, United Kingdom Simulation Society, Vol. 6. No. 5.
- O'DONOGHUE, C. [2001]: Dynamic micro-simulation: A methodological survey. *Brazilian Electronic Journal of Economics*, Vol.4, No. 2.
- ORACLE APPLICATION SERVER [2003]: Oracle Application Server 10g J2EE and Web Services. White Paper, Oracle, szeptember.

- ORACLE APPLICATION SERVER PORTAL [2003]: Oracle Application Server Portal 10g. Technical Overview. White Paper. ORACLE, szeptember.
- ORCUTT, G.–GREENBERGER, M.–KORBEL, J.–RIVLIN, A. [1961]: Microanalysis of socioeconomic systems: a simulation study. Harper and Brothers. New York.
- PRYOR, R.–BASU, N.–QUINT, T. [1996]: Development of Aspen: A microanalytic simulation model of the US economy. Sandia Report, SAND96-0434. Sandia National Laboratories, Albuquerque, NM, február.
- RUBIN, D. B. [2004]: Multiple Imputation for Nonresponse in Surveys. John Wiley and Sons. New York.
- SAUERBIER, TH. [2002] UMDBS. A New Tool for Dynamic Microsimulation. Journal of Artificial Societies and Social Simulation, Vol. 5. No. 2.
- SCHOFIELD, D.–POLETTE, J. [1998]: A comparison of data merging methodologies for extending a microsimulation model. NATSEM STINMOD Technical Paper, No. 11. National Centre for Social and Economic Modelling, University of Canberra.