



# Parameter Sweep Eszköz

*Felhasználói kézikönyv*

*Készítette:*

*Bocsi Rajmund, Gulyás László  
és Iványi Márton Dávid  
Az ELTE-IKKK részére.*



*A projekt az EU társfinanszírozásával, az Európa Terv keretében valósul meg.*

*Budapest 2007*

## Tartalom

Bevezető	3
Ágens-alapú modellezés	3
MASS	3
Parameter Sweep Tool és Monitor	4
1    Parameter Sweep Tool	5
1.1    Preferences	5
1.1.1    Általános beállítások	6
1.1.2    Elosztott futtatás beállítása	7
1.2    Modell kiválasztása	8
1.2.1    Bővíthető Class path	8
1.2.2    Erőforrásfájlok	9
1.2.3    Beállítások betöltése	10
1.3    Paramétertér beállítása	11
1.3.1    A paraméterfa	12
1.3.2    Paraméter beállítások módosítása	12
1.3.3    Új paraméterek hozzáadása	13
1.4    Adatgyűjtési beállítások	14
1.4.1    Leállási feltétel	15
1.4.2    Adatrögzítők (rekorderek)	15
1.4.2.1    Eredmények kiírása	16
1.4.2.2    Rögzítendő elemek hozzáadása	16
1.5    Scriptelés	16
1.5.1    Statisztikák	17
1.5.2    Haladó scriptelés	19
1.6    Beépített monitor	20
2    A Monitor alkalmazás	22
2.1    Az alkalmazás konfigurálása	22
2.2    Az aktuálisan futó szimuláció adatai	23
2.3    Eredmények letöltése	24
3    Gyakorta ismétlődő kérdések	25
4    Hibaelhárítás	26
5    Összefoglalás	27
6    Referenciák	28

# Bevezető

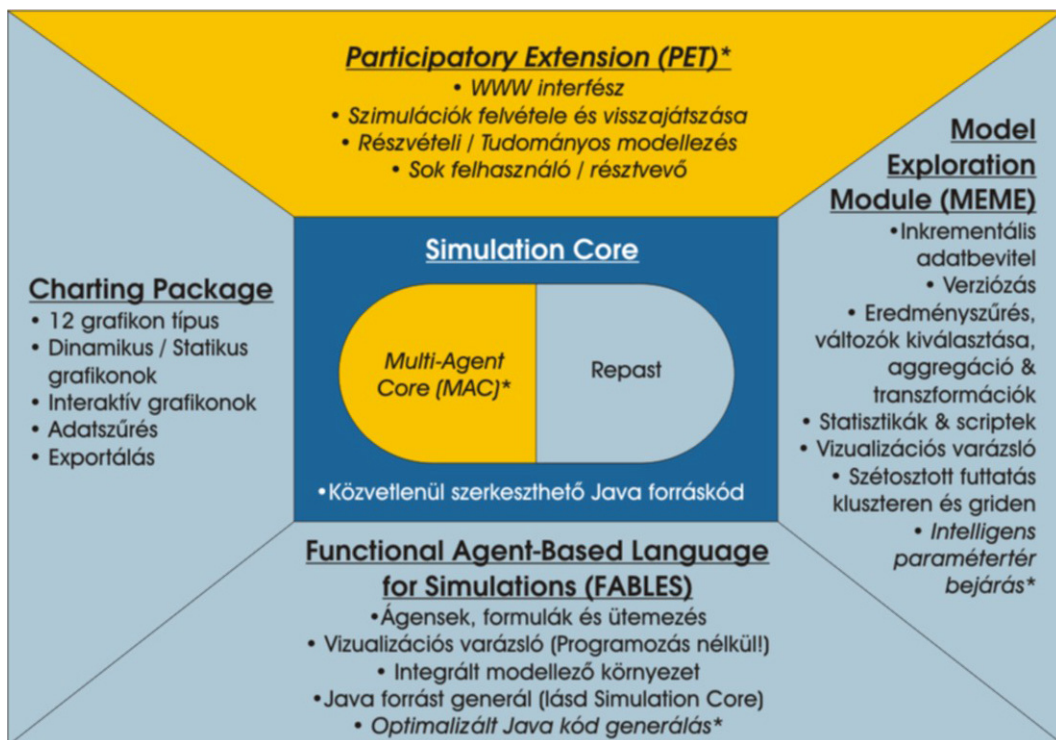
## Ágens-alapú modellezés

Az ágens-alapú modellezés a számítógépes szimulációk egy - komplex társadalmi rendszerek modellezésére különösen alkalmas - új ága. Alapvetése az, hogy az egyént modellezzük, tökéletlenségeivel (pl. korlátozott kognitív és számítási képességek), egyéni jellegzetességeivel és egyedi interakcióival együtt. A modell tehát "alulról felfelé" épül - elsősorban a mikro szabályokra koncentrálva, de a makró jelenségek kialakulását kutatva. A részvételi szimuláció, mint az ágens-alapú szimuláció alfaja egy olyan metodológia, amely az emberi szereplők és a mesterséges ágensek együttműködésére alapoz. Ezek a megoldások a képzési és döntés-támogatási területeken igen hasznosak.

## MASS

A Multi-Agent Simulation Suite (MASS) egy szoftvercsomag, amely lehetővé teszi a felhasználó számára, hogy az ágens-alapú modellezés megoldásait változatos területeken alkalmazza anélkül, hogy komoly programozási ismereteket kelljen elsajátítania.

A programcsomag négy, egy ún. szimulációs mag köré szerveződő alkalmazásból áll össze. A MASS rendelkezik egy saját maggal (ez a MAC), illetve képes futni Repast alapokon is. A több szimulációs magon való futtathatóság biztosítja, hogy a modellek mag-függetlenek legyenek, így a használható magok számát a jövőben bővíteni kívánjuk. A Functional Agent-Based Language for Simulation (FABLES) egy olyan programozási nyelv és modellezési környezet, amely kifejezetten ágens-alapú modellek fejlesztését szolgálja. A Model Exploration Module (MEME) paraméter terek bejárását, a kinyert adatok feldolgozását és megjelenítését hivatott támogatni. A Participatory Extension (PET) egy opcionális web-alapú környezet multi-ágens és részvételi szimulációk futtatásához. A MASS negyedik része, a Vizualizációs Csomag, nem jelenik meg önálló programként, a többi szoftverben használt grafikonok és vizualizációk implementációit tartalmazza.



1. ábra - Multi-Agent Simulation Suite

## Parameter Sweep Tool és Monitor

A MEME Parameter Sweep Tool lehetővé teszi Repast felhasználók számára, hogy egy grafikus felületet használva érzékenység vizsgálatokat, paramétertér-keresést végezzenek helyi számítógépen, vagy elosztott módon, egy griden vagy számítógép klaszteren. A felhasználók elkészíthetik létező Repast modelljeik ún. batch-változatát, amely alkalmas paramétertér-keresésre, mivel van benne adatgyűjtés (ún. rekorderekkel), és futtassák azt. A program segítségével úgy lehet elosztott módon paramétertér-keresést végrehajtani, hogy kifejezetten batch-módú futásra írt modellt nem kell készíteni.

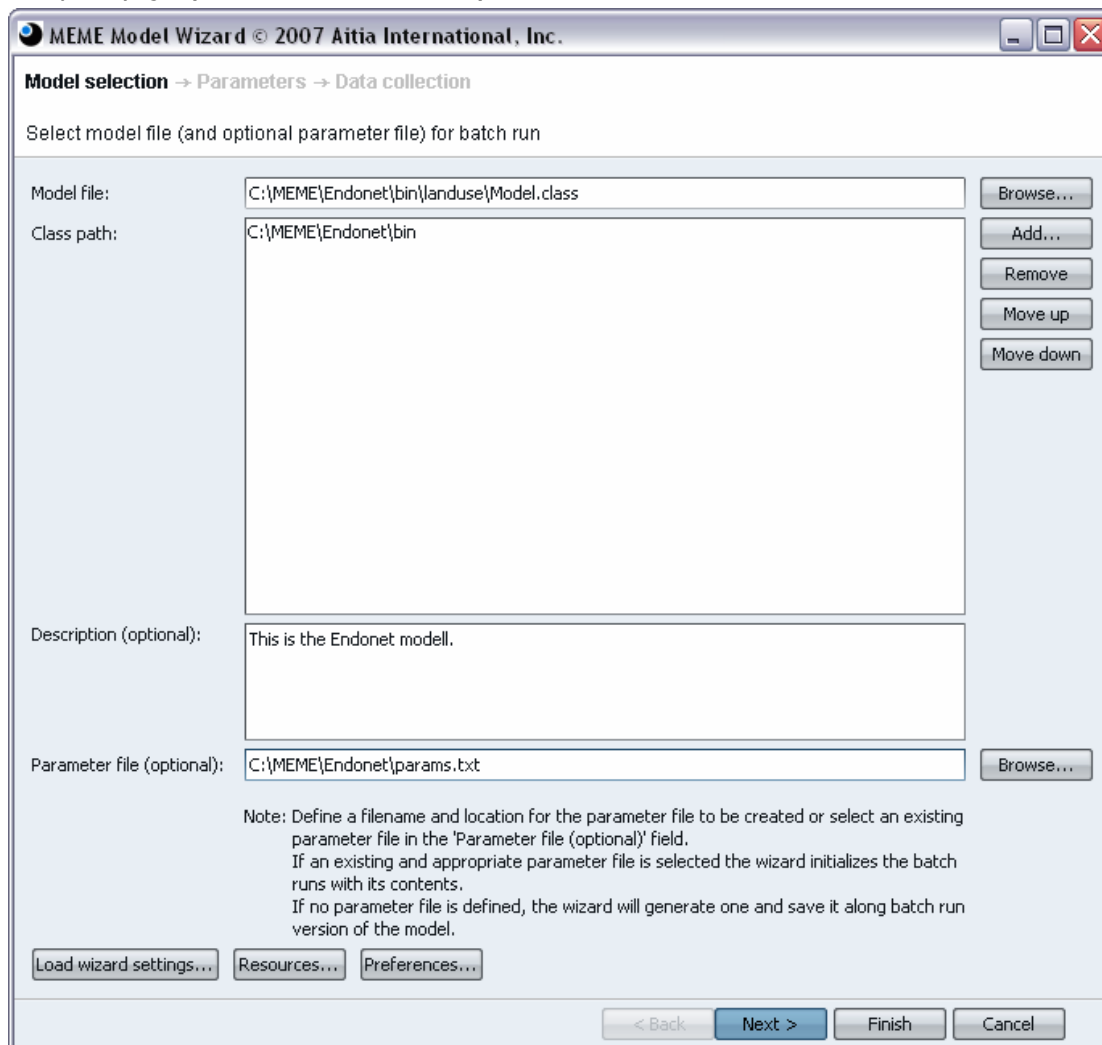
A MEME Monitor alkalmazás használatával a felhasználók nyomon követhetik egy adott griden vagy klaszteren éppen futó szimuláció aktuális állását (az aktuális futás számát), és letölthetik a befejezett szimulációk eredményeit.

A MEME Parameter Sweep Tool és Monitor jelenlegi verziója egy első verzió, ami csak korlátolt, a jövőben bővítendő funkcionalitást biztosít. A következő fejezetek ismertetik ezen funkciók használatát.

Az eszköz a fejlesztés következő lépésében integrálásra kerül a Model Exploration Module többi programrészével, annak beépülő moduljaként fog működni.

# 1 Parameter Sweep Tool

A MEME Parameter Sweep Tool segítségével a felhasználó három egyszerű lépésben elkészítheti Repast modelljének batch futtatásra alkalmas változatát, majd saját gépén, vagy rendelkezésére álló gépek hálózatán futtathatja azt. A következő szakaszok részletesen ismertetik az ehhez szükséges lépéseket. Az alábbi ábrán a program nyitóképernyője (*Model selection* oldal) látható.

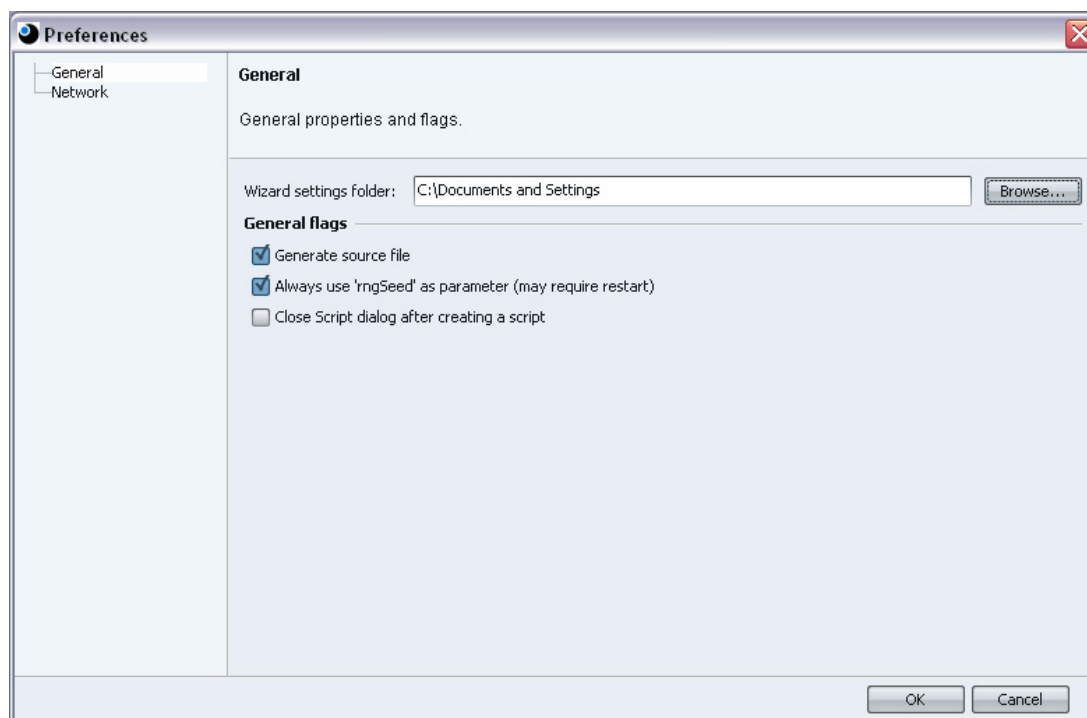


**2. ábra – Model selection oldal**

A program a modelleket alapértelmezésben helyi gépen futtatja. Elosztott módú futtatáshoz a *Preferences* párbeszédablakban módosítani kell az alapértelmezett beállításokat. A program egyéb beállítási lehetőségei szintén itt találhatóak. A *Preferences* párbeszédablak megnyitásához kattintson a *Preferences...* gombra a nyitóképernyőn.

## 1.1 Preferences

A *Preferences* menü két lapból áll. Az első, *General* oldalon az általános, míg a *Network* oldalon az elosztott futtatást érintő beállításokat lehet változtatni. Az oldalak között váltani a nevükre való kattintással lehet, a baloldalt megjelenő fában.



**3. ábra – Preferences ablak, General oldal**

Az *OK* gomb megnyomásával lehet alkalmazni a megváltozott beállításokat. Ha a program hibát talál a beállításokban, akkor figyelmeztető szöveg jelenik meg az információs panelen az oldal címe alatt.

### 1.1.1 Általános beállítások

A következő általános beállítások adhatóak meg a *General* oldalon:

*Wizard settings folder:* A modellek futtatása során a program automatikusan elmenti az összes, a futtatott modellre vonatkozó beállítást (pl. paraméterfájl, rekorderek, scriptek, stb.). Ez lehetőséget ad arra, hogy újra felhasználja ezeket a beállításokat (lásd az 1.2.3 Beállítások betöltése szakaszt a részletekért!). Ebben a mezőben adhatja meg, hogy hová mentse az eszköz ezeket a beállítás-leírófájlokat. Ez a hely alapértelmezésben a felhasználói könyvtár<sup>1</sup>.

*Generate source file:* Ha bejelöli a négyzetet, az eszköz minden generált modell mellé elkészíti annak forráskódját tartalmazó fájlt is. A forrásfájl ugyanabba a könyvtárba kerül, ahol az eredeti modell fájl (.class kitejesztéssel) található.

*Always use 'rngSeed' as parameter:* Az opció bejelölése esetén a *random seed*-et (véletlen-szám generátort inicializáló érték) paraméterként kezeli a program, akár tartalmazza ilyen formában az eredeti modell, akár nem. (A paramétereiről részletesen az 1.3 Paramétertér beállítása szakaszban, az *rngSeed*-ről és a kapcsolódó *getInitParam()* metódusról pedig a Repast dokumentációban [1] olvashat.)

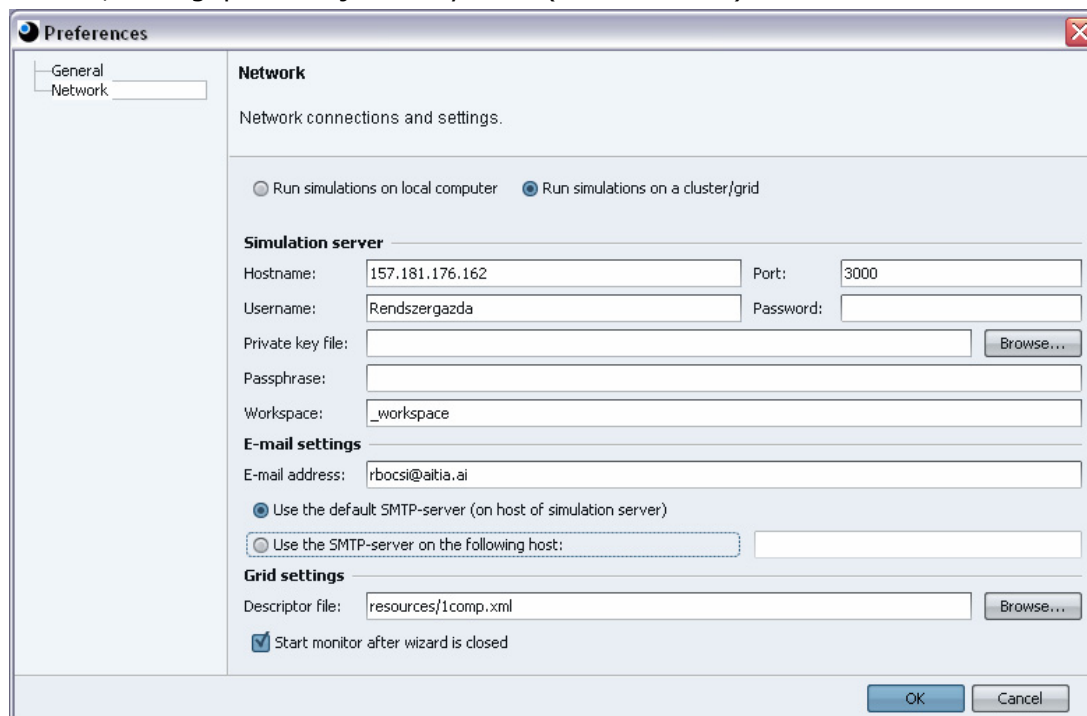
**Tipp: A fenti beállítást csak a modell kiválasztása előtt változtassa!**

*Close script dialogue after creating a script:* Azt határozza meg, hogy a Script párbeszédablak (lásd 14. ábra!) automatikusan bezáródjon-e minden script létrehozása után vagy sem. Ez utóbbi esetben a párbeszédablakot a felhasználónak kell bezárnia a *Close* gomb megnyomásával (bővebben lásd az 1.5 Scriptelés szakaszt).

<sup>1</sup> Alapértelmezésben ez a C:\Documents and Settings\

## 1.1.2 Elosztott futtatás beállítása

A *Network* oldalon beállítható, hogy a futásokat helyi szinten, egy gépen, vagy elosztottan, több gépen akarjuk lefolytatni. (Lásd 4. ábra.)



4. ábra – Preferences ablak, Network oldal

A *Run simulations on local computer* (helyi futtatás) opció az alapértelmezett. Elosztott módba a *Run simulations on a cluster/grid* lehetőség kiválasztásával lehet váltani.

*Simulation server*: Elosztott módban a program egy szerveralkalmazáshoz (amit *szimulációs szervernek* nevezünk) kapcsolódva futtatja a szimulációkat. A szerverkapcsolat beállításait *Simulation server* szakasz írja le:

- *Hostname*: A gép neve (vagy IP címe), ahol a szimulációs szerveralkalmazás található;
- *Port*: A szimulációs szerver portja (alapértelmezés: 3000-es port);
- *Username/Password*: A program a modellel kapcsolatos fájlokat SFTP-protokollon keresztül juttatja el a szerverre. Ehhez alapesetben felhasználónév és jelszó páros megadására van szükség.
- *Private key*: Némely SFTP-szerverek nem támogatják a jelszó alapú azonosítást, helyette ún. privát kulcsot használnak, melyeket egy fájl tartalmaz.
- *Passphrase*: A private key-t használó SFTP-szerverek egy része egy további jelszót igényel, melyet *passphrase*-nek neveznek.
- *Workspace*: A szerver munkakönyvtára (a távoli gépen).

**Figyelmeztetés: A munkakönyvtárat az SFTP-szerver alapértelmezett felhasználói könyvtárához relatívan kell megadni.**

*E-mail settings*: Elosztott módban lehetőség van e-mail értesítések kérésére a szimulációkkal kapcsolatos fontosabb eseményekhez (befejezés, hiba, megszakítás, stb.) kapcsolódóan. Ehhez a szimulációs szerver az *E-mail settings* szakaszban megadott beállításokat (e-mail cím, szimulációs szerveren található SMTP szerver/más SMTP szerver) használja. Ezen beállítások megadása nem kötelező.

*Grid settings*: A szimulációs szerveralkalmazás a ProActive grid-megoldást használja az elosztott futtatáshoz (lásd a ProActive dokumentációját [2]), melyhez meg kell adni egy klasztert leíró XML-fájlt. Ez a fájl minden számítógép-infrastruktúra esetén valamelyest különböző (a részleteket lásd [5]-ben).

Az <alkalmazás könyvtára>/resources<sup>2</sup> mappában található az *1comp.xml*. Ez a fájl érvényes minden ProActive klaszterre, azonban csak egyetlen gépet ír le (egy egy gépből álló klasztert). Hasznos lehet, ha a szimulációkat egy távoli gépen akarjuk futtatni a helyi gép helyett, de ennek alapján elkészíthető a több gépet leíró fájl is.

**Megjegyzés: Mindezen beállítások pontos értékeit, beleértve a megfelelő grid leírófájlt, az adott klaszter rendszergazdája biztosítani tudja.**

*Start monitor after wizard is closed*: Az opció bejelölése esetén, elosztott futásnál a program automatikusan elindítja a Monitor-t, melynek segítségével a futási eredmények nyomon követhetőek. Bővebben lásd 2 A Monitor alkalmazás fejezetet!

## 1.2 Modell kiválasztása

A program nyitó oldalán (lásd 2. ábra!) lehet kiválasztani a futtatni kívánt modellt (a modellhez tartozó class fájlt) a *Model file*: sorban. Ez lehet egy egyszerű modell (megjelenítésre és/vagy batch futtatásra vonatkozó specifikus kód nélkül), vagy egy batch-módú modell. Ha a kiválasztott modell nem batch-módú, a program segít az ilyen jellegű futtatáshoz szükséges adatrekorderek beállításában. Emellett az eszköz segítséget nyújt a paramétertér elkészítésében (az egymásba ágyazott paraméterfa, amely leírja a vizsgálni kívánt paraméterteret) és a modell futtatásában, helyi gépen, egy klaszteren vagy griden.

Amennyiben a futtatni kívánt modell további java osztályokat használ, meg kell adni azok elérési útját is a *Class path*: listában. Az a könyvtár, ahol a kiválasztott modell osztály van (a csomagstruktúrát alapul véve) automatikusan bekerül az elérési utak listájába, a *Class path*-ba. A lista bővítéséhez nyomja meg az *Add...* gombot és válasszon ki egy vagy több könyvtárat vagy JAR fájlt a megjelenő fájlválasztó ablakban. A *Class path* bizonyos elemeinek törléséhez válassza ki azokat a listában, majd nyomja meg a *Remove* gombot. Mivel bizonyos modellek esetén számít a *Class path* lista elemeinek sorrendje, a mozgató gombokkal (*Move up* és *Move down*) a megfelelő sorrend kialakítható (a további részletekért lásd az 1.2.1 Bővíthető *Class path* szakaszt!).

A *Description*: mezőbe (opcionálisan) a futtatni kívánt modell rövid leírását lehet bevinni. Ez a leírás a későbbiekben a futtatás adataival bekerül az adatbázisba.

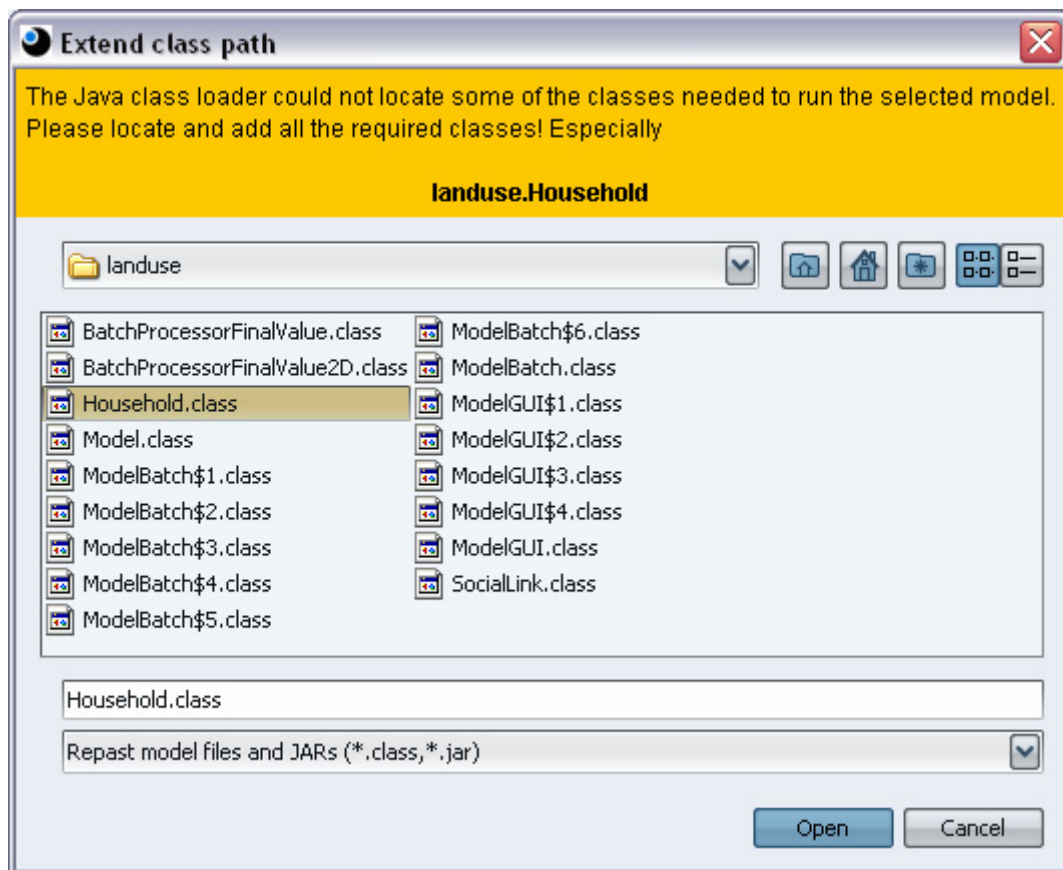
A *Parameter file*: sorban egy létező és alkalmas paraméterfájl elérési útja kerülhet opcionálisan. A Parameter Sweep Tool ennek tartalmával inicializálja a futtatásokat, azonban ez még módosítható a következő lapon. Amennyiben nincs paraméter fájl definiálva korábbról, a program generál egy újat.

### 1.2.1 Bővíthető *Class path*

Amikor megnyomja a *Next* gombot, a program megkísérli betölteni a kiválasztott modellt. Pontos *Class path* esetén a program tovább lép a paramétertér definiálására, ha hiányzik osztály, egy speciális fájlválasztó ablak (lásd 5. ábra!) segítségével megkéri a felhasználót, hogy határozza meg a hiányzó osztályhoz tartozó class fájl helyét.

A párbeszédablak tetején megjelenik a hiányzó osztály teljes neve félkövér betűkkel. A tovább lépéshez ki kell választani az osztályhoz tartozó class fájlt, vagy a JAR fájlt, ami tartalmazza az adott az osztály(oka)t. A *Class path* inkrementálisan bővíthető.

<sup>2</sup> Ez alapértelmezésben: *C:/Program files/MASS/MEME/resources/*

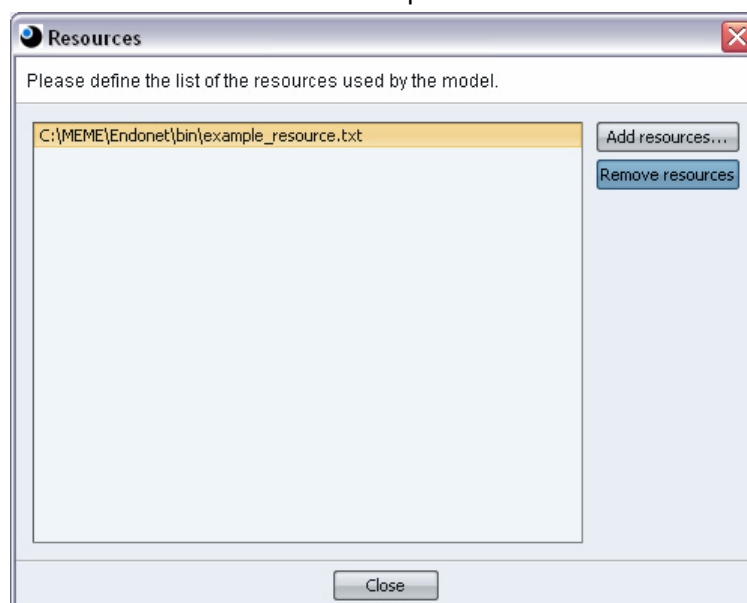


5. ábra – Class path bővítése

### 1.2.2 Erőforrásfájlok

Egyes modellek további szöveges, leíró és adat fájlokat használhatnak, amelyeket *erőforrásfájloknak* nevezünk.

Az ilyen jellegű modellek hiba nélküli elosztott futtatásához a programnak tudnia kell az erőforrásfájlok nevét és elérési útját. Ezek a *Resources...* gomb megnyomására megjelenő, az alábbi ábrán látható *Resources* párbeszédablakban adhatók meg.

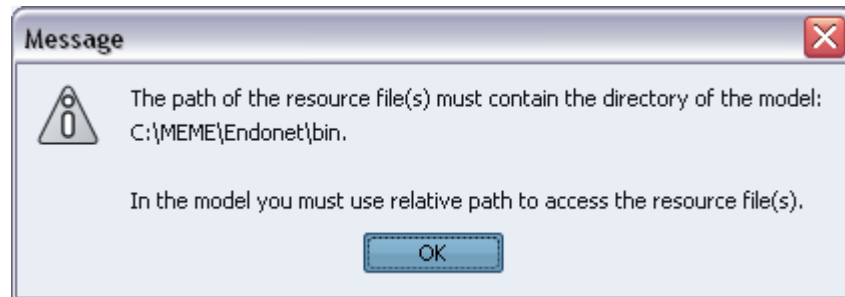


6. ábra – Erőforrásfájlok megadása

Az erőforrások listáját az *Add resources...* gomb megnyomásával és a megjelenő fájlválasztó ablakban egy vagy több fájl kijelölésével lehet bővíteni. A *Remove resources* pedig törli a kiválasztott fájlt az erőforrások listájából.

Technikai okok miatt minden erőforrásnak a modellkönyvtárban (vagy a modellkönyvtár valamelyik alkönyvtárában, tetszőleges mélységben) kell lennie. (A modellkönyvtár a kiválasztott modell csomagstruktúrájának kezdőpontja).

Ha más helyen lévő fájlt választ ki, az alább látható (7. ábra) figyelmeztetéshez hasonló üzenetet kap.



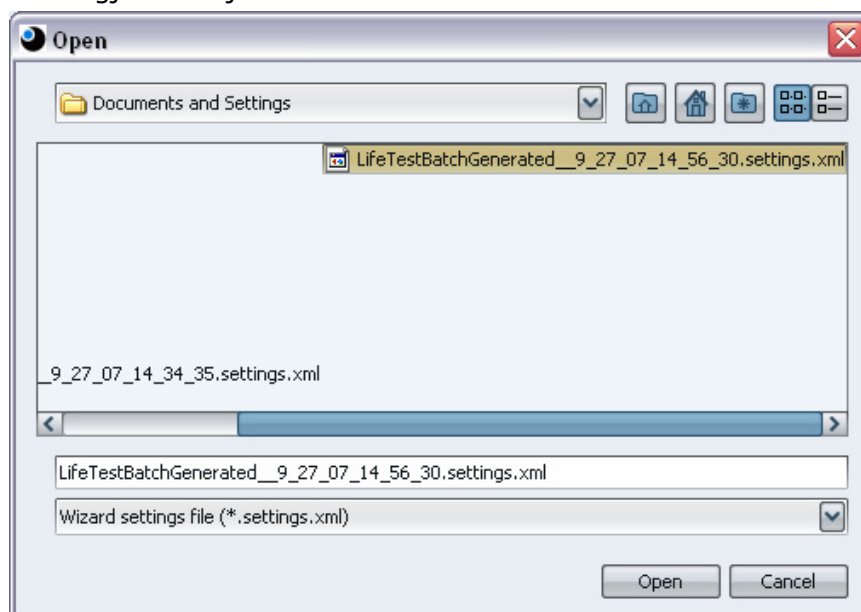
**7. ábra – Érvénytelen erőforrásfájl hibaüzenet**

**Tanács: Ahogy a fenti üzenet írja, a program segítségével csak olyan modellek futtathatók elosztott módban, amelyek relatív módon hivatkoznak az erőforrásaikra. Ha egy modell abszolút úttal definiál egy erőforrásfájlt, akkor nem fogja azt megtalálni, amikor a griden vagy a klaszteren fut.**

Ha modelljét helyben futtatja, akkor nincs szükség az erőforrások megadására (vagy arra, hogy a modell relatív módon hivatkozzon az erőforrásaira). Ezért ebben az esetben a *Resources...* gomb le is van tiltva.

### 1.2.3 Beállítások betöltése

A modell betöltése, a paramétertér definiálása és a rögzített adatok megadása után program legenerálja a batch-módon történő futtatáshoz szükséges modellt, illetve automatikusan elmenti az összes, modellre vonatkozó beállítást (pl. paraméterfájl, rekorderek, scriptek, stb.). Ezeket a beállításokat a *Load wizard settings...* gomb (*Model selection* oldal) megnyomásával és a megfelelő leírófájl kiválasztásával lehet visszatölteni. A megjelenő fájlválasztó ablak alatt látható.



**8. ábra – Beállítások betöltése**

Ez a fájlválasztó ablak annak a könyvtárnak a tartalmával nyílik meg, amelyet a *Preferences* párbeszédablakban a *Wizard settings folder* mezőben megadott. A

modellbeállításokat tartalmazó leírófájl neve mindig `.settings.xml`-re végződik. A név meglehetősen összetett, azonban rengeteg információt hordoz az adott modellbeállítás azonosításához.

*Például, a `LifeTestBatchGenerated__9_27_07_14_56_30` név alapján tudjuk, hogy*

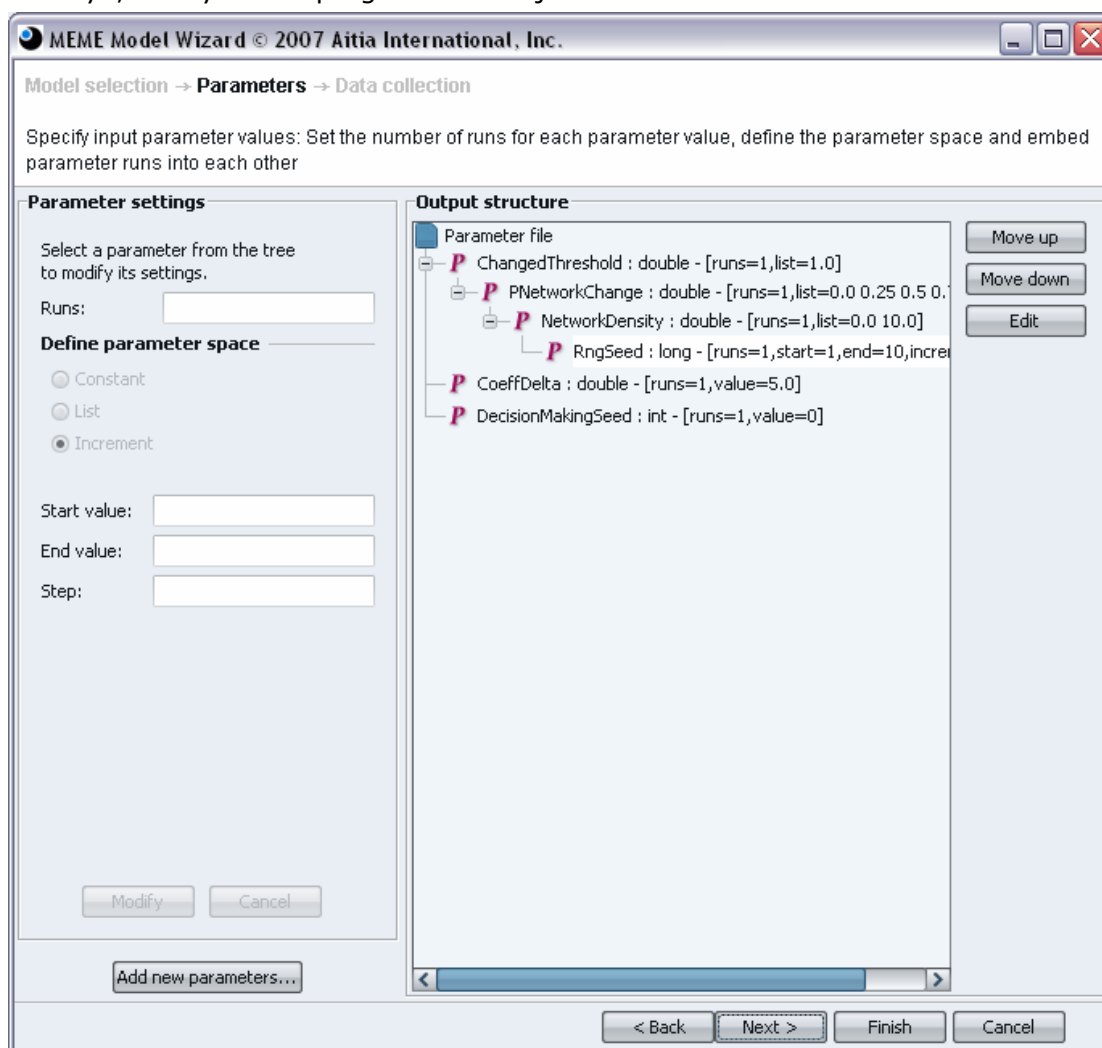
- Az eredeti modell neve `LifeTest`.
- A modellt a program generálta (mivel a név tartalmazza a „`BatchGenerated`” szót).
- A generálás dátuma 2007. szeptember 27.
- A generálás ideje 14:56:30.

Az `Open` gomb megnyomásra a program betölti a kiválasztott beállításokat, amelyekkel a modell módosítással, vagy módosítás nélkül újra futtatható.

**Megjegyzés: Amikor megnyomja a `Finish` gombot, hogy futtassa a (generált) modellt, az eszköz automatikusan elmenti a beállításokat akkor is, ha semmi változás nem történt az előző futtatás óta.**

### 1.3 Paramétertér beállítása

Az eszköz `Parameters` oldalán (lásd 9. ábra!) vehető fel a modell paraméterek azon tartománya, amelyeken a program lefuttatja a szimulációkat.



**9. ábra – Parameters oldal**

Ez az oldal egyrészt lehetővé teszi, hogy meghatározza a vizsgálandó paraméter-kombinációkat (azaz, hogy definiáljon egy paraméterfájlt), másrészt arra is lehetőséget ad, hogy módosítson egy létező paraméterfájlt.

### 1.3.1 A paraméterfa

A paraméter-kombinációk egy fával vannak reprezentálva. Ha egy paraméter egy másik „alatt” van a fában, az azt jelenti, hogy a „szülő” paraméter minden egyes értékéhez a „gyerek” paraméter felveszi az összes lehetséges értékét. Például, vegyük a *NetworkDensity* és *RngSeed* paramétereket a fenti ábrán! Az ottani beállításokból a következő sorok generálódnak a paraméterfájlba:

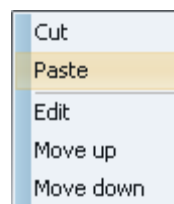
```
runs: 1
NetworkDensity {
  set_list: 0.0 10.0
    {
      runs: 1
      RngSeed {
        start: 1
        end: 10
        incr: 1
      }
    }
}
```

A Repast paraméterfájl szintaxisának megfelelően (a részleteket lásd a Repast dokumentációban [1]!) ez azt jelenti, hogy a *NetworkDensity* paraméter két értéket vesz fel, 0.0-át és 10.0-et, és mindkét értékhez az *RngSeed* paraméter sorban felveszi az 1,2, ..., 10 értékeket.

A paraméterek „Drag&Drop” technikával mozgathatóak más paraméterek „alá”. A sorrendet a mozgatni kívánt paraméter kiválasztásával és a mozgató gombok használatával lehet változtatni.

Egy adott paraméter beállításainak szerkesztéséhez jelölje ki azt a fában, majd nyomja meg az *Edit* gombot, vagy kattintson duplán a paraméter nevére a fában.

A szerkesztés a paraméterfa helyi menüjéből is elérhető, amely jobbkattintással aktiválódik. Ez a menü tartalmazza a *Cut* (Kivág) és *Paste* (Beilleszt) parancsokat is, ezek a Drag&Drop technika alternatívájaként szolgálnak paraméter-beágyazásnál.



**Megjegyzés: A Repast paraméterfájl szintaxisa nem engedi, hogy konstans értékű paramétereknek beágyazott paraméterei legyenek. A program a *Finish* gomb megnyomása után ellenőrzi ezt a feltételt.**

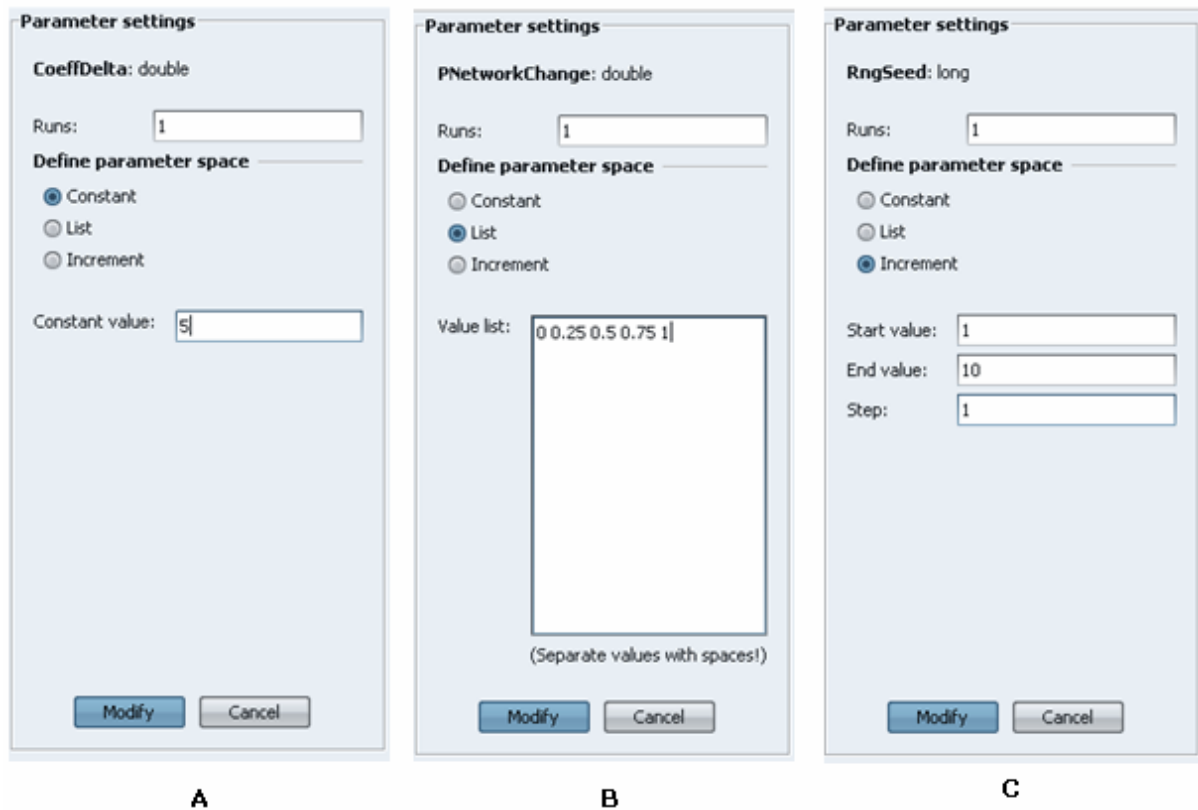
### 1.3.2 Paraméter beállítások módosítása

Az *Edit* gomb megnyomása után (vagy a dupla kattintásra a paraméterfában), definiálhatóak a kiválasztott paraméter beállításai a bal oldali panelen. A paraméterértékek háromféle módon adhatók meg (a Repast konvencióit követve):

- konstans érték megadásával (lásd 10. ábra, A eset!);
- értékek listájának megadásával (10. ábra, B eset!);
- iteráció megadásával, amit a kezdőérték, a végérték és a lépésköz megadása definiál (lásd 10. ábra, C eset!).

A *Runs* beállítás azt mutatja, hogy hány alkalommal vegye fel a paraméter ugyanazt értéket, mielőtt tovább lépne a következő értékre. Amennyiben több random seed-del futtat egy modellt, ezt érdemes változatlanul hagyni.

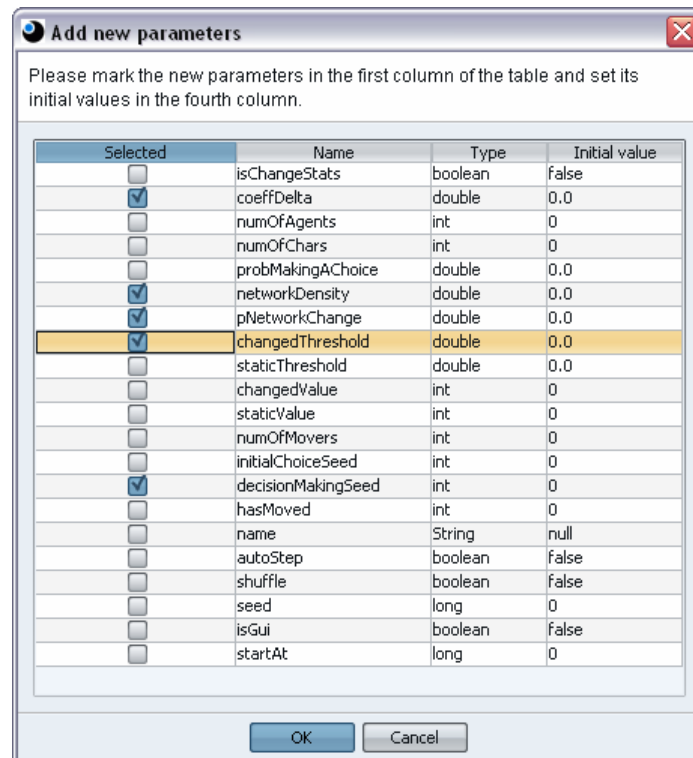
A *Modify* gomb megnyomására rögzülnek a kiválasztott paraméter új beállításai, míg a *Cancel* gomb elveti az aktuális módosításokat.



10. ábra – Paraméter beállítások módosítása

### 1.3.3 Új paraméterek hozzáadása

Az eredeti (a Repast modell *getInitParam()* metódusa által visszaadott) paraméterek köre bővíthető úgy, hogy egy nem-paraméter változót azzá nyilvánítunk. Ezt a lap baloldali panelének alján található *Add new parameters...* lenyomásával, a megjelenő ablakban lehet megtenni (lásd 11. ábra!).



11. ábra – Új paraméterek hozzáadása

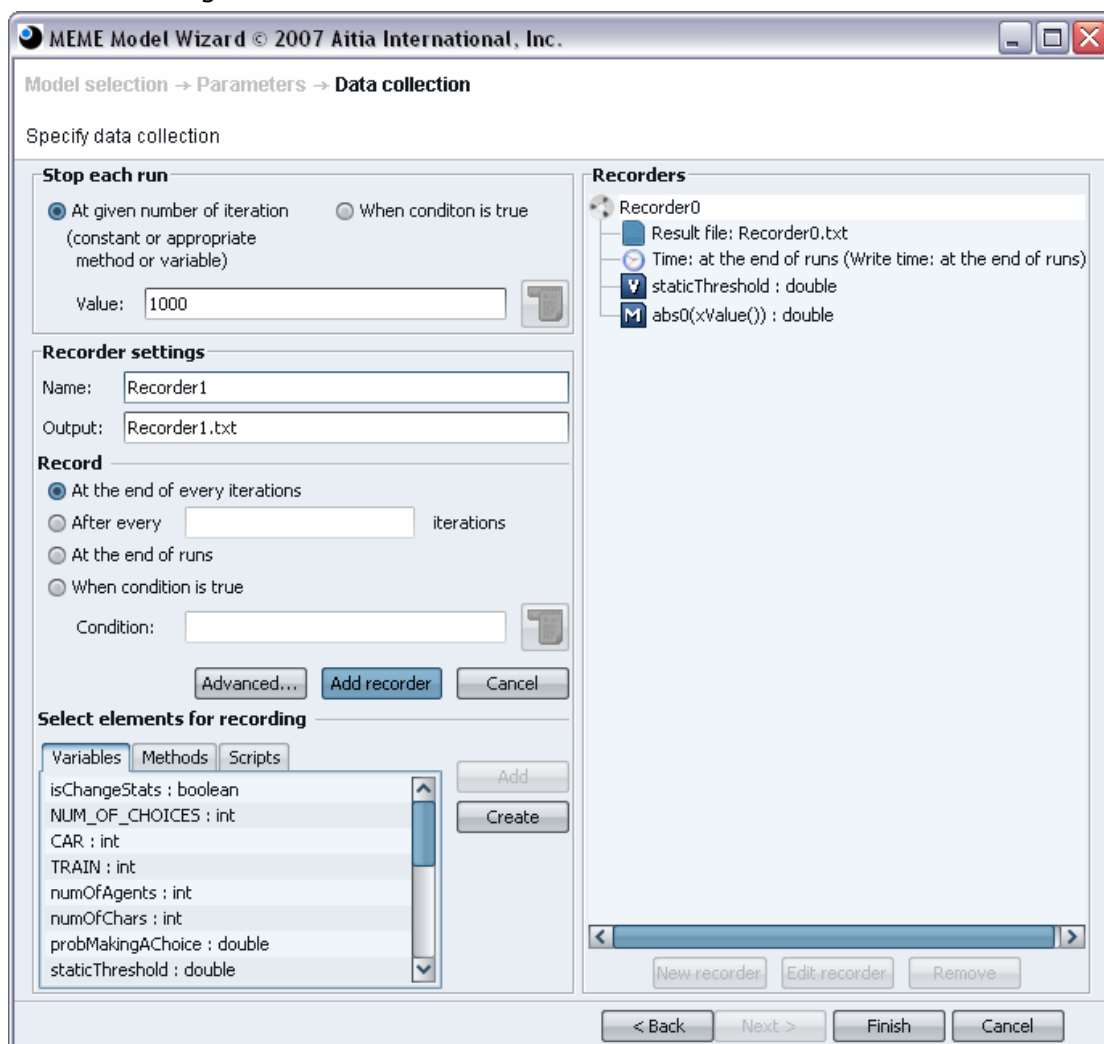
A táblázatban megjelenő változók paraméterré tehetőek a nevük előtt megjelenő négyzet bejelölésével. Az így keletkezett paraméterek kezdeti értéke is megadható; ehhez egyszerűen kattintson duplán a változó *Initial value* mezőjébe és vigye be a kívánt értéket. Az *OK* gomb megnyomásával a program paramétereket készít a kiválasztott változókból.

**Tanács: Ezt a lehetőséget még azelőtt használja, mielőtt bármilyen paraméterekkel vagy a paraméterfával kapcsolatos egyéb műveletet elvégezne, mivel alkalmazásakor minden korábbi, paraméterekre vonatkozó felhasználói beállítás elveszik.**

Ezen lehetőség használata esetén az eszköz módosítani fogja a Repast modellt: elkészíti a megfelelő lekérdező/beállító műveleteket az új paraméter változóhoz, és hozzáadja azt a paraméterek tömbjéhez.

## 1.4 Adatgyűjtési beállítások

A batch futtatást összeállító utolsó lap, a *Data collection* oldal az alábbi ábrán látható. Ezen az oldalon határozható meg, hogy a futtatások során mely adatok, milyen gyakorisággal, illetve hol kerüljenek rögzítésre. A szimuláció futásának leállási feltételét szintén itt kell megadni.



12. ábra – Data collection oldal

Az *Add recorder* gombbal lehet rekordert, adatrögzítő objektumot, illetve annak beállításait hozzáadni a modellhez, amely a kiválasztott, illetve létrehozott változók értékeit a futások során összegyűjti.

A *Cancel* gombbal az éppen szerkesztett rekorder változtatásait lehet elvetni, míg a rekorder beállításainak módosításai a *Modify recorder* gombbal rögzíthetőek.

Az *Advanced...* gombbal a rögzített adatok fájlba írásának szabálya adható meg (bővebben lásd 1.4.2.1 Eredmények kiírása szakaszt!).

Miután a kívánt beállítások rögzítésre kerültek, a *Finish* gomb megnyomására a program legenerálja a batch-módú futtatásra alkalmas modellt, elmenti a beállításokat és megkezdi a futtatást.

Az alábbiakban a fenti ábrán látható oldal beállításai kerülnek részletes kifejtésre.

### 1.4.1 Leállási feltétel

A leállási feltétel azt határozza meg, hogy egy adott futás hány lépés, vagy milyen feltétel teljesülésekor álljon le. Előbbi értelemszerűen egy szám, azonban ez nem feltétlenül egy konstans, lehet egy alkalmas típusú („egész”) változó neve, vagy egy olyan metódus neve, amelynek a visszatérési értéke alkalmas típusú. Utóbbi egy logikai feltétel, amelyet a modell változóiból és metódusaiból áll össze.

**Tanács: A program a *Finish* gomb megnyomása után ellenőrzi a leállási feltételt, de ez csak abban az esetben történik meg, ha szükséges új modellverzió generálása, azaz ha van definiált rekorder és/vagy új paraméterek.**

### 1.4.2 Adatrögzítők (rekorderek)

A jobboldali panelen található fa tartalmazza a létrehozott rekorder(ek)e)t. A fában megjelenik minden lényegi információ az adatrögzítő(k)ról:

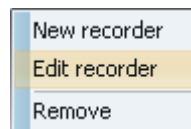
- Név;
- Kimeneti fájl neve (rekorderenként egy);
- Adatrögzítés ideje/gyakorisága (zárójelben a fájlba írás ideje);
- Rögzített változók és metódusok.

A rekorder összegyűjti a megadott változók és metódusok értékeit a futások alatt, amelyek végül a megadott eredményfájlba kerülnek. Hogy a szimulációk futását felgyorsítsuk, először a memóriában kerülnek eltárolásra az adatok, és csak megadott időközönként íródnak ki fájlba.

A MEME Parameter Sweep Tool segítségével a Repast modell létező változói és metódusai mellett származtatott értékek is lehetnek rögzítendő adatok. (Bővebben lásd 1.5 Scriptelés szakaszt!)

A létező rekorderek módosításához ki kell választani az adott rekorder tetszőleges elemét (pl. a nevét, valamelyik rögzítendő elemét, stb.) a fában és meg kell nyomni az *Edit recorder* gombot, vagy duplán kell kattintani a rekorder tetszőleges elemére.

Adatrögzítő eltávolításához jelölje ki a rekorder nevét, az eredményfájl nevét, vagy a rögzítés idejét/gyakoriságát a fában és nyomja meg a *Remove* gombot. Az egyes rögzítendő változók vagy metódusok a rekorderből való eltávolításához jelölje ki a törölni kívánt elemet és nyomja meg a *Remove* gombot.



A fenti műveleteket a fa helyi menüjéből (jobb kattintás) szintén el lehet érni.

Az adatrögzítők beállításait a *Recorder settings* felületen, a baloldalon lehet megadni. A program automatikusan felajánl egy alapértelmezett nevet a rekordernek és az eredményfájlnak, amelyek természetesen megváltoztathatóak.

Az adatrögzítés idejének/gyakoriságának megadásakor – *Record* felület – négy lehetőség közül lehet választani:

- *At the end of every iterations*: Minden iteráció után (azaz minden lépésben).
- *After every x iterations*: Rögzítés x iteráció után (például minden 10. lépésben).
- *At the end of runs*: Minden futás végén (azaz egy adatsor minden paraméter-kombinációra).

- *When condition is true*: Feltétel teljesülésekor. Ez a modell változóiból és metódusaiból készített logikai kifejezés lehet. Az eszköz ellenőrzi a kifejezés érvényességét a rekorder létrehozásánál/módosításánál.

#### 1.4.2.1 Eredmények kiírása

Az eredmények a memóriából fájlba történő kiírásának idejét az *Advanced...* gomb lenyomására megnyíló *Advanced recorder settings* párbeszédablakban lehet megadni. Ennek főként elosztott módban van jelentősége, ahol a túlságosan gyakori fájlba írás és az ebből eredő host-szerver kommunikáció nagyban lassíthatja a szimulációk futását. Három lehetőség közül lehet választani:

- *After every recording*: Minden adatrögzítés után;
- *At the end of the runs*: Minden futás végén (ez az alapértelmezett);
- *After x iterations*: Kiírás *x* iteráció után (például minden 20. lépésben).

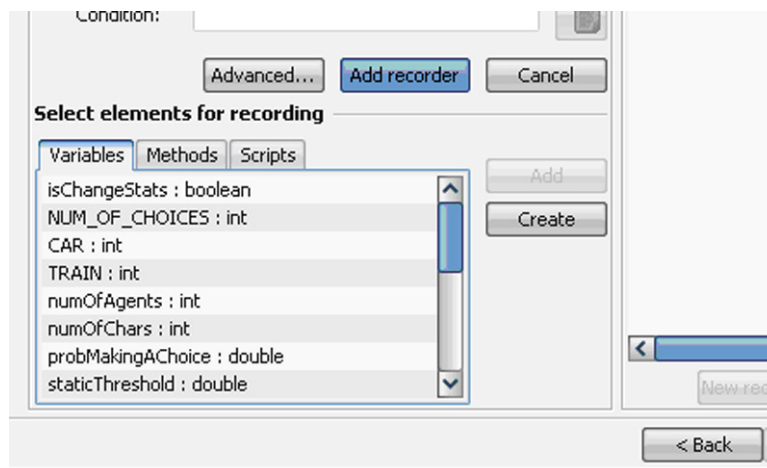


**Megjegyzés: Ezen opciók némelyike le lehet tiltva a rögzítés ideje/gyakorisága beállításával való összhangban. Például az *After x iterations* fájlba írási lehetőség értelemszerűen nem választható, ha adatrögzítésre az *At the end of the runs* opció van kijelölve.**

**Megjegyzés: A fájlba írás ideje automatikusan változhat, ha az adatrögzítés idejét/gyakoriságát a fájlba írás idejének definiálása UTÁN változtatja. Az általános szabály a következő: Nincs írás rögzítés előtt.**

#### 1.4.2.2 Rögzítendő elemek hozzáadása

Az adatrögzítők elkészítésének egyik legfontosabb lépése a rögzítendő változók és metódusok hozzáadása.



13. ábra - Rögzítendő elemek listái

A rekorderehez úgy lehet rögzítendő elemeket hozzáadni, hogy kijelöljük a fában a megfelelő rekordert (annak tetszőleges elemét), majd kijelöljük a baloldali panel alján található listában a kívánt elemeket, és megnyomjuk az *Add* gombot. A *Variables* fülön a modell változói, a *Methods* fülön a metódusai, míg a *Scripts* fülön a származtatott értékek szerepelnek. Ezek létrehozásáról lásd a következő alfejezetet!

## 1.5 Scriptelés

Ahogy korábban már említettük, a program lehetővé teszi a felhasználó számára, hogy származtatott értékeket (pl. statisztikákat) hozzon létre létező változókból és metódusokból egy könnyen kezelhető grafikus felületen és/vagy egyszerű scripteléssel.

Scriptek létrehozásához nyomja meg a *Create* gombot a *Data Collection* lapon (az *Add* gomb alatt található). Ekkor a *Create script* párbeszédablak jelenik meg (lásd 14. ábra!). Kétféle scriptet különböztetünk meg; a statisztikákat és a haladó scripelést, ahol forráskódot lehet írni.

### 1.5.1 Statisztikák

A statisztikák előre elkészített függvények, amelyek valamilyen számszerű információt adnak meg a szimulációról a megadott paraméterek alapján. A felhasználónak nincs más dolga, mint az adott statisztika aktuális paramétereit kiválasztani; ekkor a program elkészíti az adott statisztika egy olyan változatát, egy ún. statisztikapéldányt, amely a megadott paraméterek alapján számol. Ez a statisztikapéldány aztán felhasználható rögzítendő elemként, vagy más statisztikapéldányok/scriptek definiálásában.

A *Create script* párbeszédablak *Statistics* (statisztikák) lapját a 14. ábra mutatja be.

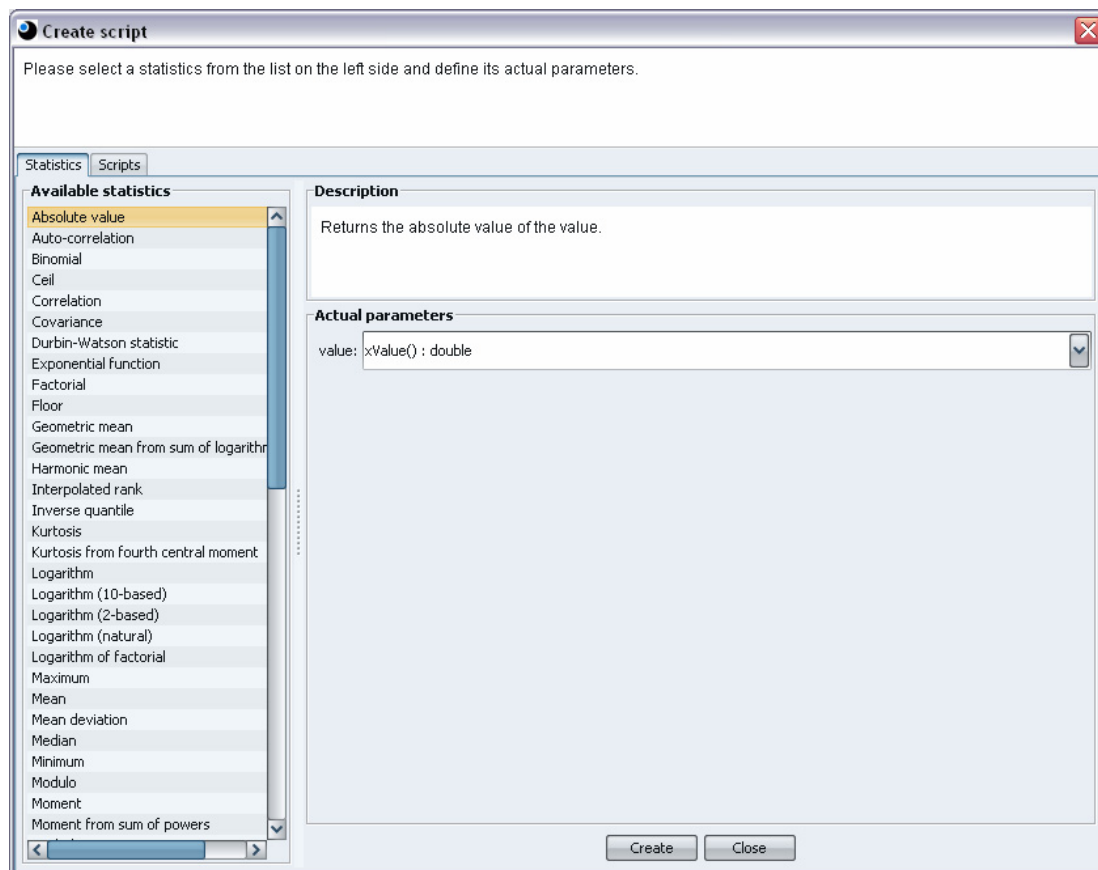
Az ablak bal oldalán található a rendelkezésre álló statisztikák listája (ábécé sorrendben). A jelenlegi verzióban több mint 60 darab van belőlük.

Amikor kiválaszt egyet közülük, a jobboldali panel tetején megjelenik az adott statisztika rövid leírása. A statisztikák nagyrészt a Colt könyvtáron alapulnak, ezért további információk a Colt dokumentációjában [3] található róluk.

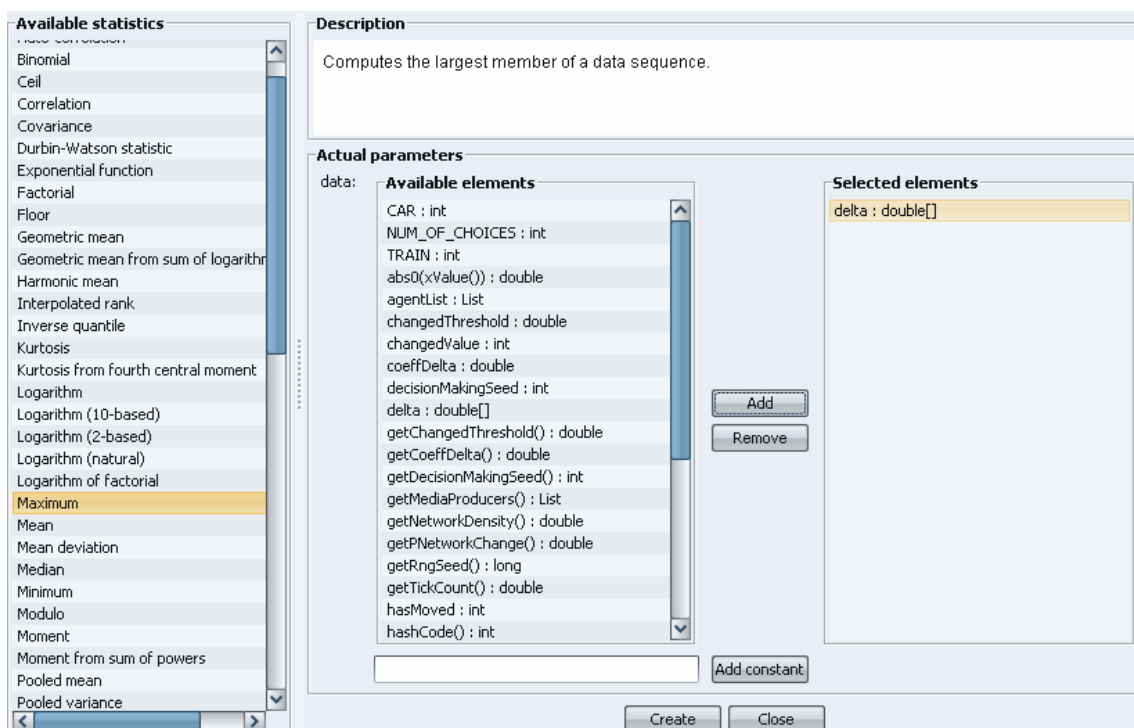
A leírás alatti területen lehet a kiválasztott statisztikához aktuális paramétereket rendelni. Általánosságban, kétféle paraméter kategória van:

- amikor egy számot kell megadni (például az Absolute value (abszolút érték) statisztika esetén lásd 14. ábra!), illetve
- amikor egy numerikus gyűjteményt kell definiálni (például a Maximum statisztika esetén, lásd 15. ábra!).

Az első esetben az aktuális paramétert egy szerkeszthető legördülő listából lehet kiválasztani. Ez a lista tartalmazza a Repast modell összes számváltozóját és megfelelő metódusát (azokat, amelyeknek a visszatérési értéke egy szám). A korábban generált metódusok (statisztikapéldányok és scriptek) közül is megjelennek a megfelelő visszatérési értékűek. Konstans paramétert is meg lehet adni a legördülő lista szerkesztésével.



14. ábra – Create script ablak, Statistics lap



15. ábra – Statisztika numerikus gyűjtemény kategóriájú paraméterrel

Gyűjtemény kategóriájú paraméterek esetében egy numerikus kollekciónak kell összeállítani a jobboldali listában (*Selected elements*). Ezt úgy lehet elérni, hogy a baloldali listában (*Available elements*) kijelöljük a kívánt elemeket, majd megnyomjuk az

**Add** gombot. A jobboldali lista tetszőleges eleme törölhető a megfelelő elemek kijelölésével és a *Remove* gomb használatával. A baloldali lista tartalmazza az összes szám és numerikus kollekció (például, `int []`, `double []`, stb.) változóját és megfelelő metódusát (amelyek visszatérési értéke szám vagy numerikus kollekció) a Repast modellnek (létező és korábban generált metódusok egyaránt szerepelhetnek). Ez a lista tovább bővíthető konstans értékek hozzáadásával. Ehhez be kell írni a kívánt értéket a lista alatti szövegmezőbe, majd meg kell nyomni az *Add constant* gombot.

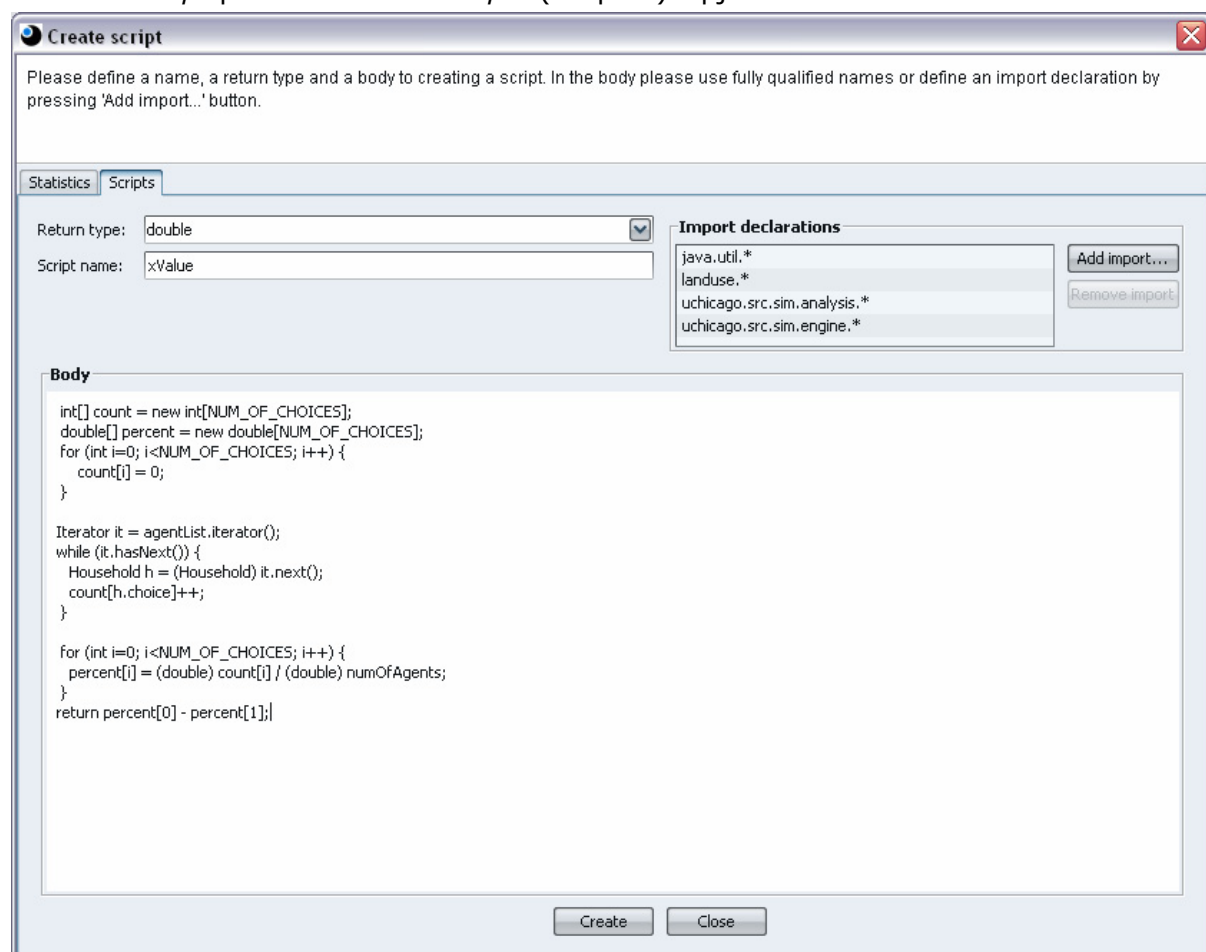
Statisztikapéldány készítéséhez nyomja meg a *Create* gombot. Minden hibaüzenet (például, ha a *Selected elements* lista üres) az információs panelen jelenik meg a párbeszédablak tetején.

## 1.5.2 Haladó scriptelés

Ahogy a neve is mutatja, ez a lehetőség csak gyakorlott felhasználók számára ajánlott, mivel némi programozási tudásra van szükség hozzá Java-ban.

Haladó scriptelés használatával a felhasználó új metódusokat (sctrieket) adhat hozzá a modellhez, amelyeket aztán felhasználhat rögzítendő elemként, vagy más sctriek/statisztikapéldányok készítésénél.

A *Create script* párbeszédablak *Scripts* (scriptek) lapját a 16. ábra szemlélteti.



**16. ábra – Create script ablak, Scripts lap**

Egy script elkészítéséhez a következőket kell definiálni:

- *Script name*: A script neve, egyedinek kell lennie (olyannak, ami még nem létezik a Repast modellben). A program automatikusan biztosít egy nevet, de ez megváltoztatható. Ha a megadott név már foglalt, a program hibát fog jelezni, amikor ellenőrzi a scriptet.
- *Return type*: A script visszatérési értékének típusa. Kiválasztható a megfelelő a legördülő listából, vagy be is gépelhető egy új típus.

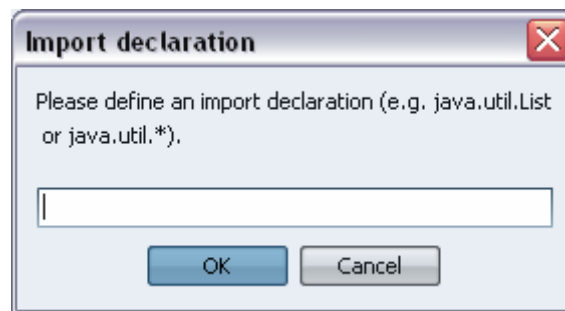
**Megjegyzés: Ha a visszatérési érték típusa nem szám, logikai vagy String, akkor az elkészült script nem használható fel, mint rögzítendő eleme egy rekordernek. De más scriptekben természetesen fel lehet használni.**

- *Body*: Törzs, a script forráskódja (lásd a Korlátok szakaszt a későbbiekben!)

A törzs és a visszatérési érték típusának megadásakor a Java osztályok teljes nevét kell használni, kivéve ha az alábbi csomagok valamelyikének részei:

- java.lang
- java.util
- uchicago.src.sim.analysis
- uchicago.src.sim.engine

A teljes nevek helyett lehetőség van import deklarációk megadására. Ehhez meg kell nyomni az *Add import...* gombot, és a megjelenő *Import declaration* párbeszédablakban (lásd 17. ábra!) definiálni kell a megfelelő import deklarációt (hasonlóan, mint Java-ban, kivéve, hogy nem kell használni az `import` kulcsszót). Import deklarációk eltávolításához a listából jelöljük ki az eltávolítandó elemeket és nyomjuk meg a *Remove import* gombot. Az előre definiált import deklarációkat nem lehet törölni.



**17. ábra – Import deklarációk megadása**

A script törzsében nem lehet a Java nyelv összes eszközét felhasználni, van néhány korlátozás.

#### **Korlátok:**

- A J2SE 5.0 által bevezetett új szintaxis (beleértve a felsorolási típusokat és a generikusságot) nem támogatott.
- A tömb inicializátorok (kifejezések vesszővel elválasztott listája { és } zárójelk között) használata csak egydimenziós tömbök esetén engedélyezett.
- A belső vagy névtelen osztályok alkalmazása nem támogatott.
- A címkézett `continue` és `break` utasítások használata is tilos.

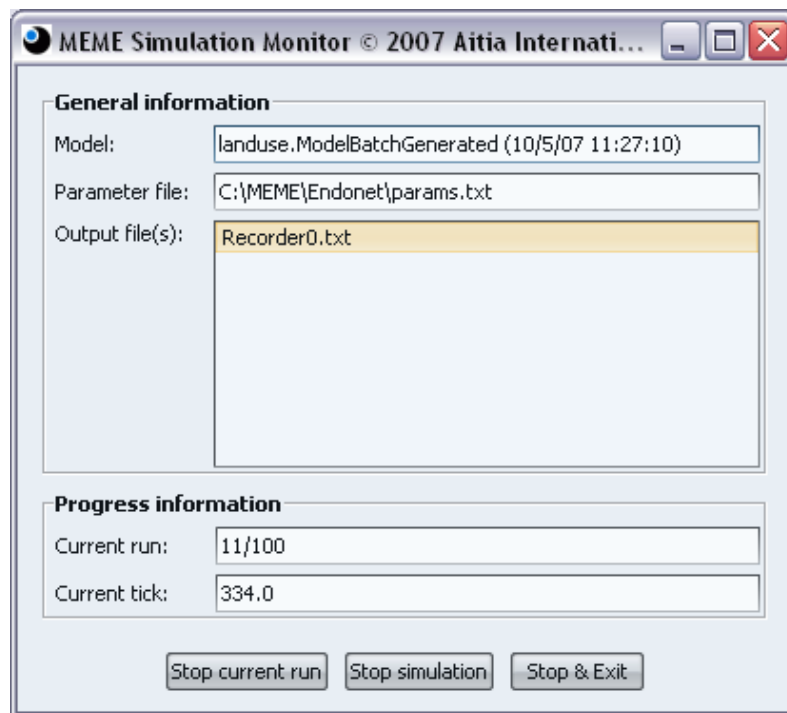
A definiált script létrehozásához nyomja meg a *Create* gombot. A program ellenőrzi a scriptet, minden hibaüzenet a párbeszédablak tetején lévő információs panelen jelenik meg.

**Tanács: A közeljövőben a fejlett scripteléshez sokkal több támogatást fogunk biztosítani. A *Create script* párbeszédablak felhasználói felülete sokkal inkább felhasználóbarát lesz.**

## **1.6 Beépített monitor**

A MEME Parameter Sweep Tool tartalmaz egy beépített monitort. Ez arra szolgál, hogy a felhasználó nyomon követhesse a helyi gépen éppen futtatott szimulációja haladását. A monitor automatikusan megjelenik, amint a szimuláció első futása elkezdődik. (Griden vagy klaszteren futtatott szimulációt a MEME Monitor alkalmazás használatával lehet megfigyelni. Bővebben lásd 2 A Monitor alkalmazás fejezetet!)

A beépített monitort a 18. ábra szemlélteti.



**18. ábra – Beépített monitor**

A monitor a következő információkat jeleníti meg a futó modellről:

- A modell teljes neve (a generálás/indítás dátumával és idejével),
- a használt paraméterfájl neve,
- az eredményfájlok nevei,
- a jelenlegi és az összes futás száma, és
- a jelenlegi lépésszám az aktuális futásban (a Repast *tick*-nek nevezi).

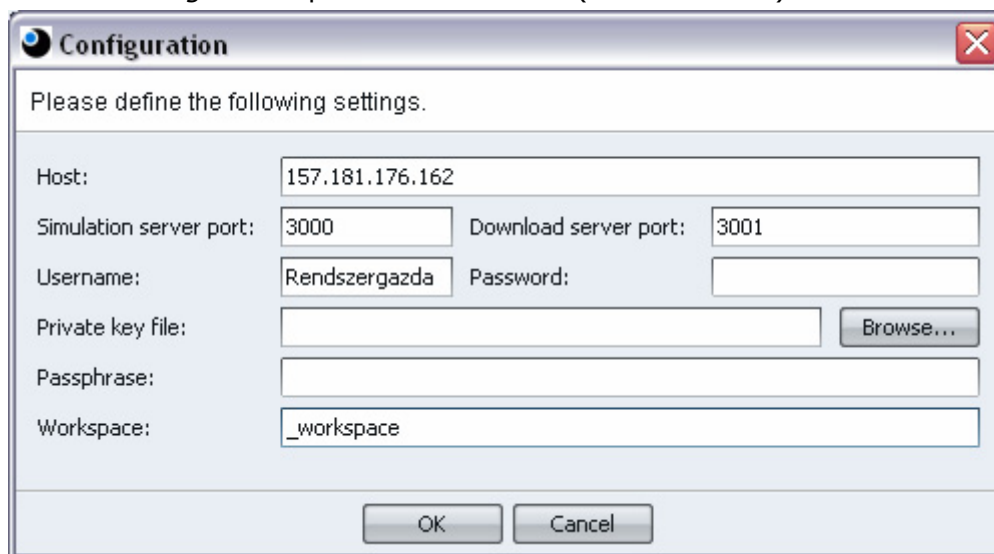
A *Stop current run* gomb félbeszakítja az aktuális futást és elkezdi a következőt (ha van). A *Stop simulation* gomb befejezi a szimulációt, de nem zárja be az alkalmazást, míg a *Stop&Exit* gomb a szimuláció leállítása után megteszi azt.

## 2 A Monitor alkalmazás

A MEME Monitor alkalmazás használatával a felhasználók nyomon követhetik egy adott griden vagy klaszteren éppen futó szimuláció aktuális állását (az aktuális futás számát), és letöltsék a befejezett szimulációk eredményeit.

### 2.1 Az alkalmazás konfigurálása

Jelenleg, ez a monitorozó eszköz egy külön álló alkalmazás, nem része a MEME Parameter Sweep Tool-nak. Ezért, a hálózati kapcsolatot a monitor számára szintén definiálni kell a *Configuration* párbeszédablakban (lásd 19. ábra!).



19. ábra – Monitor konfigurálása

A monitor első elindításakor ez a párbeszédablak automatikusan megjelenik. Később, ha a beállításokat módosítani akarjuk, a monitor főablakának *Configure...* gombjával lehet ezt az ablakot előhozni. Ahogy látható, az ablakban szereplő beállítások nagyon hasonlóak a MEME Parameter Sweep Tool *Preferences* párbeszédablakában lévőkhöz.

- *Hostname*: annak a gépnek a neve (vagy IP címe), ahol a szimulációs szerver és a letöltési szerver (ez egy másik szerveralkalmazás, amit a szimulációk eredményfájljainak letöltésekor használunk) található
- *Simulation server port*: a szimulációs szerver portja (alapértelmezésben a 3000-es port)
- *Download server port*: a letöltési szerver portja (alapértelmezésben a 3001-es port)
- *Username/Password*: A program a szimulációs eredmények letöltését SFTP-protokollon keresztül végzi. Ehhez alapesetben felhasználónév és jelszó páros megadására van szükség.
- *Private key*: Némely SFTP-szerverek nem támogatják a jelszó alapú azonosítást, helyette ún. privát kulcsot használnak, melyeket egy fájl tartalmaz.
- *Passphrase*: A private key-t használó SFTP-szerverek egy része egy további jelszót igényel, melyet *passphrase*-nek neveznek.
- *Workspace*: a szerverek munkakönyvtára (ami a távoli gépen található).

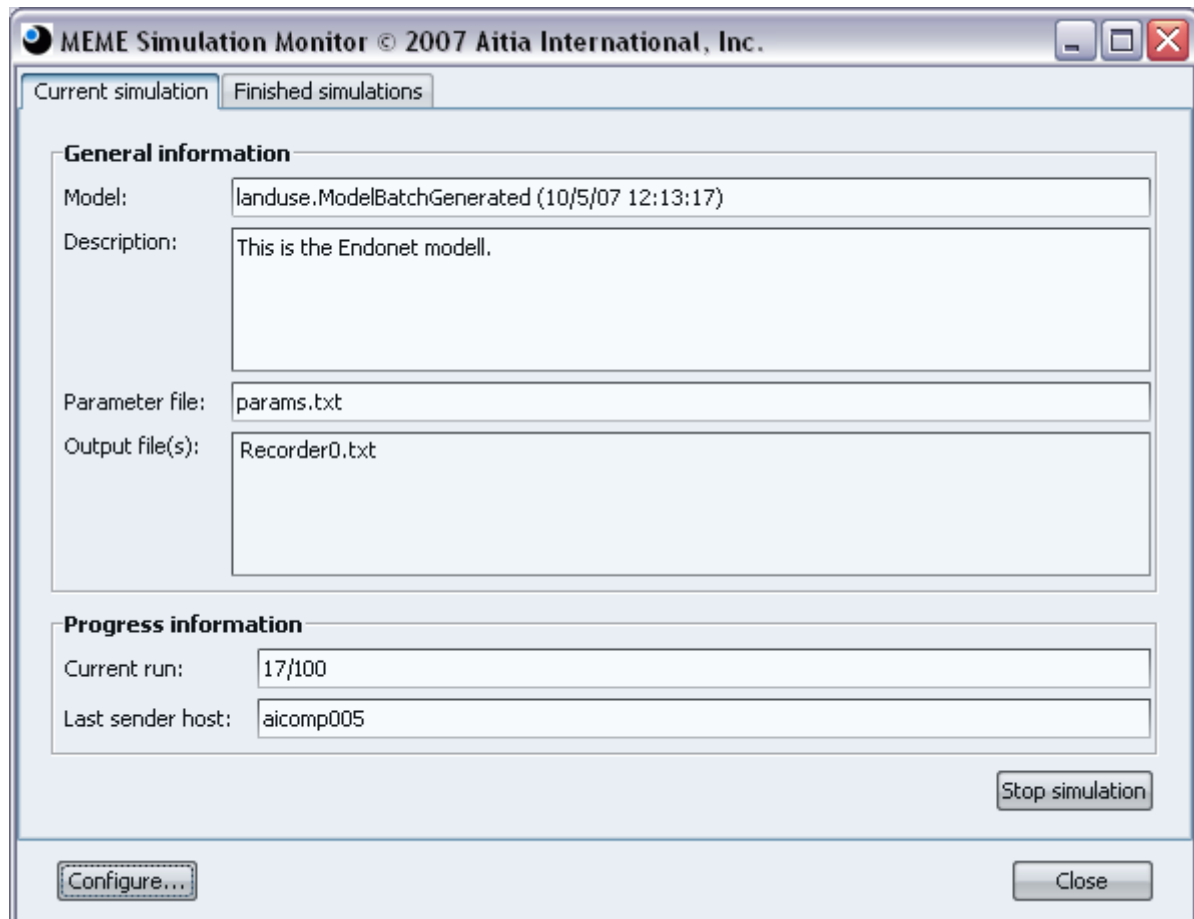
**Figyelmeztetés: A munkakönyvtárat az SFTP-szerver alapértelmezett felhasználói könyvtárához relatívan kell megadni.**

Ha a monitoralkalmazást a MEME Parameter Sweep Tool indította el automatikusan (lásd az 1.1.2 Elosztott futtatás beállítása szakaszban!), akkor nincs szükség a manuális

konfigurálásra, mivel a monitor örökli a varázsló hálózati beállításait. Az egyetlen kivétel a *Download server port* lehet, mivel a varázsló nem használja a letöltési szervert. A varázsló alapértelmezésként a *Simulation server port* + 1 értéket adja át, mint a letöltési szerver portját.

## 2.2 Az aktuálisan futó szimuláció adatai

Indítás után, a monitor összegyűjti és megjeleníti a szimulációs szerver által aktuálisan futtatott modell információit. (Lásd 20. ábra!)



**20. ábra – Az aktuálisan futó szimuláció adatai**

A monitor a következő információkat jeleníti meg a futó modellről:

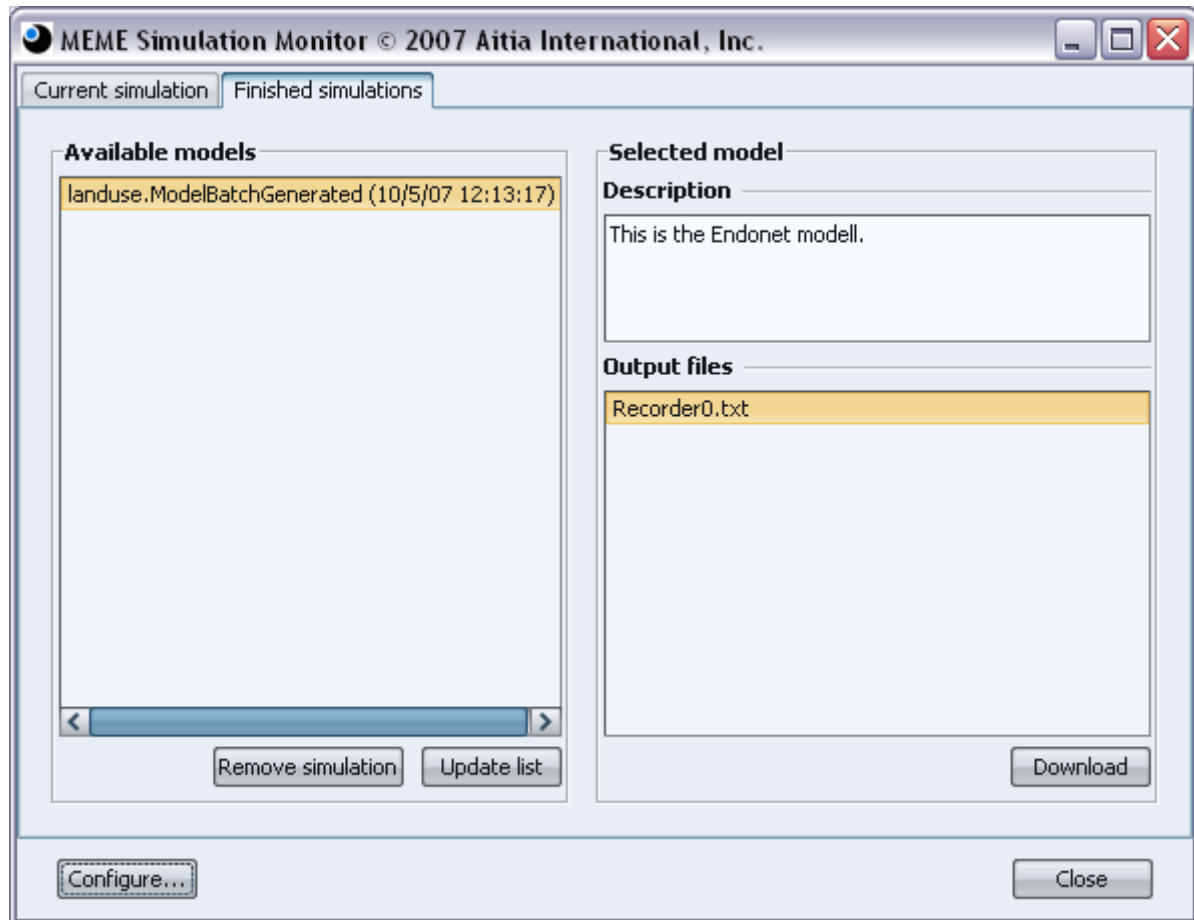
- A modell teljes neve (a feltöltés dátumával és idejével),
- a modell leírása,
- a használt paraméter-fájl neve,
- az eredményfájlok nevei,
- a jelenlegi és az összes futás száma, és
- annak a gépnek a neve, amelyiken a szimuláció aktuális futása megy.

**Tanács: Az első futás elindítása előtt a szimulációs szervernek inicializációs feladatokat kell elvégeznie, ami némi időbe kerülhet, ezért az első futás sokkal lassabbnak tűnhet, mint a más futások**

A *Stop simulation* gomb megnyomásával utasíthatja a szimulációs szervert az aktuális modell futásának megszakítására és a következő indítására. A szimulációs szerver megvárja, amíg a gridhez vagy klaszterhez tartozó gépeknek már kiosztott futások végetérnek, csak aztán állítja le a kísérletet.

## 2.3 Eredmények letöltése

A monitoralkalmazás másik lapja a befejezett szimulációkkal (*Finished simulations*) foglalkozik. Itt lehet letölteni az elvégzett kísérletek eredményeit (lásd 21. ábra!).



**21. ábra – Finished simulations lap**

A baloldali lista tartalmazza a befejezett szimulációkat (azok teljes neve és a feltöltés dátuma és ideje jelenik meg itt). Ez a lista automatikusan frissül, amikor egy kísérlet véget ér, de kézzel is lehet frissíteni az *Update list* gomb megnyomásával.

Ha kiválaszt egy elemet a listából, az adott szimulációhoz tartozó leírás megjelenik a jobboldali panel tetején, az alatt pedig a letölthető fájlok listája. A *Download* gomb megnyomásával lehet letölteni a *kiválasztott* kimeneti fájlokat.

A befejezett szimulációk törölhetők is a szerverről. Ehhez válasszuk ki a kívánt szimulációt a baloldali listából, majd nyomjuk meg a *Remove simulation* gombot. Óvatosan használja ezt a gombot, mivel ilyenkor az összes, a szimulációhoz tartozó kimeneti fájl véglegesen elveszik.

## 3 Gyakorta ismétlődő kérdések

**Kérdés:** A MEME Parameter Sweep Tool *Data collection* lapján definiálok egy leállási feltételt, de egyetlen adatrögzítőt sem adok a modellemhez. Fog a program generálni egy új modellt?

**Válasz:** Nem. Csak akkor generál modellt, ha adatrögzítőket és/vagy új paramétereket adunk a modellhez.

**Kérdés:** Kiválasztottam egy batch-verziójú modellt (kézzel írt leállási feltétellel), majd definiáltam hozzá rekordereket és egy másik leállási feltételt a MEME Parameter Sweep Tool-lal. Mikor fog a generált modell leállni?

**Válasz:** Ebben az esetben mindkét leállási feltétel aktív lesz. Amikor valamelyik igazgató válik (vagy a szimuláció eléri azt a lépést, amit valamelyikben megadtak), a szimuláció leáll.

**Kérdés:** Elindítottam két hosszú szimulációt a MEME Parameter Sweep Tool-lal. A szimulációk párhuzamosan fognak futni? Hogyan jeleníti meg a monitor az aktuális állásukat ebben az esetben?

**Válasz:** Amikor elindítunk egy szimulációt a programmal, az küld egy üzenetet a szerveralkalmazásnak – miután az összes szükséges fájlt felmásolta a távoli gépre –, amiben arra utasítja azt, hogy futtassa le a szimulációt. A szerver egy sorban tárolja a beérkező kéréseket, és egymásután teljesíti azokat. A legkorábban beérkezett kérés teljesül leghamarabb. A monitor pedig mindig az aktuálisan futó szimulációról szolgáltat információkat.

**Kérdés:** Szeretném a modelletem a helyi gépen futtatni és követni annak menetét, de a *Start monitor after wizard is closed* opció a *Preferences* párbeszédablak *Network* lapján csak elosztott mód esetén választható ki. Miért?

**Válasz:** Ha a modellt a helyi gépen futtatja, akkor a futás aktuális állapotát a program automatikusan mutatja a beépített monitoron. A *Start monitor after wizard is closed* opció a különálló monitoralkalmazásra vonatkozik, amelyet csak elosztott modú futtatás esetén használunk.

**Kérdés:** A MEME Parameter Sweep Tool-t használtam a modellem futtatásához és kaptam egy nagyméretű eredményfájlt? Hogyan tudom a Model Exploration Module (MEME) alkalmazást használni, hogy rendszerezem és megjelenítsem az eredményeimet?

**Válasz:** Jelenleg, a MEME Parameter Sweep Tool, a MEME Monitor és a MEME különálló alkalmazások. Ezért külön el kell indítania a MEME-t, és ott a *File / Import / Repast result file* menüpont segítségével importálnia kell az eredményfájl tartalmát a MEME adatbázisába. Ezután már felhasználhatja az MEME összes lehetőségét az adatok rendszerezésére és vizualizálására. (Részleteket lásd a MEME Felhasználói kézikönyvben [4]!). A közeljövőben integrálni fogjuk a Parameter Sweep Tool-t és a monitort a MEME-vel, azt követően a MEME automatikusan össze fogja gyűjteni az eredményeket, ami tovább fogja egyszerűsíteni a modellezői munkát.

## 4 Hibaelhárítás

A MEME Parameter Sweep Tool és a MEME Monitor folyamatos fejlesztés alatt áll, így bármilyen visszajelzést és hibabejelentést szívesen fogadunk. Ugyanakkor ez azt is jelenti, hogy új, javított verziók rendszeresen kerülnek kiadásra a következő oldalon:

<http://meme.aitia.ai/>

Ha bármilyen probléma lépett fel a programok használatával kapcsolatban, vagy ötlete, javaslata van, kérjük lépjen velünk kapcsolatba a [mass@aitia.ai](mailto:mass@aitia.ai) címen.

Amennyiben hibajelentést küld, kérjük csatolja az alkalmazások programkönyvtárában megtalálható *PSW.log* fájlt, valamint írja le azokat a lépéseket, amelyeket a hiba felmerülése előtt tett. Ha a hibát akkor észlelte, amikor elosztott módon futtatott egy modellt, kérjük csatolja az *outLog.txt* és *errLog.txt* fájlokat is, amelyeket a szimulációs szerver küldött Önnek.

## 5 Összefoglalás

A MEME Parameter Sweep Tool lehetővé teszi Repast felhasználók számára, hogy egy grafikus felületet használva paramétertér-keresést végezzenek, és tegyék ezt akár elosztott módon (egy griden vagy klaszteren).

A MEME Monitor alkalmazás használatával a felhasználók nyomon követhetik egy adott griden vagy klaszteren éppen futó szimuláció aktuális állását (az aktuális futás számát), és letöltsék a befejezett szimulációk eredményeit.

A közeljövőben számos módon továbbfejlesztjük az alkalmazásokat. A tervezett fejlesztések többek között a következők:

- A MEME Parameter Sweep Tool integrációja a Model Exploration Module (MEME) alkalmazással.
- A MEME Parameter Sweep Tool integrációja a FABLES IME alkalmazással.
- A statisztikakészítés szolgáltatás bővítése – a modell tagjainak adattagjai/metódusai szintén lehetnek aktuális paraméterek.
- A fejlett scriptelés lehetőségeinek bővítése – külső könyvtárak is felhasználhatók; lehetőség lesz saját adattagok definiálására és inicializációjára
- Az intelligens paramétertér-keresés támogatása – jelenleg a MEME Parameter Sweep Tool csak a legegyszerűbb paramétertér-keresési stratégiát támogatja, amit „hiperkockának” neveznek, és ahol minden paraméter-kombináció független egymástól. De léteznek más – sokkal hatékonyabb – stratégiák is, ahol a következő vizsgálandó paraméter-kombináció függ az előző eredményektől.

## 6 Referenciák

- [1] Repast - Recursive Porus Agent Simulation Toolkit  
<http://repast.sourceforge.net/>
- [2] ProActive  
<http://proactive.inria.fr/>
- [3] Colt  
<http://dsd.lbl.gov/~hoschek/colt/>
- [4] Model Exploration Module - Felhasználói kézikönyv