



Plant Simulation alaptanfolyam

Hallgató neve: _____

Hallgató cége: _____

Tanfolyam dátuma: _____



Tanfolyam Plant Simulation

1. modul: Alapok



1. fejezet

Mi a szimuláció?

A szimuláció egy rendszer leképezése dinamikus folyamataival együtt egy olyan modellben, amellyel kísérletezni lehet. Célja olyan eredmények szerzése, amelyek a valóságban felhasználhatók.

VDI 3633, Blatt 1, 1993

Új létesítmény tervezése:

- ▶ Az idők és teljesítmény meghatározása és optimalizálása.
- ▶ Méretezés meghatározása.
- ▶ Teljesítménykorlátok meghatározása.
- ▶ A hibák bekövetkezésének elemzése.
- ▶ Emberi erőforrásigények felmérése.
- ▶ A létesítmény működéséről információ gyűjtése.
- ▶ Lehetséges vezérlési stratégiák meghatározása
- ▶ Különböző alternatívák kipróbálása.

Meglévő létesítmény optimalizálása:

- ▶ Vezérlési stratégiák optimalizálása.
- ▶ Műveleti sorrend optimalizálása.
- ▶ Napi teljesítmény tesztelése.

- A létesítmény kihasználtsága
- A gépek kihasználtsága
- A teljesítmény

növekszik

- Gyártási idő
- Emberi erőforrás
- Tároló hely méret

csökken

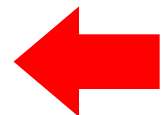
- Tárolóméretek
- Szállítóeszközök, AGV-k száma
- Munkadarab hordozók száma

meghatározható

- Tervezési alternatívák
 - Vezérlési stratégiák
 - Tervezési hibák
 - Beruházások
- kipróbálása
 - optimalizálása
 - elkerülése
 - megőrzése



A létesítmény profitjának növelése!



- Nem pontos leírások és paraméterek a szimulációs modellben.
- A cél hibás vagy hiányzó volta.
- Túl részletes és túl kiterjedt modellezés.
- Túl egyszerűsített modellezés, ami használhatatlan eredményeket ad.
- Nem eléggé ellenőrzött modellje a létesítménynek.

3 szakasz*

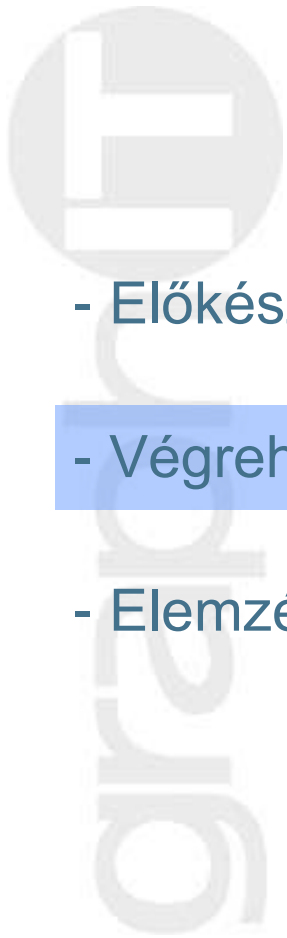
- Előkészítés

- Végrehajtás

- Elemzés

- Döntés: megéri szimulációt készíteni?
- A feladat és a cél definiálása
- Erőfeszítés megbecsülése
- Adatok gyűjtése, előkészítése, ellenőrzése
- Analitikus durva becslés
- Szimulációs modell készítése, ellenőrzése

* forrás: VDI 3633 és ASIM útmutató, 1997

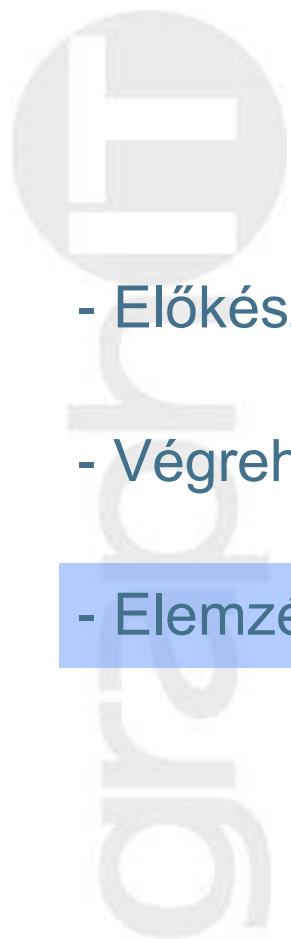


- Előkészítés

- Végrehajtás

- Elemzés

- Plant Simulation kísérletek
- Szimulációs kísérletek végrehajtása
- Szimulációs kísérletek ellenőrzése



- Előkészítés
- Végrehajtás
- Elemzés

- Eredmények formázása
- Eredmények értelmezése
- Eredmények dokumentálása

- Probléma és cél definiálása



- Rendszer analízis



- Adatgyűjtés



- Modell készítése



- Modell ellenőrzése



- Kísérletezés és modell analízis



- Eredmények értelmezése



- Javaslatok készítése



▪ Probléma és cél definiálása

- Probléma lebontása
- Érdemes a problémát szimulálni?
- Szimuláció céljainak definiálása

Például:

Gyár kapacitásának megbecslése

Tárolók, pufferek méretezése

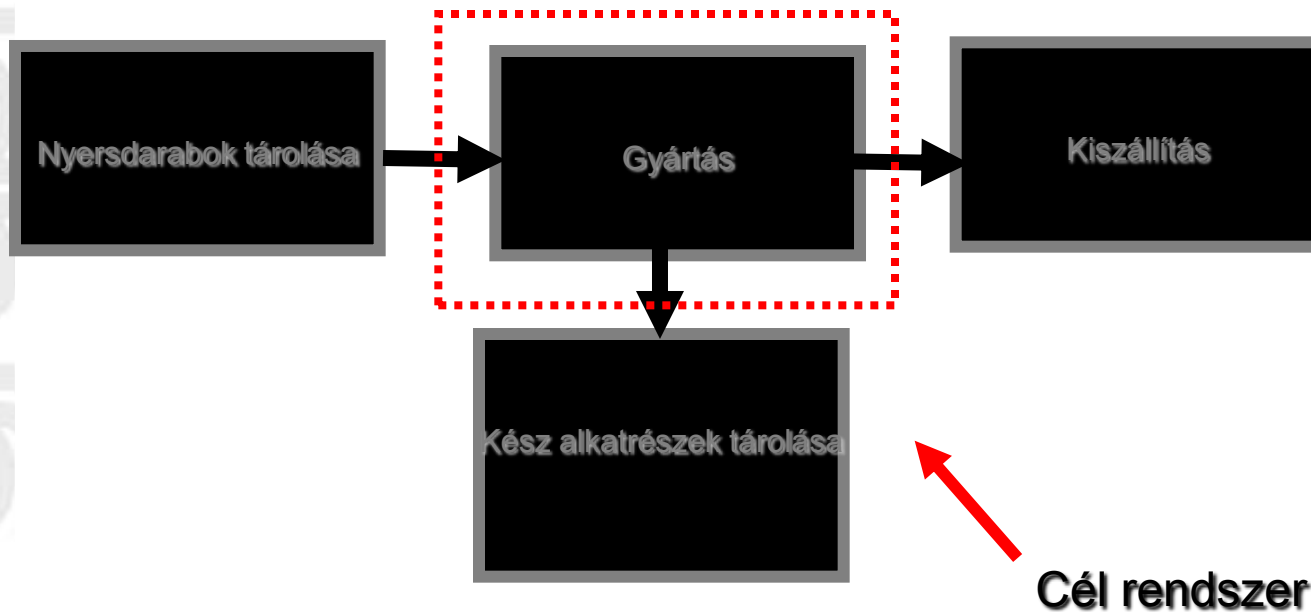
Megfelelő vezérlési stratégiák készítése

Optimális számú személyzet kialakítása

▪ Probléma és cél definiálása

▪ Rendszer analízis

- A szimulálni kívánt rendszer leírása
- A rendszer struktúrájának definiálása
- A szimuláció környezetének definiálása (a környezet határai)



▪ Probléma és cél definiálása

▪ Rendszer analízis

▪ Adatgyűjtés

- Lebontás különálló rendszerelemekre
- Rendszerelemek adatainak gyűjtése, jellemzők definiálása



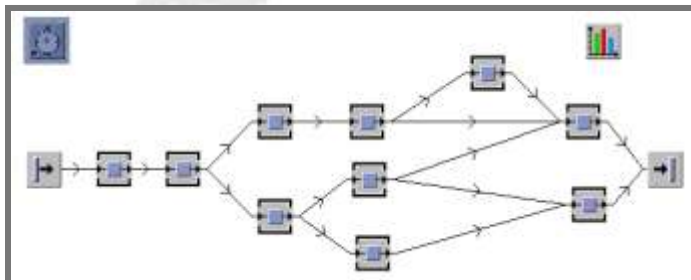
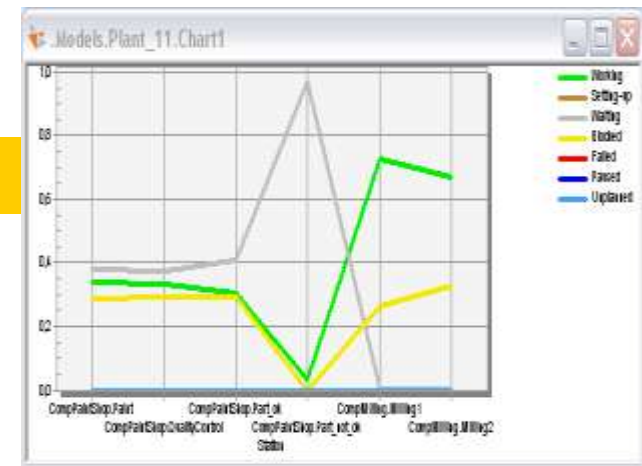
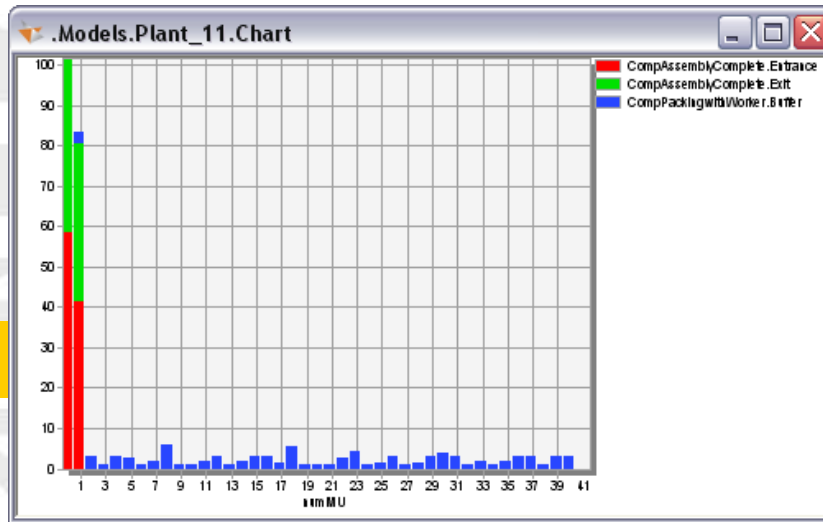


▪ Modell készítése

- A valós rendszer modellé alakítása a kísérletezéshez

▪ Modell ellenőrzése

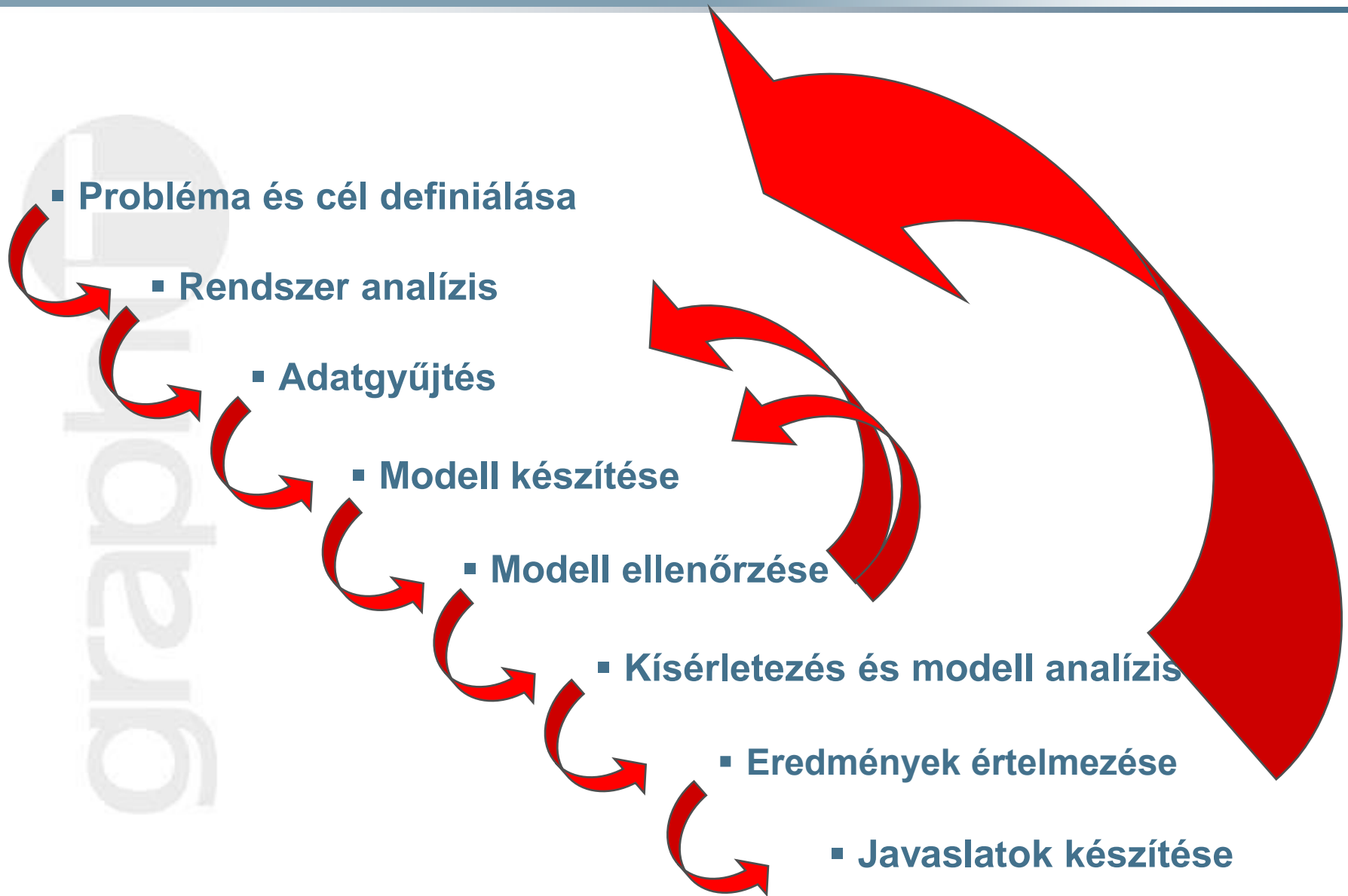
- Annak ellenőrzése, hogy a modell és a valóság megfelelően megegyezik

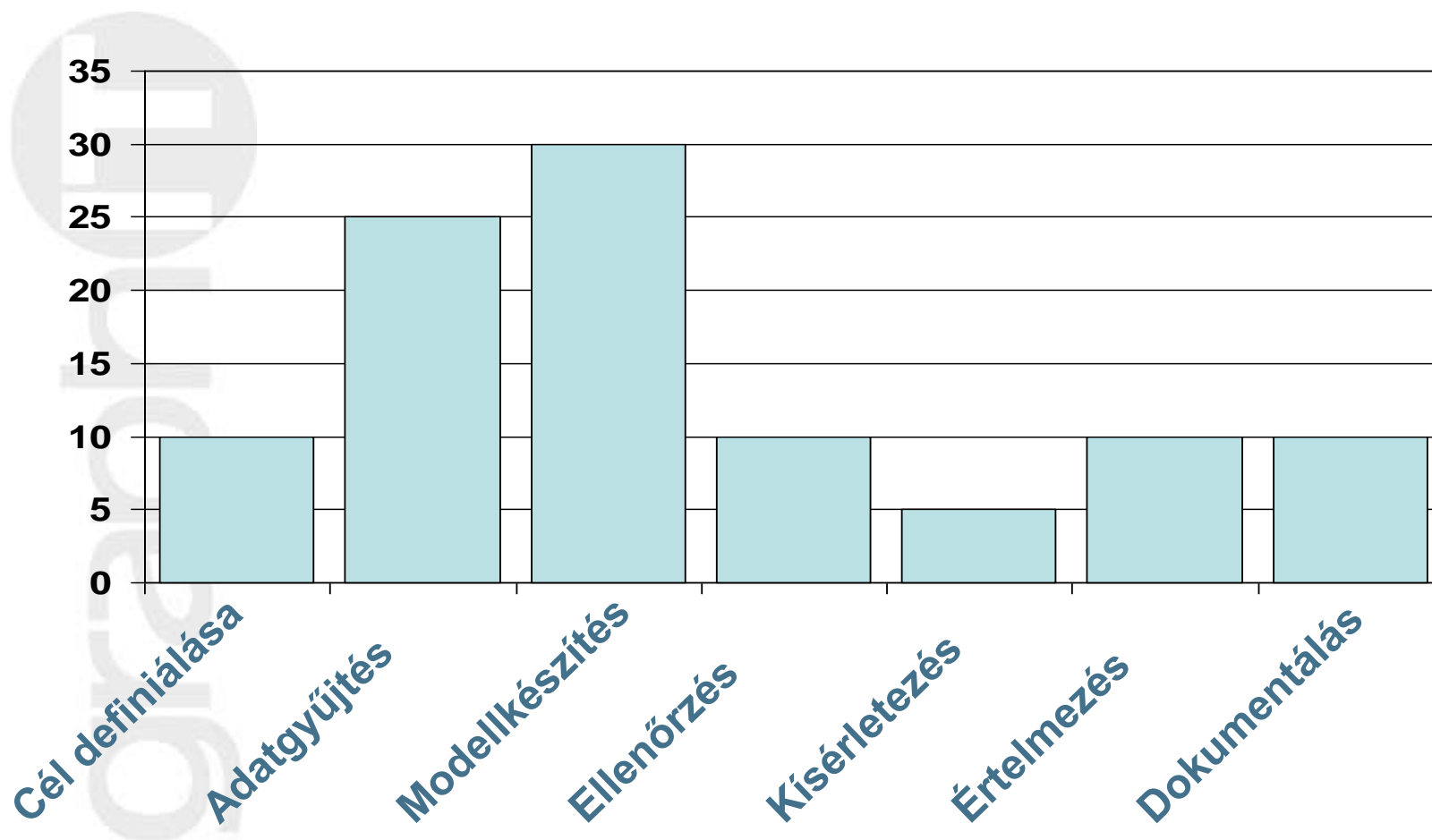


▪ Kísérletezés és modell analízis

▪ Eredmények értelmezése

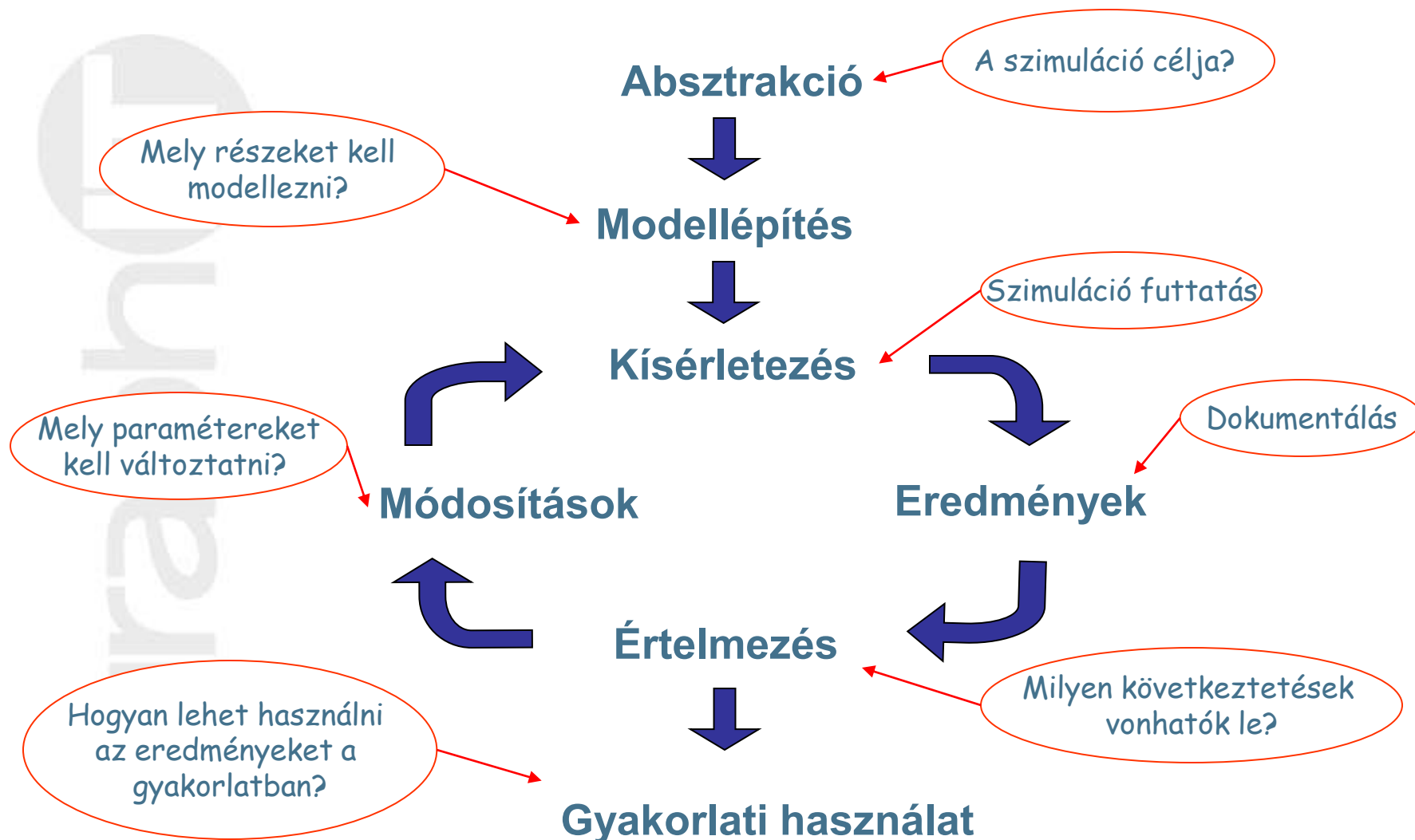
▪ Javaslatok készítése





Feladatok megoszlása: szimulációs projektekre* (egyszeri projektekre, forrás IWB 1997)

- Ne általánosítson szimulációs eredmények alapján
- Tervezze meg pontosan a kísérleteket
- Ellenőrizze a modellt
- Dokumentálja a bemeneti adatokat és kapcsolatokat pontosan
- Készítsen annyira részletes modellt, amennyire szükséges, ami annyira általánosított, amennyire lehetséges
- A szimulációt végezze valós adatokkal



A diszkrét esemény-orientált szimuláció bizonyos időpillanatokban mutatja a modell komponensek állapotváltozását, nem folytonosan az időben. Amikor bizonyos esemény megtörténik, a megfelelő modell komponensek állapota megváltozik. A Plant Simulation ezeket az eseményeket lépcsőről lépcsőre dolgozza fel. A fő előnye ennek a megközelítésnek, hogy a Plant Simulation nem foglalkozik az események közötti idővel.

Emellett a Plant Simulation objektum orientált alkalmazás, ami azt jelenti, hogy gyermek objektumok képesek tulajdonságokat és jellemzőket örökölni a szülő objektumuktól.

2. fejezet

Plant Simulation Licenctípusok

Licencípusok:

- Plant Simulation Professional
- Plant Simulation Standard
- Plant Simulation Application
- Plant Simulation Runtime

Oktatási csomagok:

- Plant Simulation Research*
- Plant Simulation Educational**

* Kutatási célra

** Oktatási célra

Opcionális termékek:

- Plant Simulation 3D
- Plant Simulation Interface Package
- Plant Simulation Gantt
- Plant Simulation Aris
- Plant Simulation SDX
- Process Designer Interface
- Plant Simulation Assembly
- Plant Simulation Shop

3. fejezet

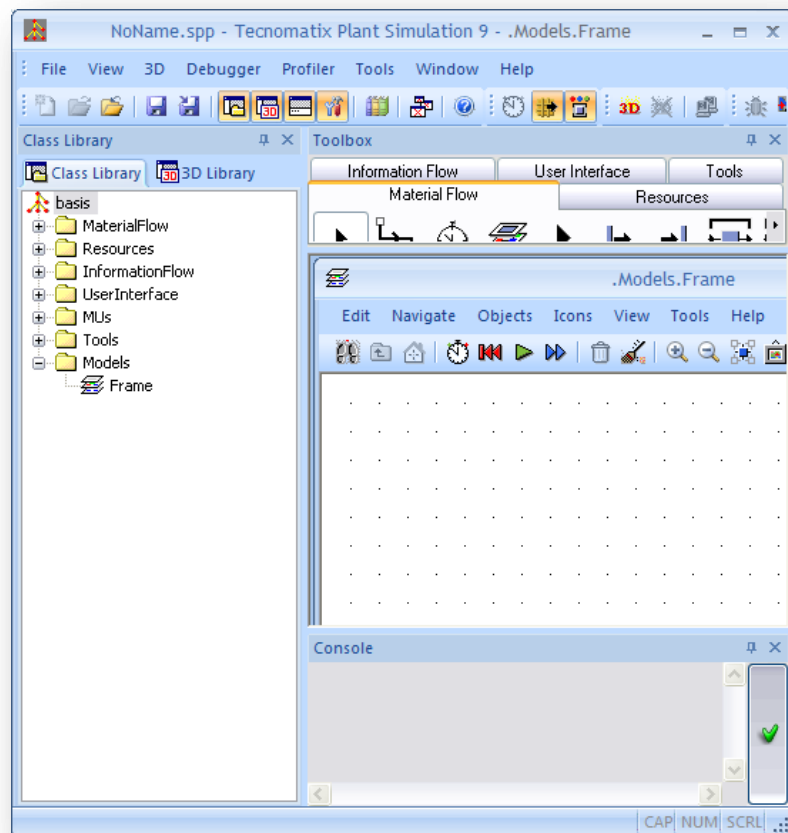
A Plant Simulation modellfájl

A Plant Simulation a Tecnomatix szabványos platformján a TUNE alrendszeren fut.

MS Windows 2000 és MS Windows XP operációs rendszerek alatt

A Plant Simulation indításakor a megfelelő felhasználói licenc ikonját kell elindítani, pl:

*Start > Programs > Tecnomatix > Plant Simulation x.y > Plant Simulation
(licenctípus neve)*



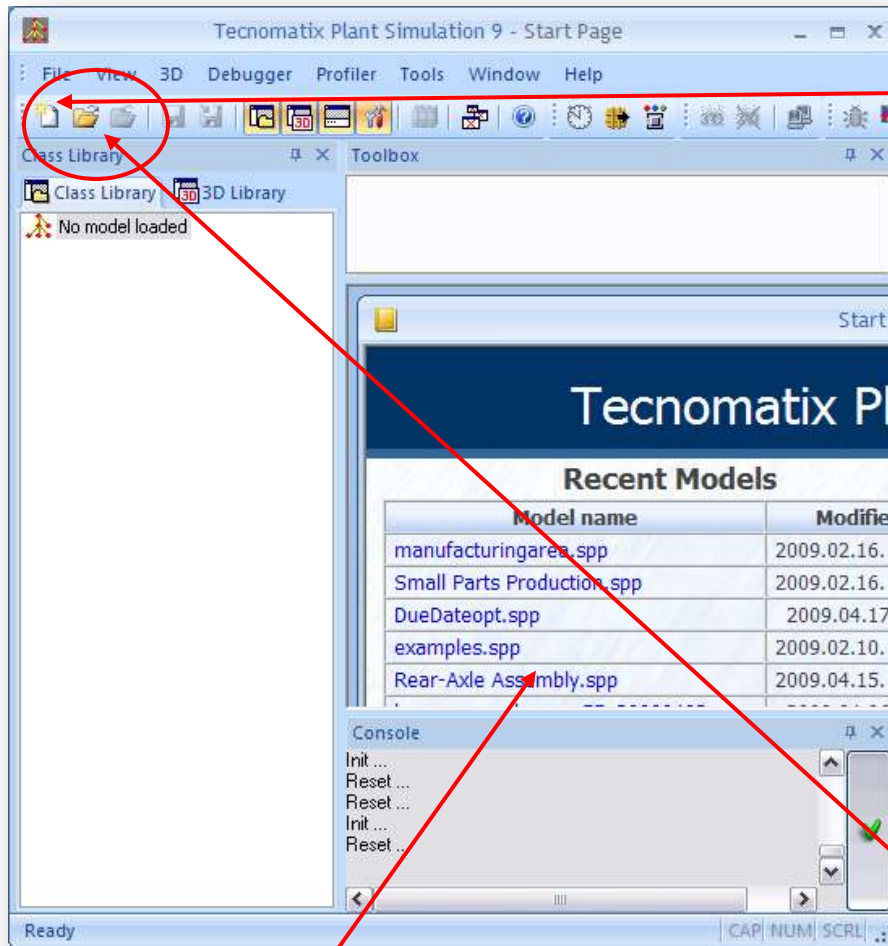
A Plant Simulation programablaka **menüket**, **eszköztárakat** és Viewer (néző) ablakokat tartalmaz:

- az üres **Toolbox**
- az **Explorer** és
- a **Console**.

Megjelenítésük a

View > Viewers > megjeleníteni kívánt elem
menüpont alatt kapcsolható be.

Az **Info Pages** (információs lap) hivatkozásokat tartalmaz a Plant Simulation weboldalaira és számos kapcsolódó témára.



Erre a gombra kattintva, vagy a menü

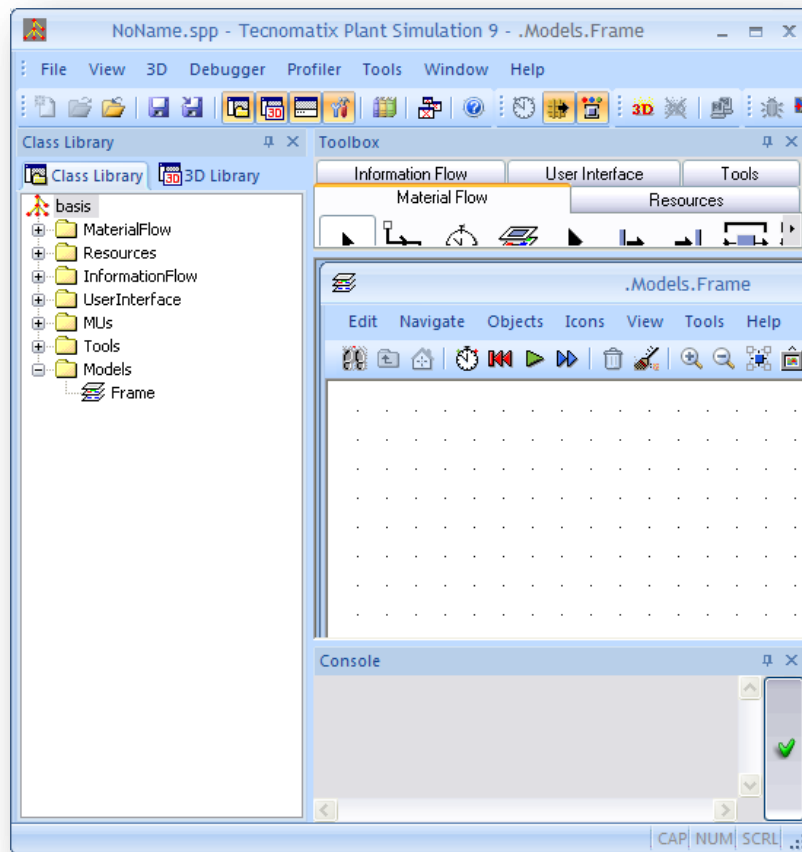
File > New Model

parancsát kiválasztva új modellfájl nyílik meg. Ez a fájl tartalmazza a *Class Library* - t (osztálykönyvtárt) a beépített Plant Simulation objektumokkal.

Kattintson erre a gombra meglévő modellfájl megnyitásához, vagy válassza ki a következő menüparancsot

File > Open Model

A legutóbb használt fájlok listája a legutóbb használt nyolc darab modellt mutatja.



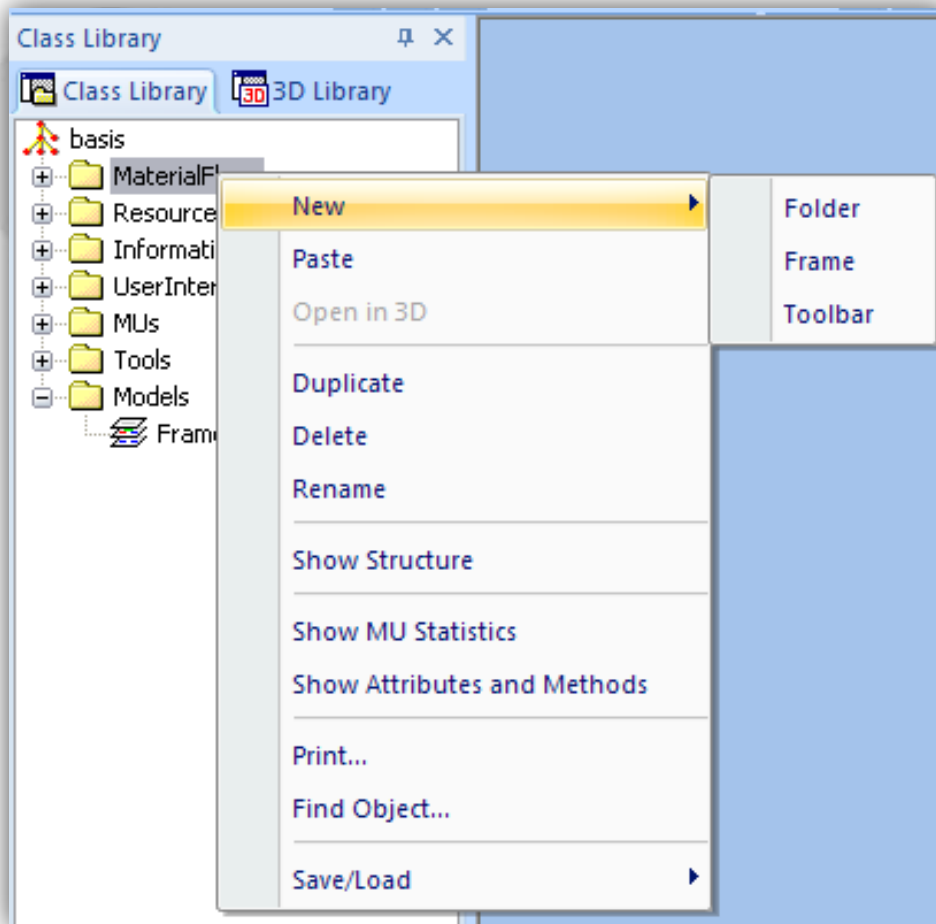
A modellfájl tartalmazza a Class Library - t a beépített objektumokkal hierarchikus faszerkezetben.

A különböző mappák különböző objektum osztályokat tartalmaznak.

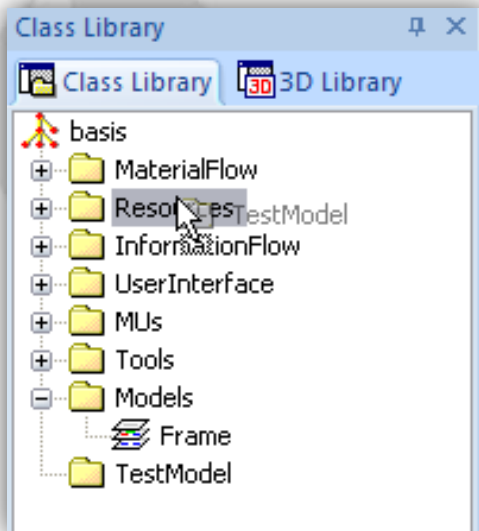
A struktúra módosítható és bővíthető a szimulációs projekt igényeinek megfelelően.

Például készíthetők új mappák a szimulációs modellek tárolására, tesztelésre.

* MU: mozgó objektumok, amelyek az egyik anyagáramlási objektumtól a másikig mozognak a szimulációs modellben



Új mappa (*Folder*) vagy *Frame* vagy eszköztár hozzáadása a tárolásukra használt mappa helyi menüjéből a *New* menüpontot és a hozzáadni kívánt objektum típusát kiválasztva lehetséges.

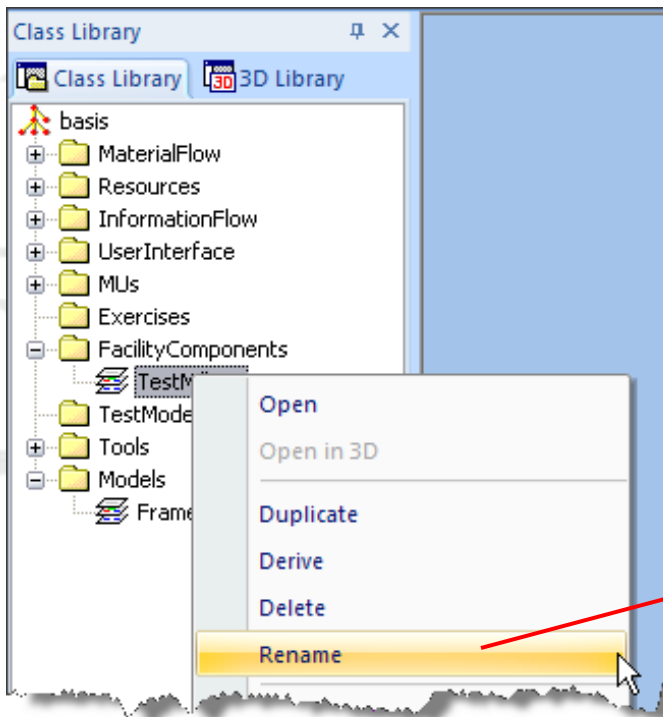


Egy mappa vagy objektum átmozgatása ugyanabban a struktúrában (azonos szinten) egyszerűen húzd és ejtsd módszerrel történik.

Egy mappa vagy objektum átmozgatása más struktúrába (más szintre) az elem kiválasztásával, majd a húzd és ejtsd művelet közben a **Shift** billentyűt nyomva tartva lehetséges.

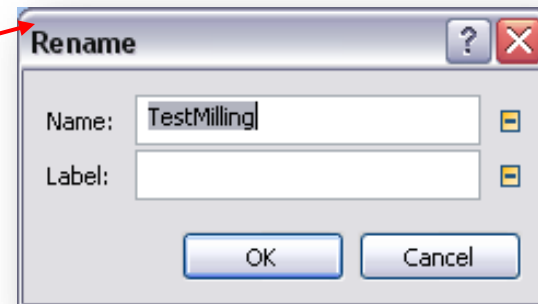


Fontos tudni, hogy a struktúra módosítása a mappanevek megváltoztatása egy meglévő szimulációban azt okozhatja, hogy a szimuláció a továbbiakban nem fog működni.



Objektum átnevezése:

- Kétszer kattintva rajta,
- Az **F2** billentyű lenyomásával,
- A helyi menüből a **Rename** parancs kiválasztásával lehetséges.



Egy megnyitott objektum az **F2** billentyű lenyomásával nevezhető át a **Rename** párbeszédablakban.

- **Betűvel** kell kezdődni, amit betűk vagy számok követhetnek. Aláhúzás (_) használható. **Ne használjon speciális karaktereket!**
- Nem érzékenyek a kis- és nagybetűkre, ezért használhatók a kis- és nagybetűk tetszőlegesen.
- Nem tartalmazhatnak kulcsszavakat, mint például **if, then, else, from, until, loop, result, ...**
- Egy névterületen belül csak egyszer fordulhatnak elő.

Példák:

Helyes: `Station_1, Station1, millingMaschine, conveyor`

Helytelen: `Station 1, IF, 35-conveyor, 275Station`

Mentés történhet az eszköztár szabványos mentés gombjára kattintással, vagy a következő menüpont kiválasztásával:

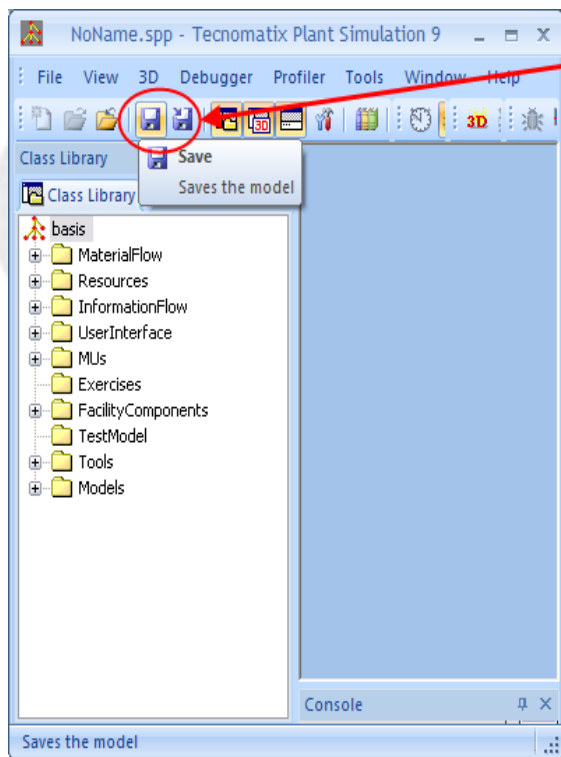
File > Save Model vagy **File > Save Model As**

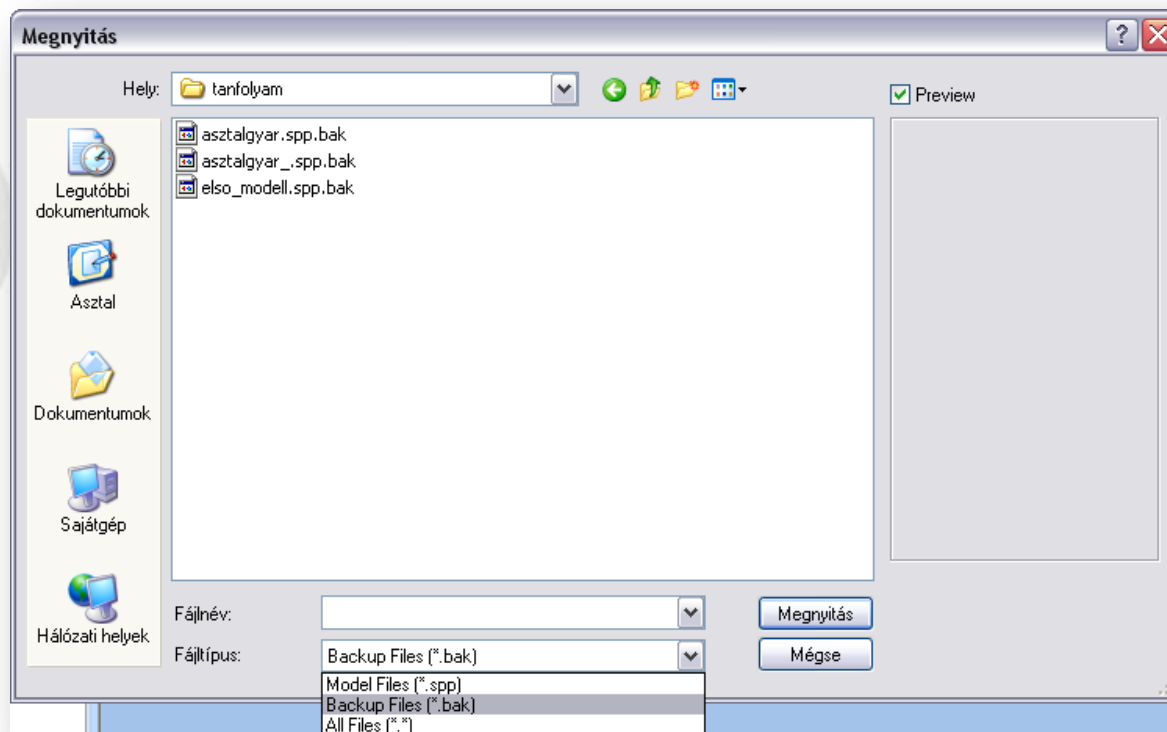
A modell első mentésekor a Plant Simulation létrehozza a fájlt **.spp** kiterjesztéssel.

Ezek után a Plant Simulation a modell változásait ebbe a modellfájlba (*.spp) menti.

Ezzel egy időben a Plant Simulation átnevezi az előzőleg mentett verzióját a modellfájlnak **.spp.bak** kiterjesztésű, azonos nevű fájlra.

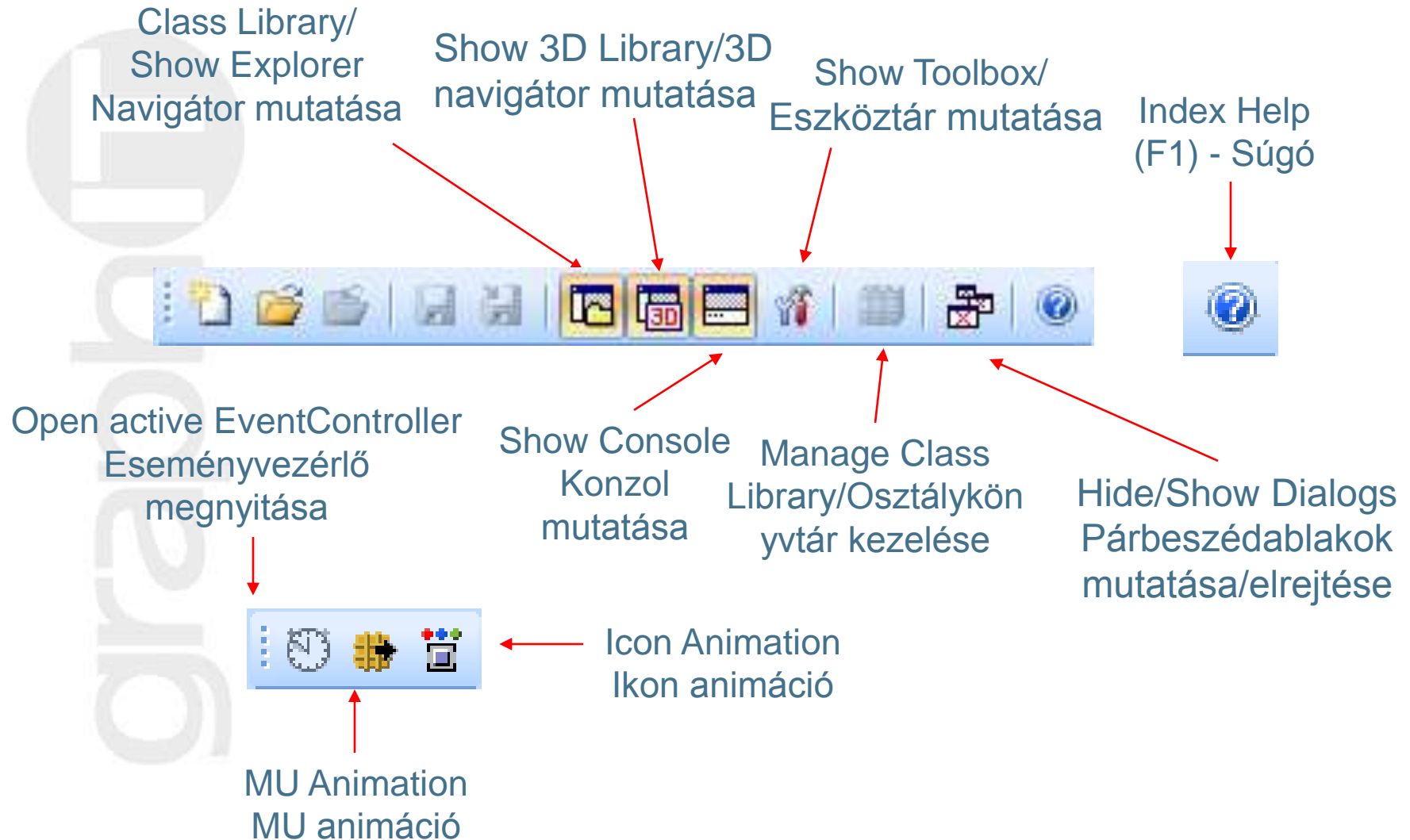
Ez a biztonsági mentésfájl a modell utoljára mentett változatát tartalmazza. A Plant Simulation - ben szükség esetén a **.bak** fájl közvetlenül megnyitható.



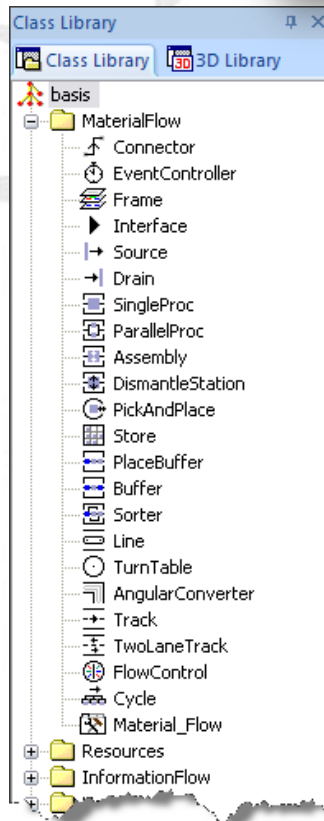
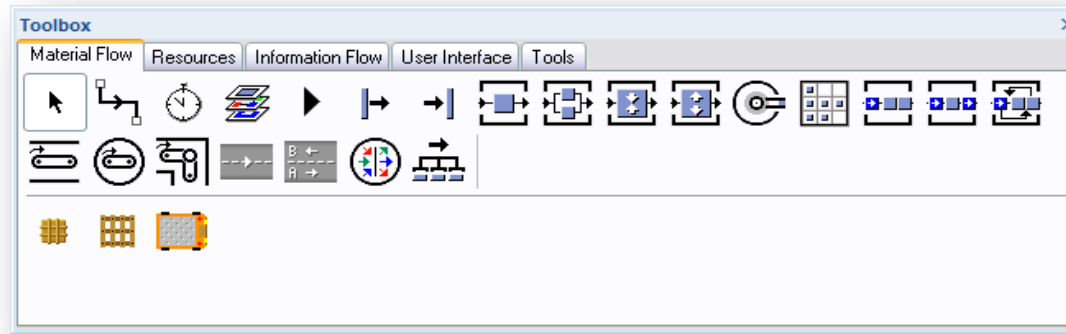


A lenyitható listából kiválasztható a megnyitni kívánt modell típusa **Files of type:** **.spp** (modellfájl) vagy **.bak** (biztonsági mentésfájl).

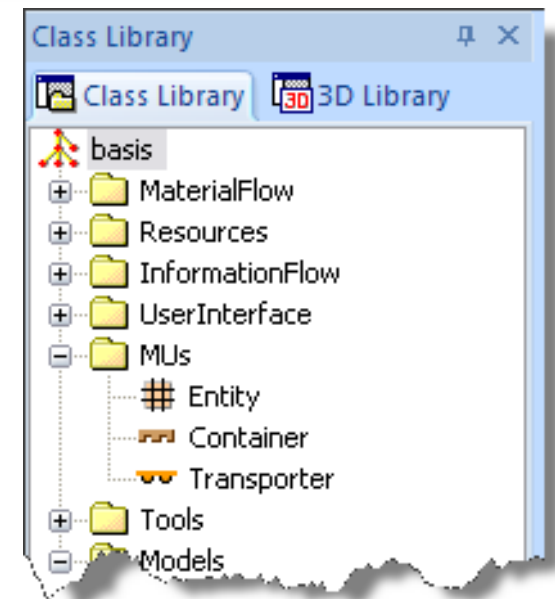
Ha a modell 3D modell, akkor előnézet (preview) is látható róla.

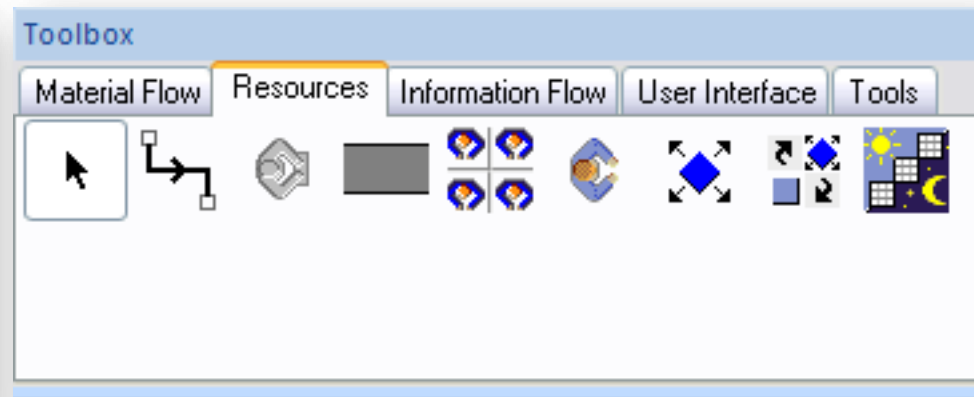
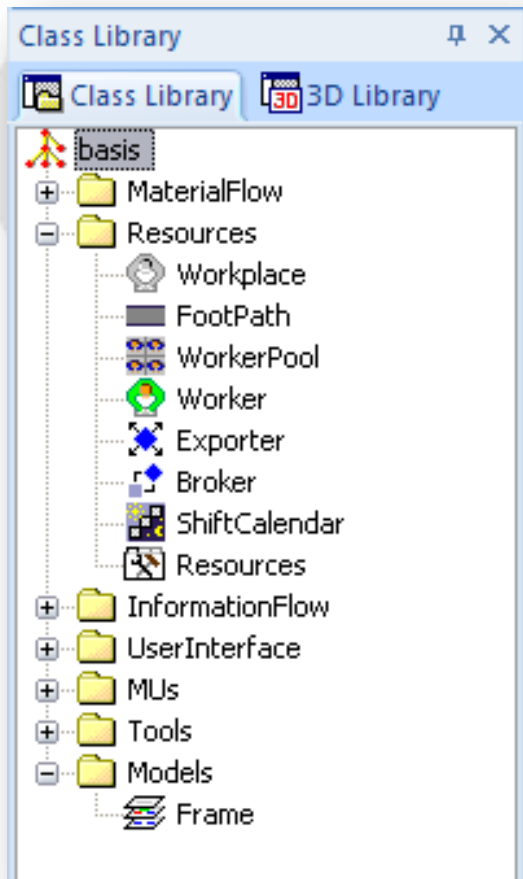


Anyagáramlás (material flow) objektumok és mozgó objektumok (MU)



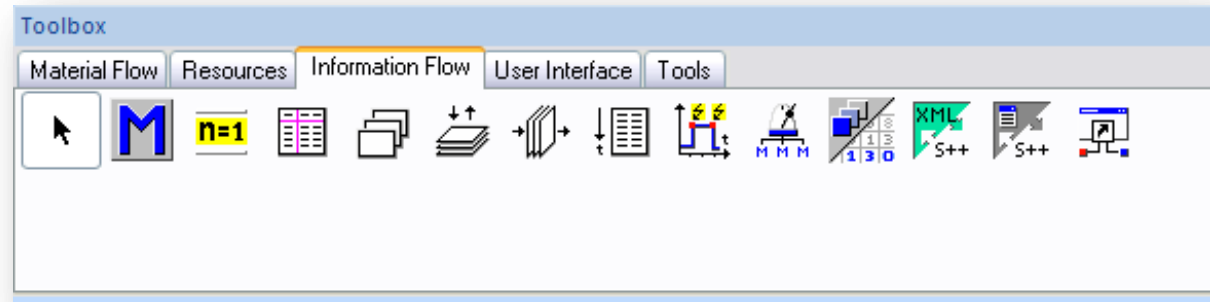
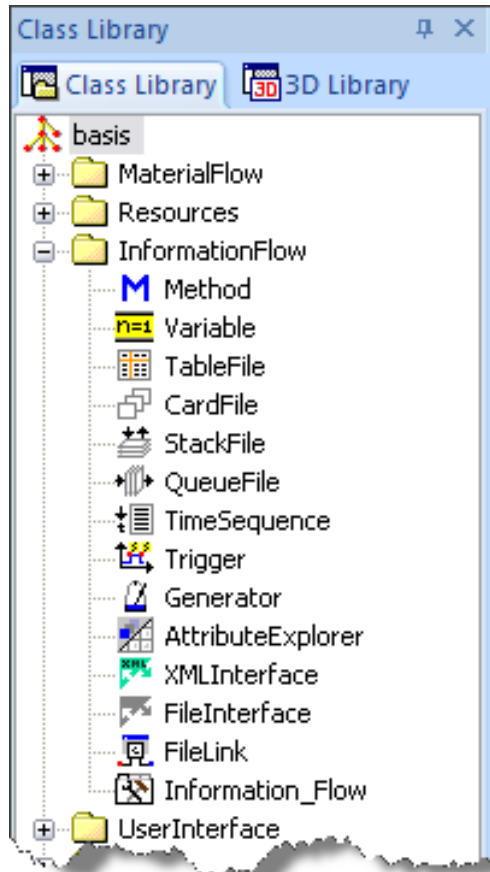
- Ezekkel lehet modellezni az anyagáramot a modellben.
- Az objektumok között megtalálhatók az egyszerű feldolgozó elemektől a bonyolult szerelő elemekig számos objektum



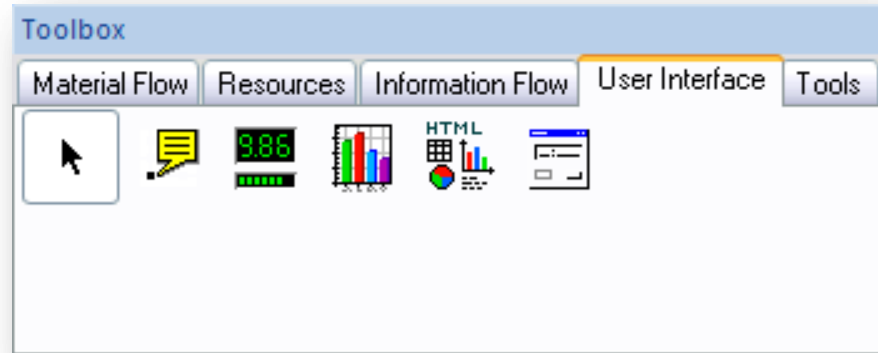
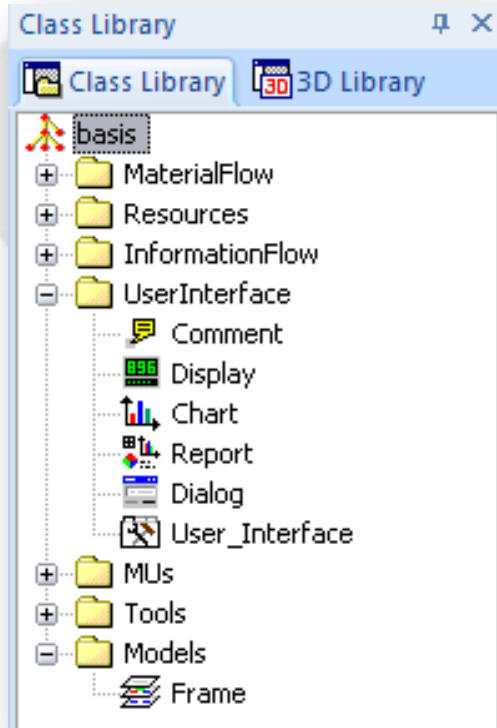


Dolgozók és más osztott erőforrások (pl. szerszámok) modellezésére szolgálnak

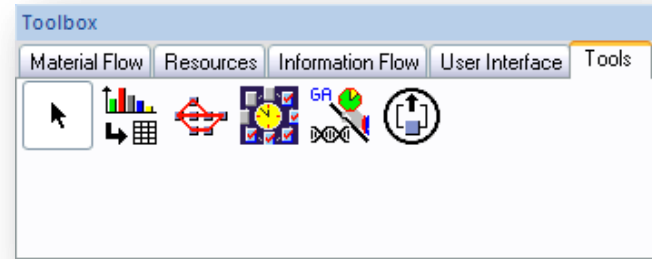
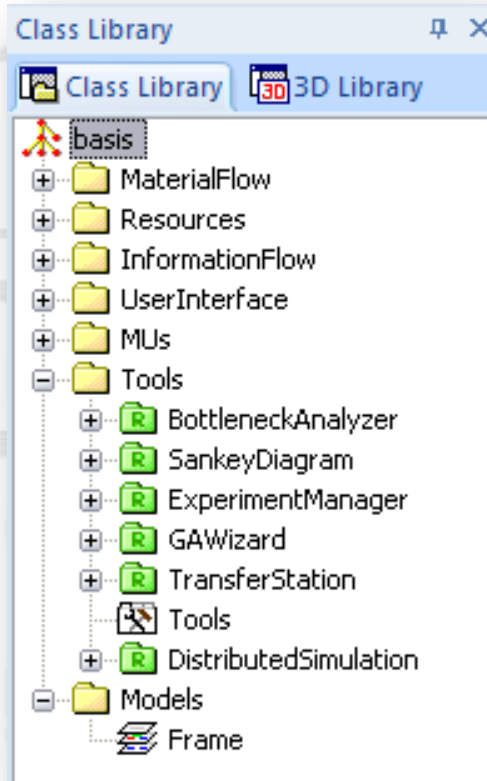
Információ áramlás (information flow) objektumok



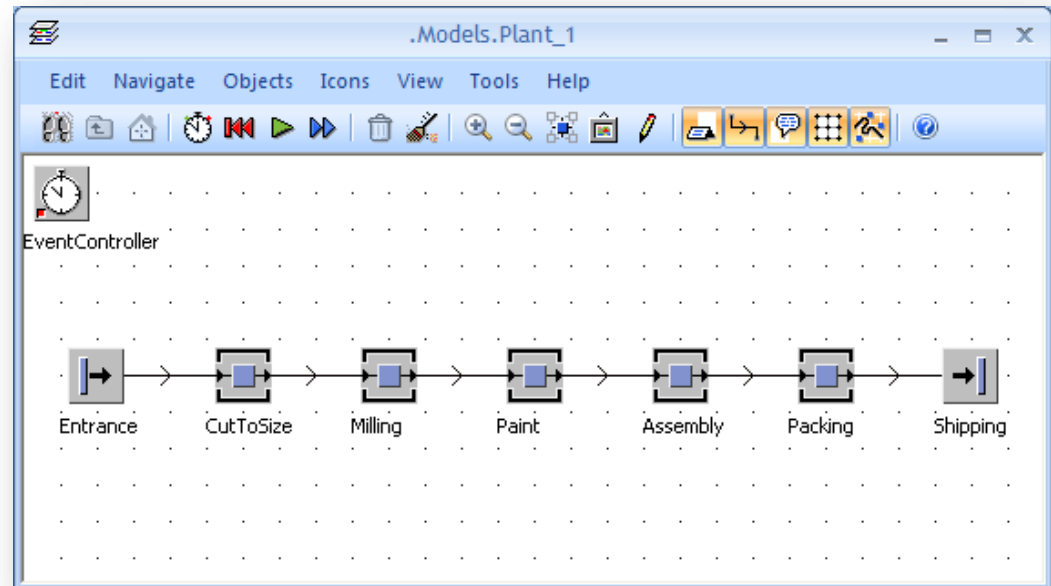
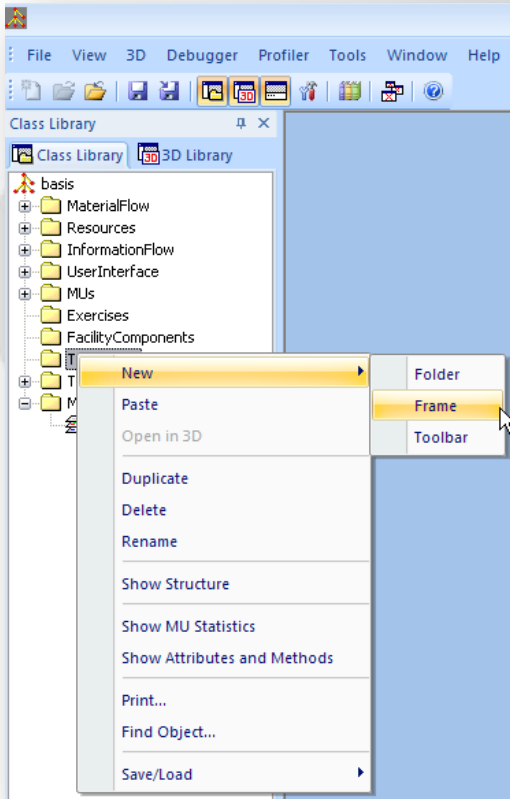
Kontroll stratégiák készítésére, információ kezelésére és
eredmények feldolgozására használhatók



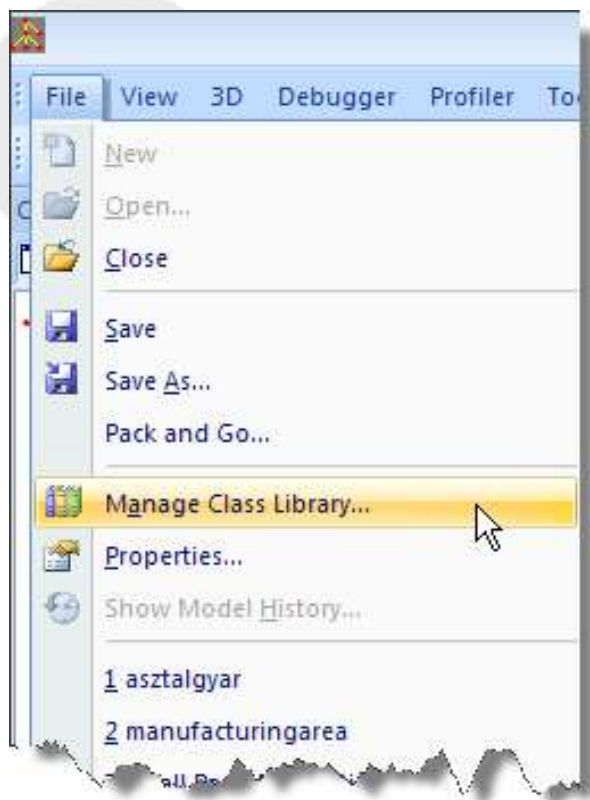
Eredmények gyűjtésére szolgálnak, és
felhasználói felület készítésére



Információ gyűjtő objektumok (*Sankey*,
Bottleneck Analyzer) és speciális objektumok
(*PortalCrane* és *TransferStation*)



A szimulációs modellek keretekben (Frame) készíthetők el anyagáram objektumok beillesztésével és összekapcsolásával. A szimuláció is a kereten belül futtatható le. Tetszőleges számú új Frame készíthető a hierarchia bármely szintjén a Class Library - ban. A hierarchia elemének helyi menüjében a **New > Frame** menüpontot kiválasztva készíthető el az új keret.



A Class Library kibővíthető további Plant Simulation alkalmazásokkal.

Általában ezek opcionális modulok, amelyekhez külön licenc szükséges, és bizonyos speciálisabb modell optimalizációs feladatokra nyújtanak megoldást.

A Class Library kezelése, bővítése a File > Manage Class Library... menüparanccsal lehetséges.

ActiveX

ActiveX vezérlőkhöz szükséges interfész, amivel ilyen vezérlőkkel lehet adatot cserélni.

Aris

Aris - IDS, Prof. Scheer üzleti folyamatok modellezésére szolgáló objektumokhoz szükséges interfész.

GA

Modell automatikus optimalizálására szolgáló genetikus algoritmus.

Gantt

Termelési tervek grafikus megjelenítése és interaktív módosítása.

ODBC

Kétirányú adatcsere interfész Windows ODBC alapú adatforrásokhoz.

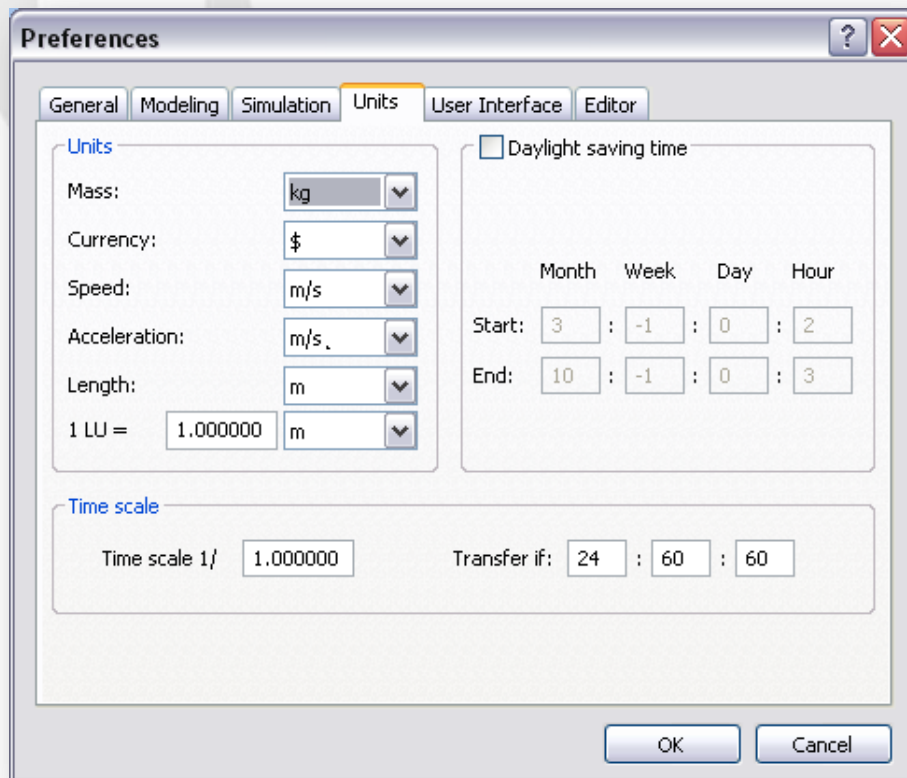
Oracle

Tetszőleges Oracle adatbázishoz kapcsolódás interfésze.

Socket

Szabványos TCP/IP protokoll alapú alkalmazásokhoz kapcsolódás interfésze.

A Plant Simulation beállítások a **Tools > Preferences** vagy **Tools > Model Settings** menüpontok alatt adható meg. A Preferences beállításai új modellekre vonatkoznak, míg a **Model Settings** beállítások az aktuális modellre.



A Plant Simulation a megadott értékeket automatikusan konvertálja a **Tools > Options > Units** menüpont alatt megadott **Mass (tömeg)**, **Speed (sebesség)** és **Length (hossz)** mértékegységekre, ha más mértékegységet választ.

A **Length (hossz)** esetében megadható olyan **Length Unit (hossz egység)**, amely megfelel az igényeknek, mint pl: 1LU = 0.341 miles.

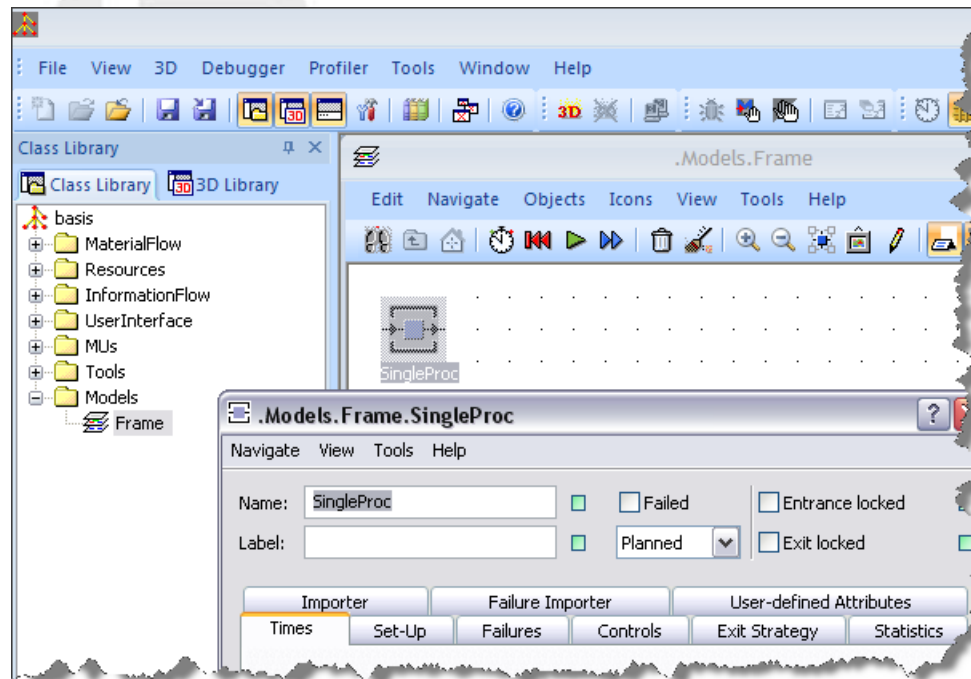
A Plant Simulation nem konvertálja a **Currency (pénznem)** értékét.

A **Time scale** (idő mérték) megadható az igényeknek megfelelően a megfelelő értéket beírva a szövegmezőbe.



4. fejezet

A Plant Simulation
felhasználói felülete



A Plant Simulation egy több dokumentumos alkalmazás, amely szabványos Windows felhasználói felületet nyújt.

Különböző jellegű ablakok vannak. A párbeszédablakok általában más ablakok előtt jelennek meg.

A menü és az eszköztárak, akárcsak a megjelenítő ablakok mérete, testreszabható.



A TUNE rendszer három típusú ablakot jelenít meg:

Viewer (megjelenítő) ablakok:

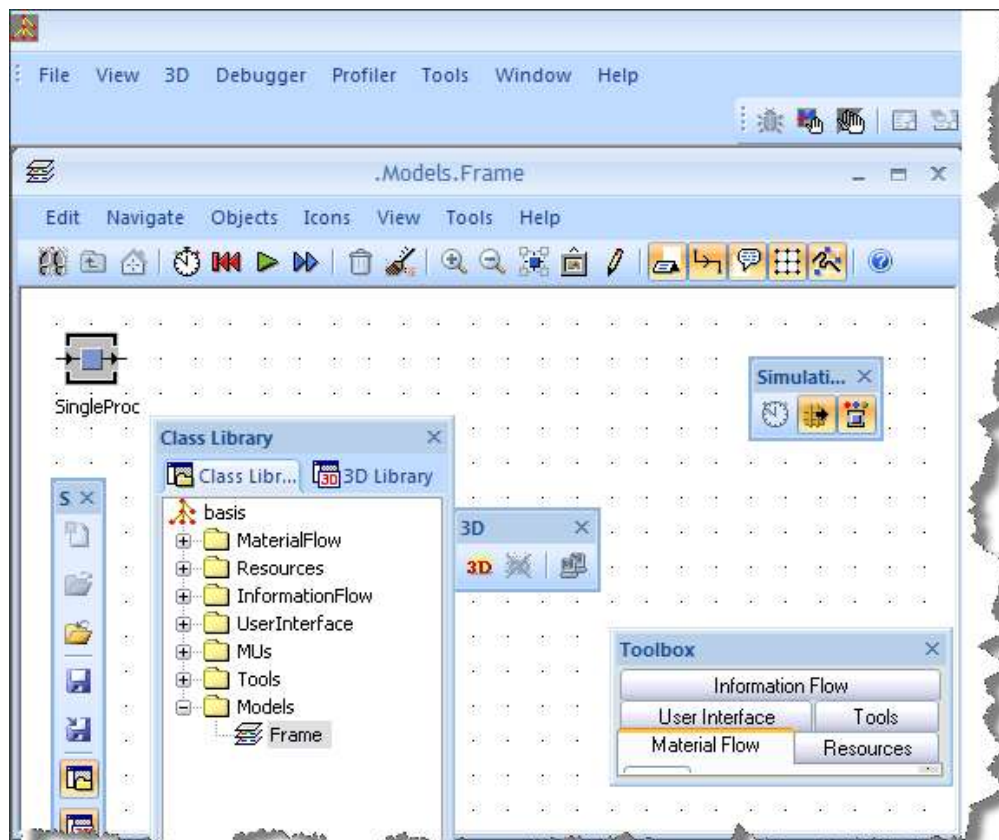
- **Explorer (Tab Viewer)**
- **Console**
- **Objects Palette (Toolbox)**

Objektum ablakok

- **Frame**
- **Method**
- **TableFile, StackFile, ...**

Párbeszéd ablakok

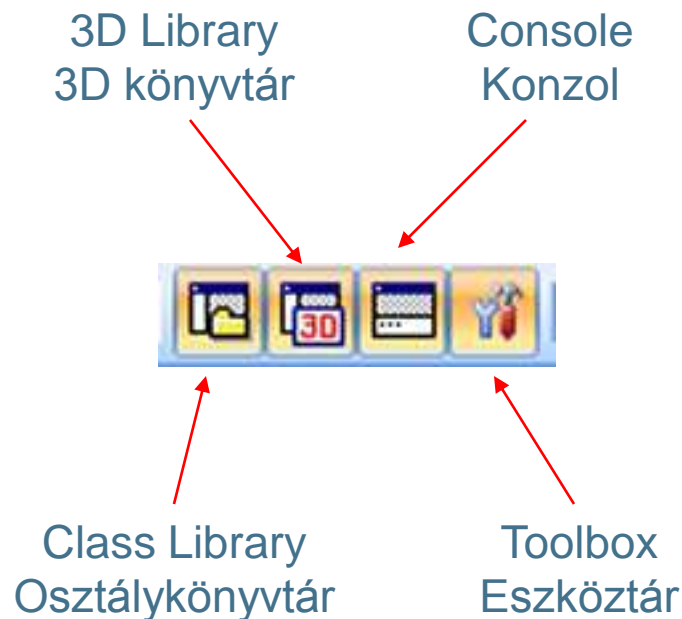
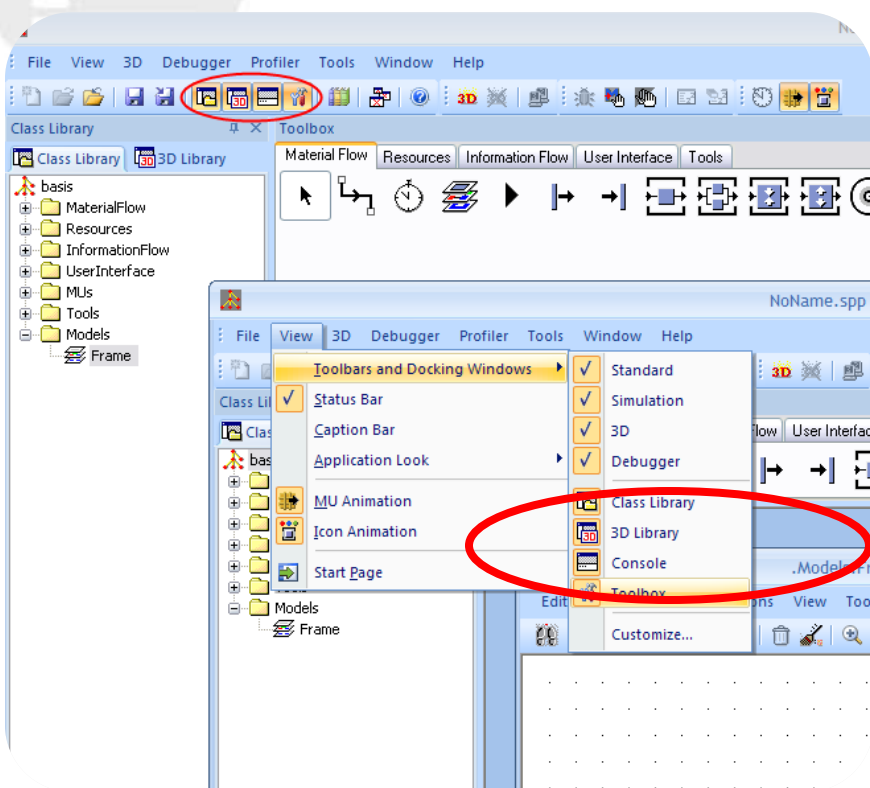
- **Az objektumokhoz**



A megjelenítő ablakok, a menük és az eszköztárak dokkolhatók, vagy a dokkolásból kivehetők és mozgathatók.

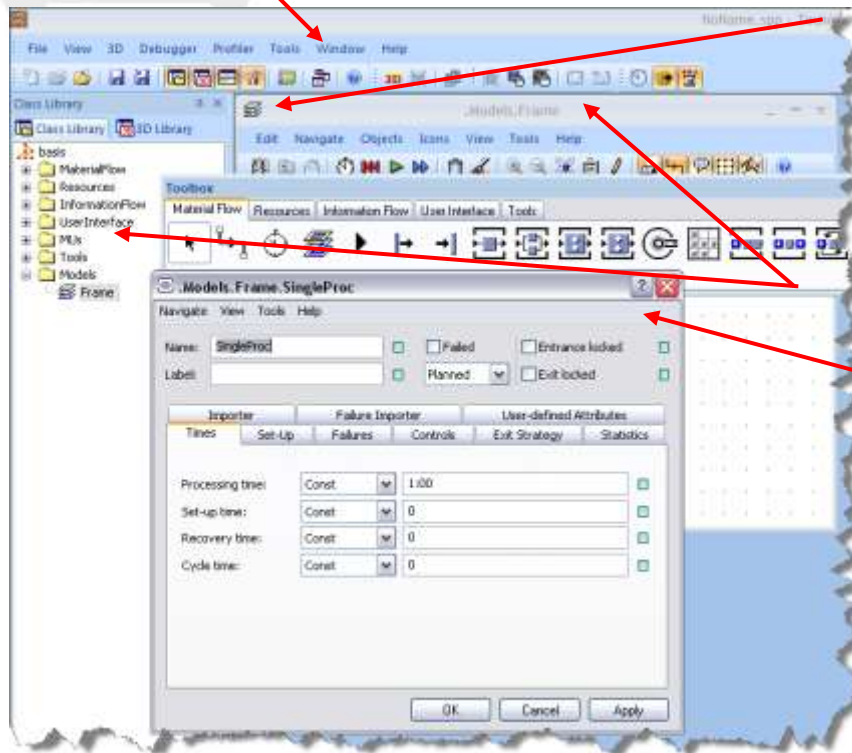
Ha egy megjelenítő ablakot mozgat, és nem akarja, hogy az ablak széléhez dokkoljon, tartsa lenyomva a **Ctrl** billentyűt.

A megjelenítő ablakok bezárhatók és megjeleníthetők megint az eszköztáron az ikonjukra kattintva. Újra megjeleníthetők a **View > Viewers** menüparancs alatt a megfelelő ablak nevének kiválasztásával.



A megnyitott objektumablakok a következő menüpontokkal rendezhetők el:

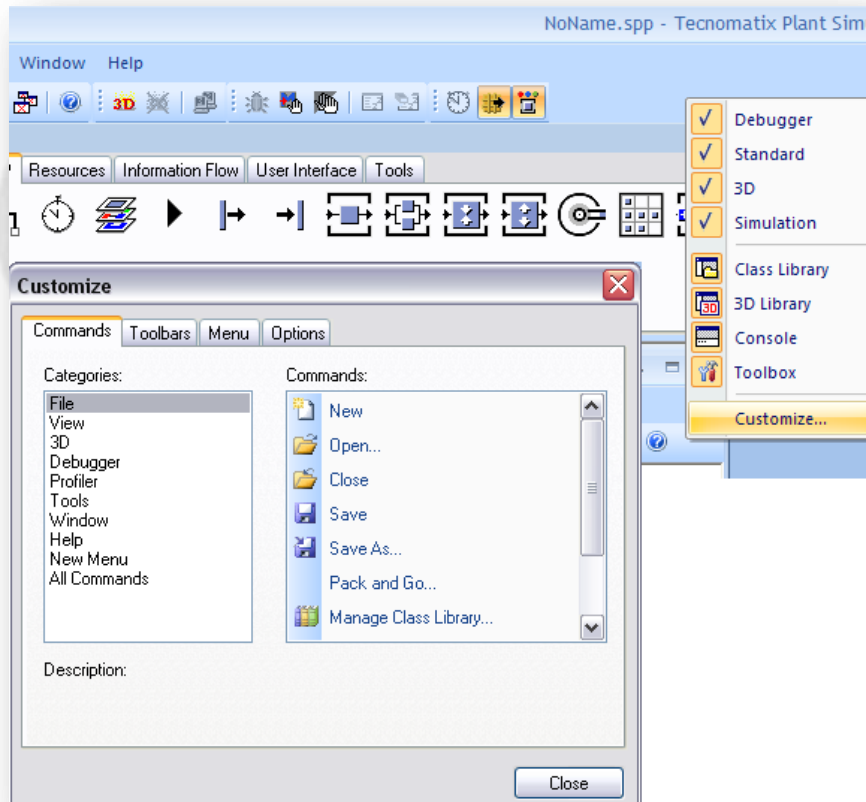
Window > Cascade (lépcsőzetes), **Tile Horizontally** (vízszintes mozaik),
Tile Vertically (függőleges mozaik).



Az objektumok ablakai a fő ablakon belül helyezkednek el, nem lehet azon kívül mozgatni.

A megnyitott megjelenítő ablakok az objektumablakok előtt jelennek meg.

A megnyitott párbeszédablakok a megjelenítő ablakok és az objektumablakok előtt jelennek meg. Ezekben értékeket lehet megadni szövegmezőkben, vagy értékeket lehet kiválasztani listadobozokban.



Plant Simulation menük és eszköztárak testreszabása:

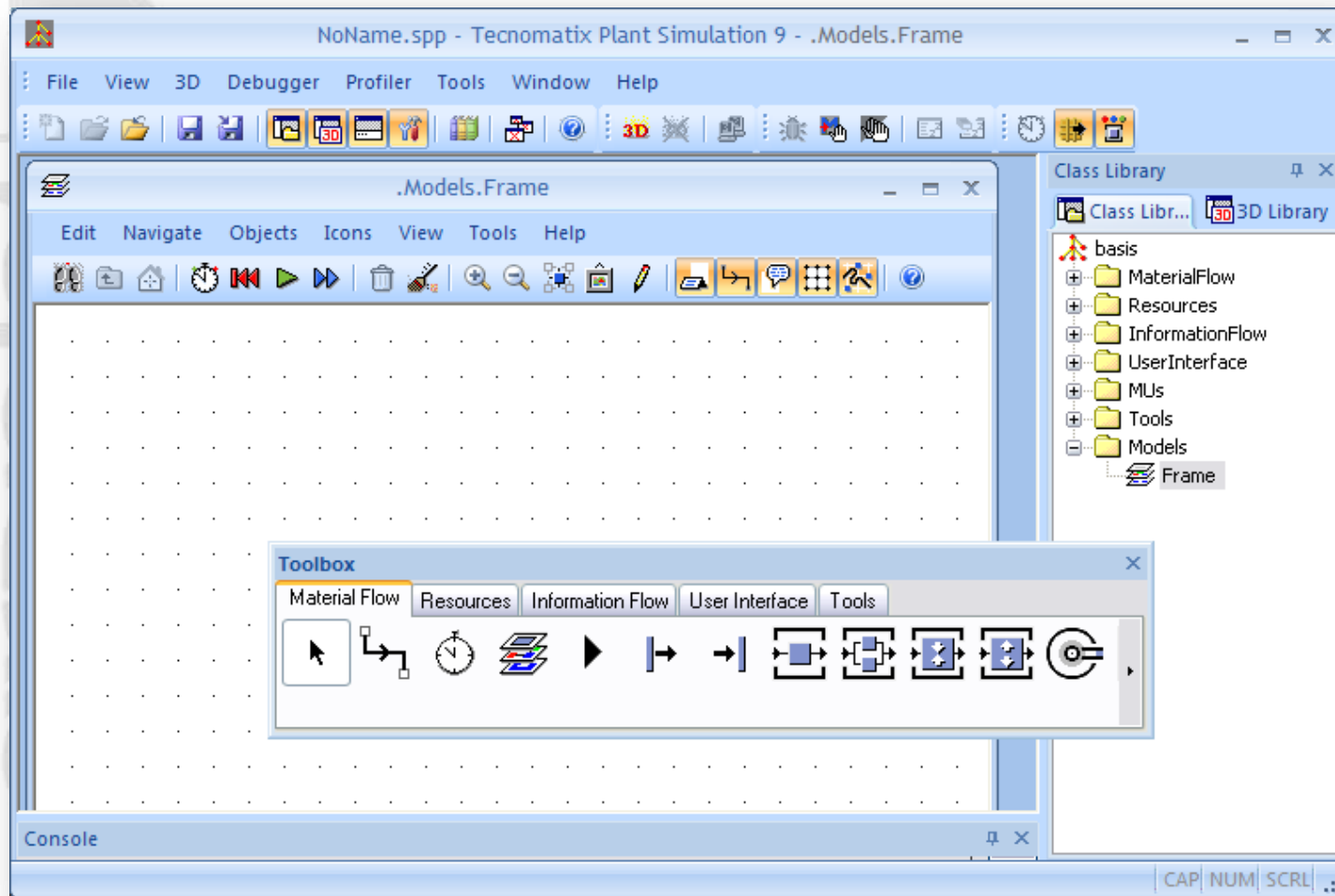
A menü vagy eszköztár helyi menüjéből a **Customize** (testreszabás) menüpontot kiválasztva. Ekkor megjelenik a Customize párbeszédablak.

Megjelennek a parancsok és kategóriáik, és a hozzájuk kapcsolódó menüparancsok és ikonok.

Válassza ki a parancsot a listában, és húzza a kívánt helyre a menüben vagy az eszköztáron, és ejtse oda.

1 .lecke: Munka a Plant Simulation ablakokkal

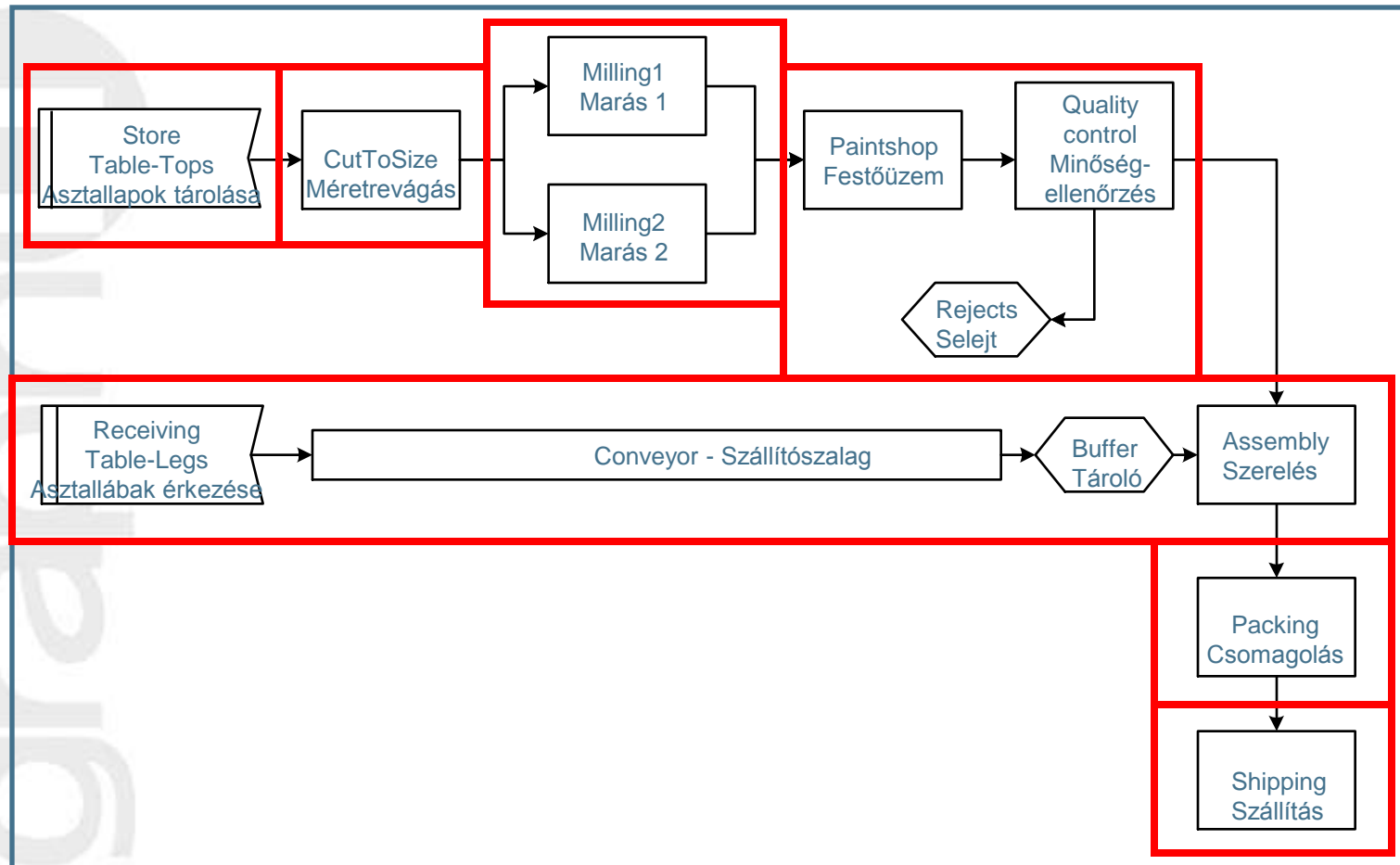
- Próbálja ki a megjelenítő ablakok mozgatását, dokkolását, dokkolásból kivételét.





5. fejezet

Létesítmény szimulációja

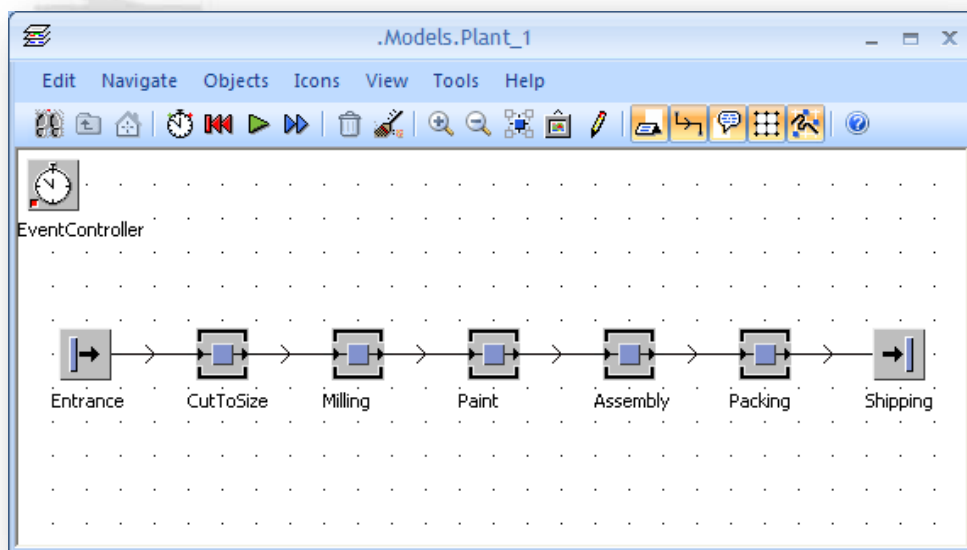


A tesztgyár asztalokat készít:

- Először egy fűrészállomáson az asztallapokat méretre vágják.
- Aztán két marógép megmunkálja az asztallapok éleit.
- Ezután festőrobotok az asztallapokat különböző színűre festik. A minőségellenőrzés a hibásan festett lapokat visszaküldi a selejt állomásra, ahol a festést kijavítják.
- A szerelőállomáson az asztallapokhoz hozzáadják az asztallábakat.
- Az elkészült asztalok a csomagolásra mennek, ami után készek a szállításra.
- A becsomagolt asztalok az ügyfélhez történő szállításra várnak.

Először elkészül a gyártóüzem egy durva szimulációs modellje. Ebből a célból a gyártást kisebb egységekre kell felbontani, amelyek megfelelnek egy-egy feldolgozási cellájának a gyártásnak.

Mindegyik egységet külön modellezzük, és az adott feldolgozó cella viselkedését a szükséges részletességgel dolgozzuk ki.



Így megismerjük a legfontosabb Plant Simulation objektumokat, és a működésüket. Emellett a szimulációs modell elkészítésének gyakorlására is mód nyílik.

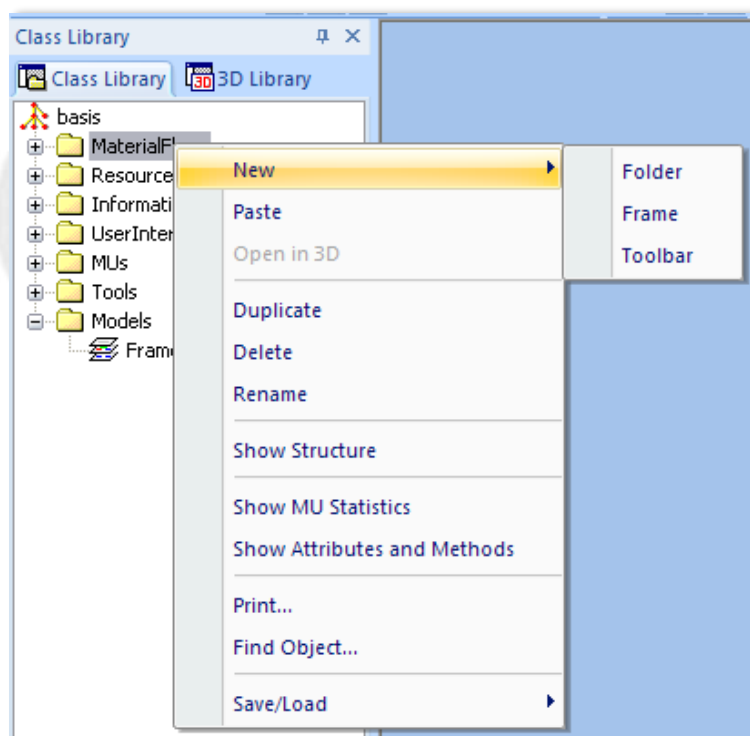
Föntről-lefelé stratégia:

Egy durván strukturált modellel jellemzett, ami fokozatosan lépésről lépésre kerül lebontásra. Ahogy finomodik a modell úgy lesz egyre részletesebb is. Ez a megközelítés jelentős absztrakciót igényel.

Lentről-fölfelé stratégia:

Egy részletes modellel jellemzett, előre gyártott struktúrákból, jelentősen alacsonyabb absztrakciós szinten. Lehetővé teszi a részletek gyors elemzését, viszont megnehezíti a nagy egész áttekintését.

A tanfolyam során mindkét módszert használni fogjuk.



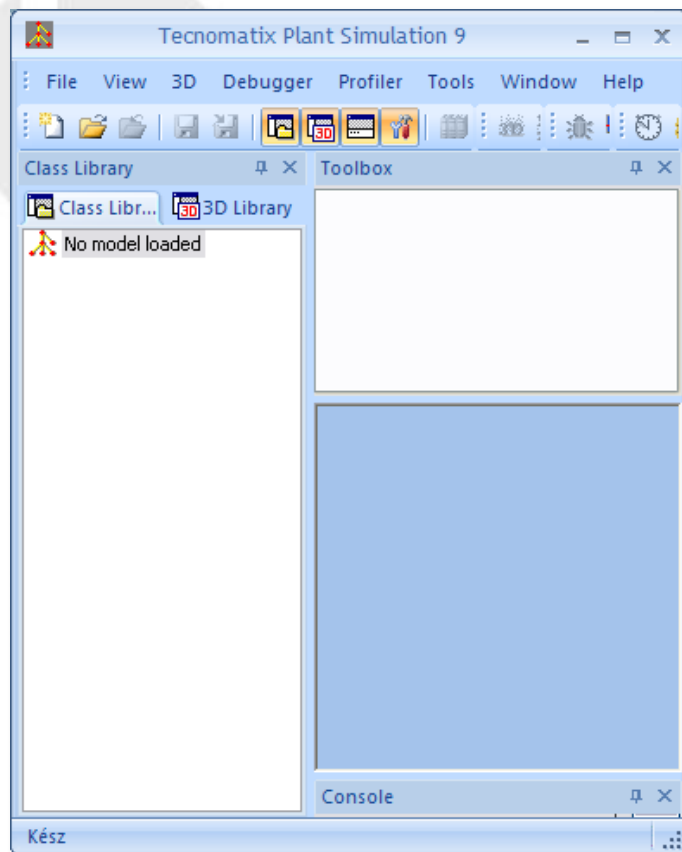
Hogy áttekinthetőek legyenek a komponensek, először elkészítjük a Class Library -ben a projekt struktúráját:

Új mappa készítéséhez válassza ki Class Library -ben a létrehozás helyén lévő mappa helyi menüjéből a

New > Folder parancsot.

Egy meglévő mappa átnevezéséhez válassza ki a mappa helyi menüjéből a **Rename** parancsot, vagy nyomja le az **F2** billentyűt.

Plant Simulation indítása



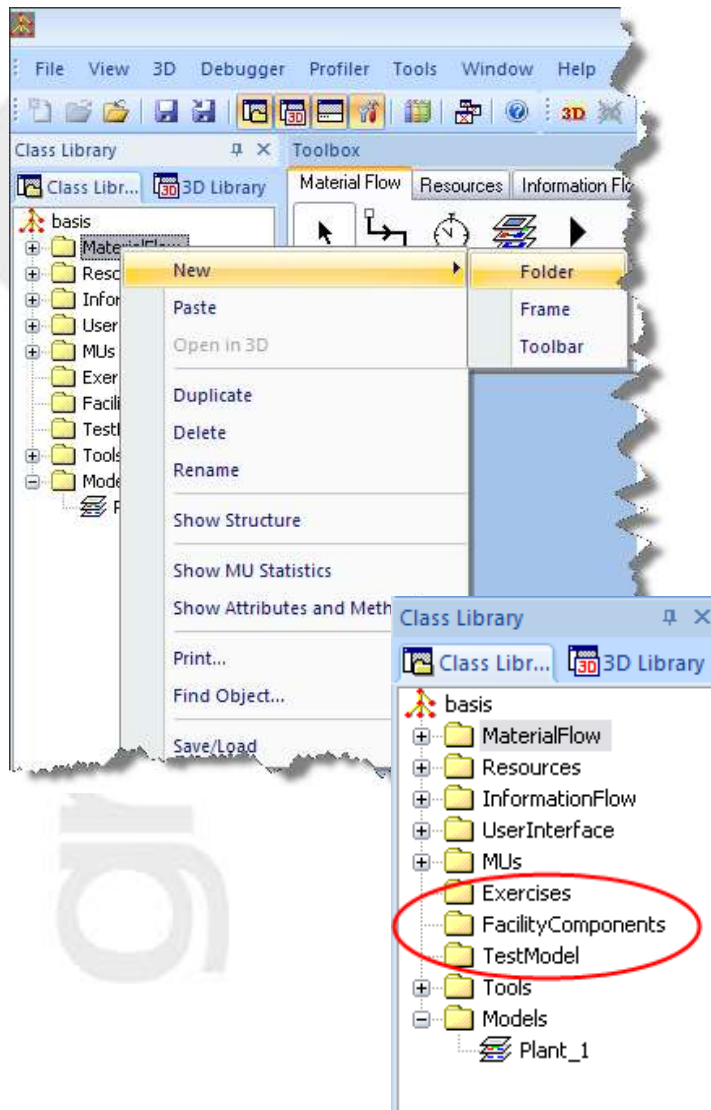
Zárjon be minden megnyitott modellfájlt.

Nyissa meg ezt a modellfájlt:

Mappa: [az oktató által megadott útvonal]

Fájl: **StartingModel_M1_V76**

Mentse el a modellt tetszőleges saját néven!



A Class Library (osztálykönyvtár) szerkezete

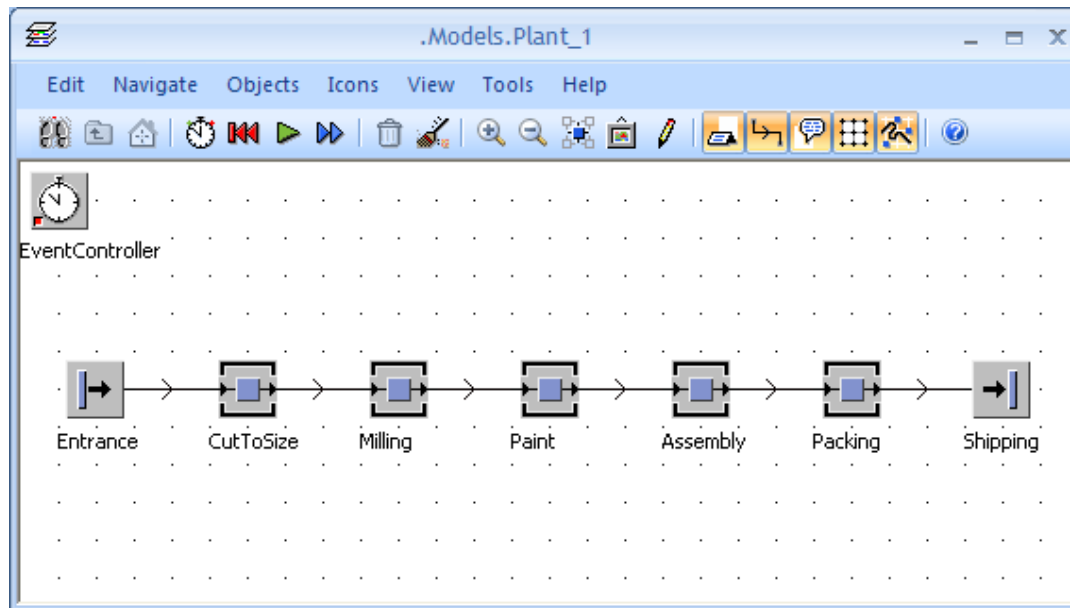
Készítse el az ábrán látható mappaszerkezetet a *Class Library*-ban az osztálykönyvtár legfelső szintjén:

- Exercises
- FacilityComponents
- TestModel



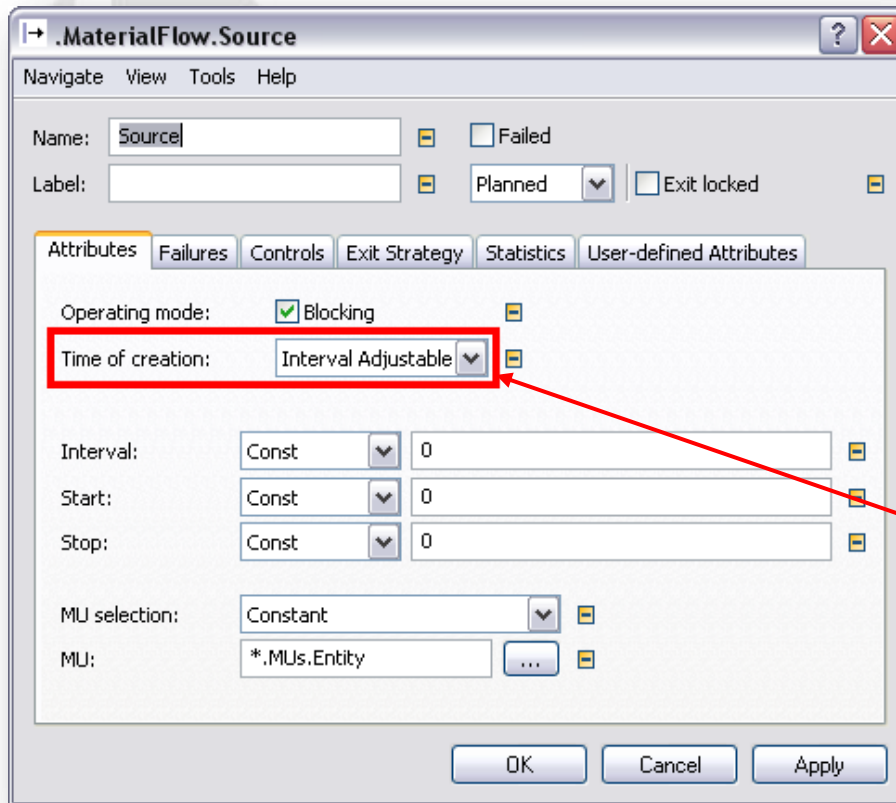
6. fejezet

A szimulációs modell elkészítése



A **fentről-lefelé stratégiával** elkészítjük a létesítmény durva modelljét egy Frame-ben. A gyár minden egyes cellájának modellezésére egy *SingleProc* objektumot használunk.

Először megismerjük a lépéseit a modell létrehozásának, és utána ezeket a lépéseket a gyakorlatban is alkalmazzuk.



Jellemzők:

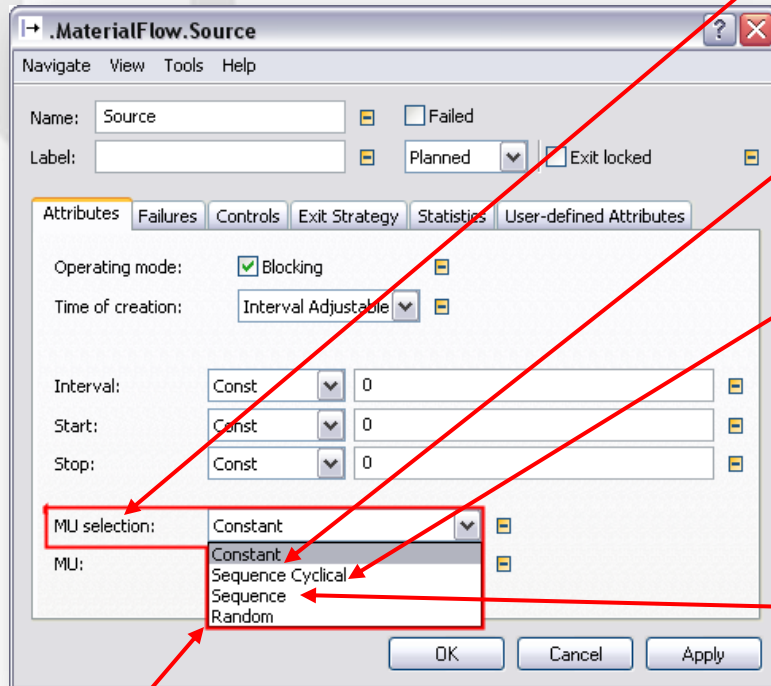
- Ikon:
- Kapacitás: bármennyi
- Aktív anyagáramlás objektum

A Source hozza létre az MU-kat (mozgó egységek), a megadott paraméterek alapján:

Time of creation (létrehozás ideje)

Interval Adjustable (intervallum),
Number Adjustable (szám) vagy
megadott **Delivery Table** (kibocsátási táblázat) alapján.

A Source hozza létre az MU elemeket az **MU selection** lenyíló lista beállítása alapján:

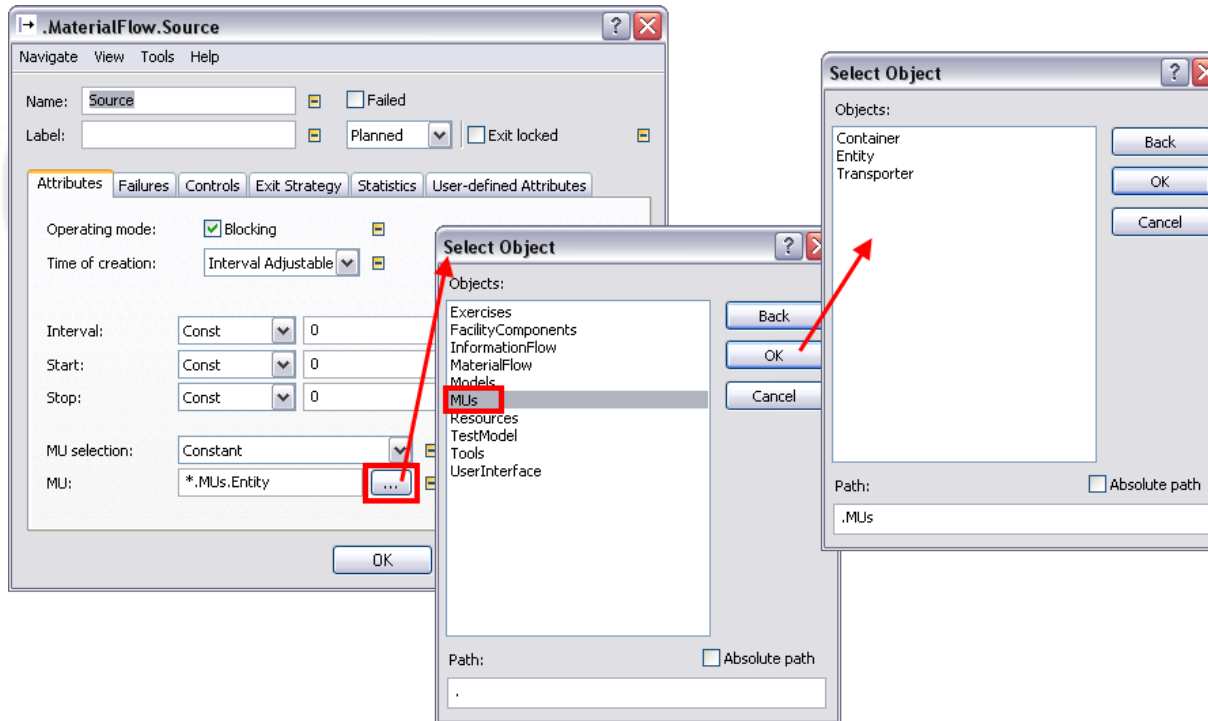


Constant (konstans): Amikor az MU típusa mindig ugyanaz (alapértelmezés).

Sequence Cyclical (ismétlődő sorrendben): A táblázatban megadott sorrendnek megfelelő az MU típus. A tábla további információt is tartalmazhat, a Plant Simulation a táblát ismétlődően dolgozza fel.

Sequence (sorrendben): a Plant Simulation a megadott nevű táblázatot egyszer dolgozza fel.

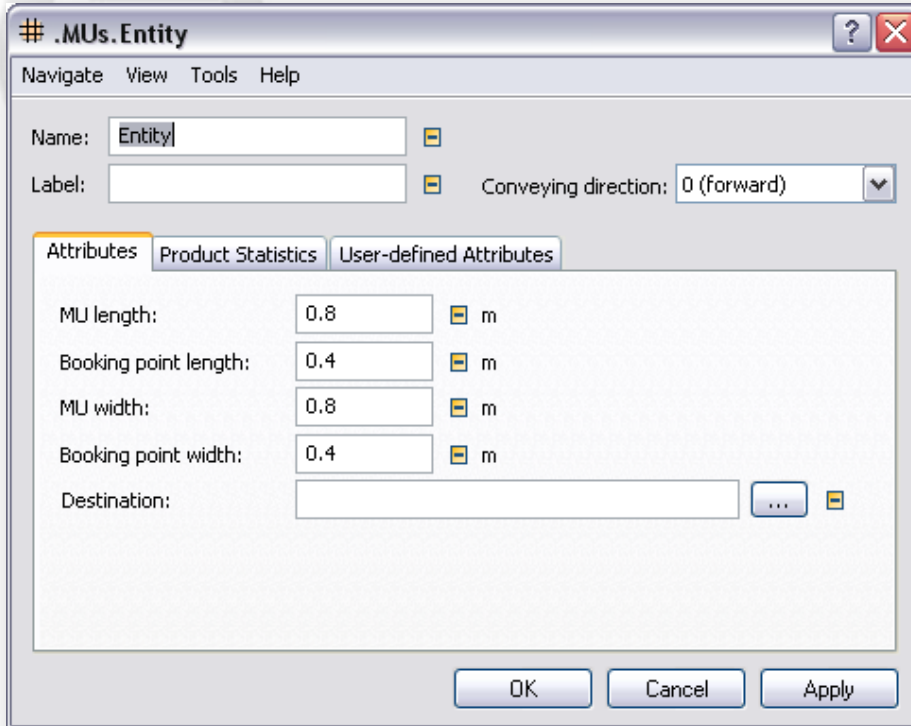
Random (véletlenszerűen): a Plant Simulation a táblában megadott MU-kat hozza létre véletlenszerűen a megadott módon.



Az MU kiválasztásához ki kell választani az MU melletti hárompontos gombra:


A **Select Object** párbeszédablakban ki kell választani az MUs mappát.

A kibocsátandó MU típusát ki kell választani ebből a mappából. Itt találhatók meg a felhasználó által definiált MU típusok is.

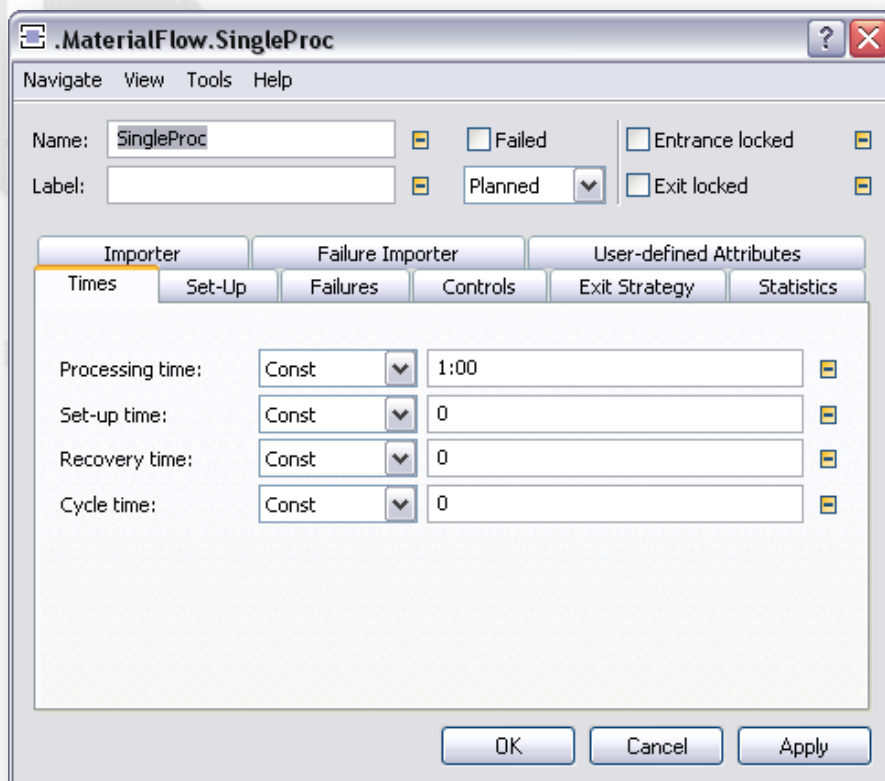


The screenshot shows a software dialog box titled ".MUs.Entity". It has a menu bar with "Navigate", "View", "Tools", and "Help". The "Name" field contains "Entity". The "Label" field is empty. The "Conveying direction" is set to "0 (forward)". There are three tabs: "Attributes" (selected), "Product Statistics", and "User-defined Attributes". Under the "Attributes" tab, there are five rows of attributes, each with a text input field, a unit icon, and a unit label: "MU length:" (0.8, m), "Booking point length:" (0.4, m), "MU width:" (0.8, m), "Booking point width:" (0.4, m), and "Destination:" (empty, with a dropdown arrow and unit icon). At the bottom are "OK", "Cancel", and "Apply" buttons.

Jellemzők :

- Ikon: 
- Kapacitás: 0
- Mozgó anyagáramlás objektum (MU)

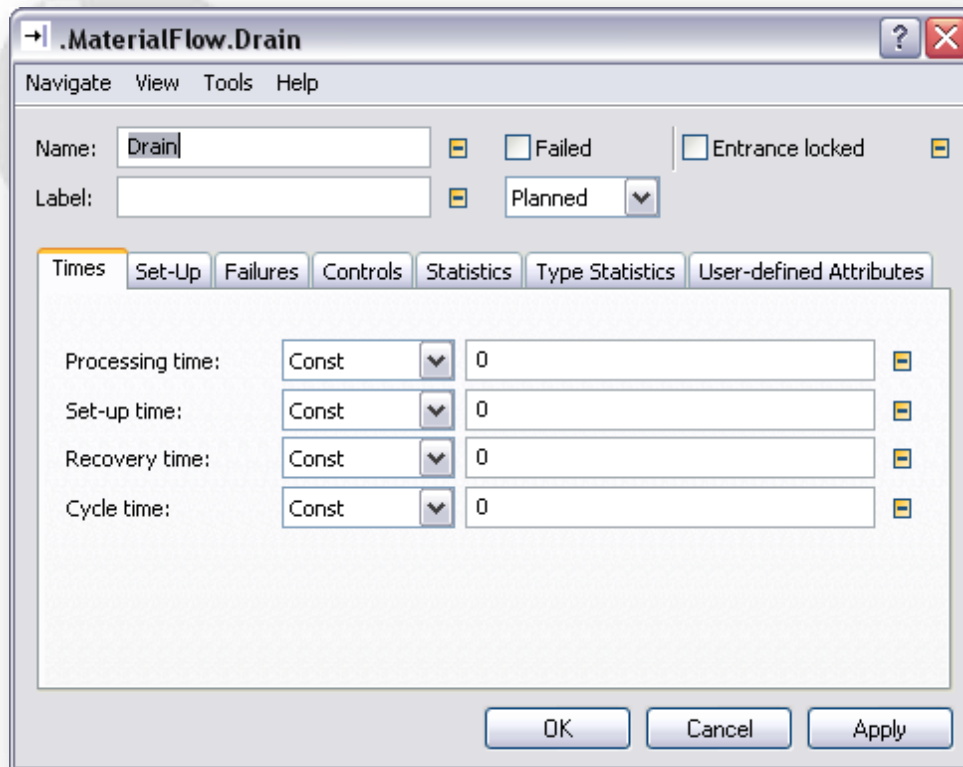
Az *Entity* egy olyan objektum, amit a *Source* hoz létre, az anyagáramlás objektumok dolgoznak fel, és a *Container* és *Transporter* objektumok szállítanak. Az *Entity* nem szállíthat, nem tartalmazhat másik alkatrészt vagy elemet.



Jellemzők :

- Ikon: 
- Kapacitás: 1
- Aktív anyagáramlás objektum

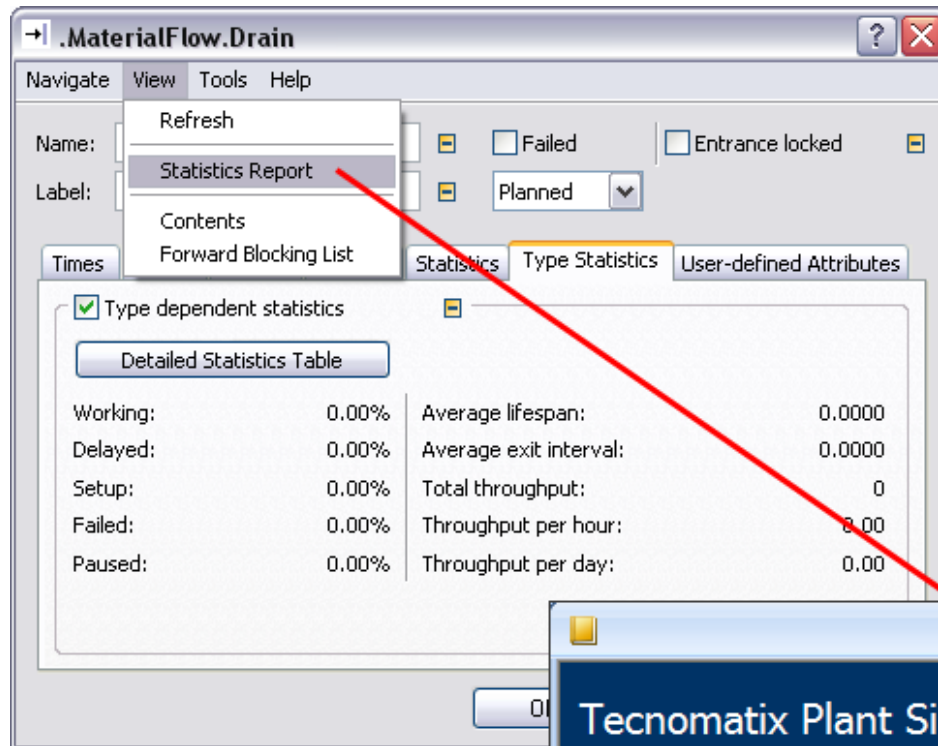
Egy MU-t fogad, feldolgozza azt, és továbbmozgatja a következő állomásra az anyagáramban. Ha az MU típusa különbözik a korábbi típustól, akkor a *SingleProc* objektumot fel kell készíteni a feldolgozására.



Jellemzők :

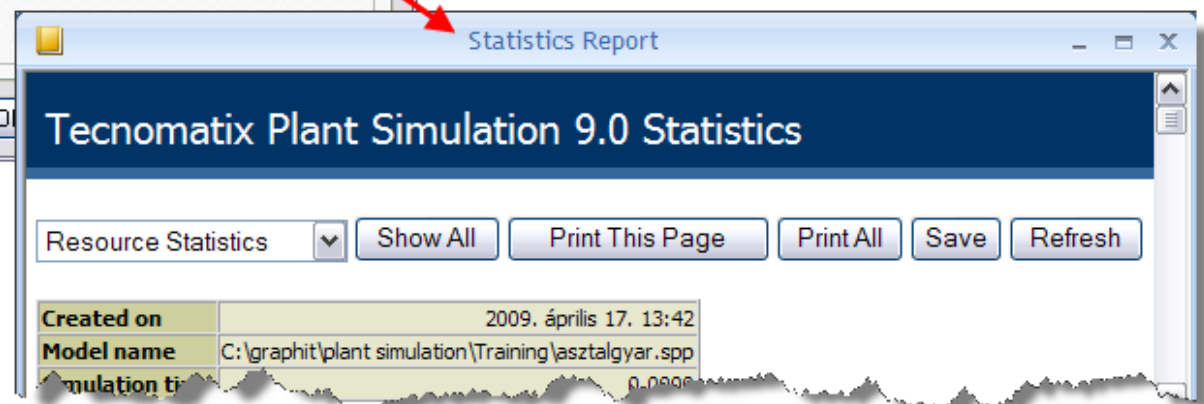
- Ikon: 
- Kapacitás: 1
- Aktív anyagáramlás objektum

Eltávolítja az MU-t az anyagáramból amint a megadott set-up time (beállítási idő) és processing time (feldolgozási idő) eltelik.



A Drain statisztikát készít az érkezett MU-król.

Az MU statisztikai adatai és a teljes statisztika megjeleníthető, illetve MU típus szerint is kinyerhető



Jellemzők :

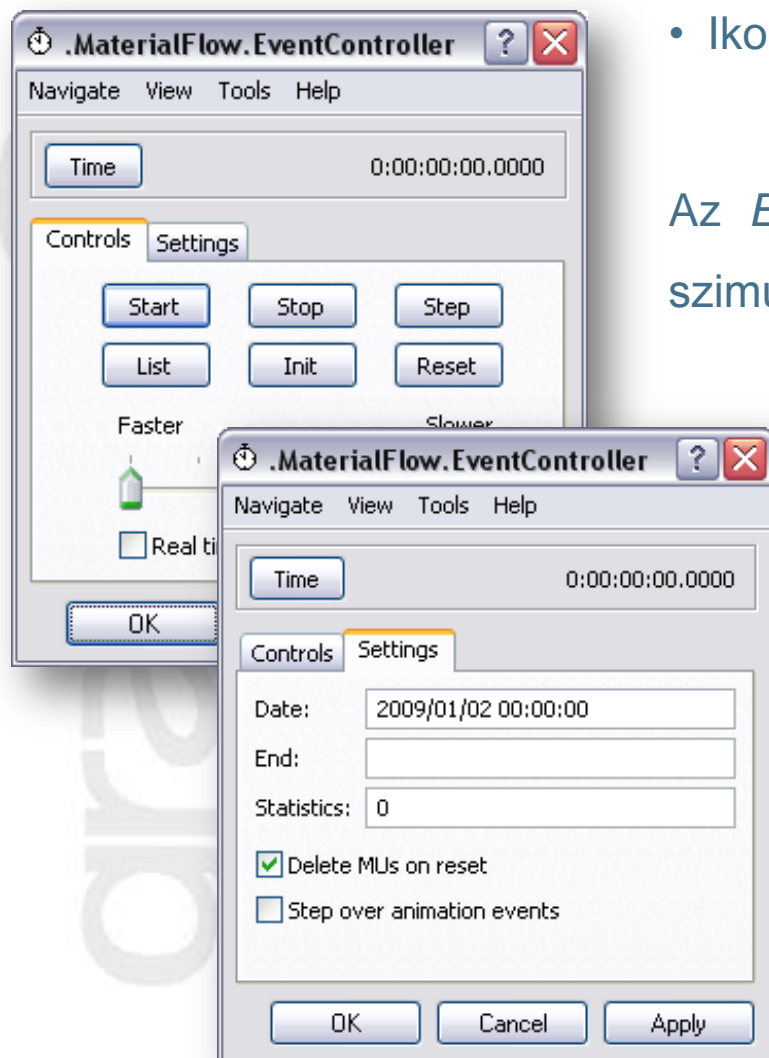
- Ikon:



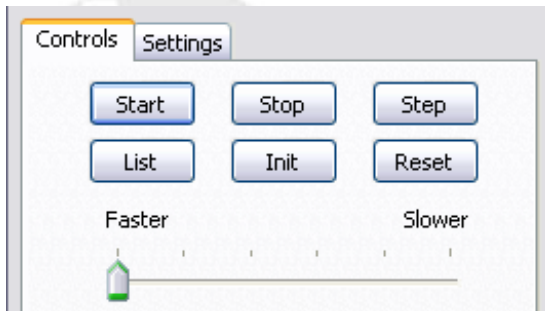
Az *EventController* koordinálja és szinkronizálja a szimuláció futása közben történő eseményeket.

A **Settings** fülön megadható a szimuláció kezdődátuma (**Date**) és az idő, amikor befejeződik (**End**).

Az időpont megadásakor az EventController törli a statisztikákat, és újratekdi a statisztikai adatok gyűjtését. A **Statistics** mezőben megadható, hogy a szimuláció futása során mikortól kezdődjön a statisztikai adatok gyűjtése.



A **Controls** fülön lehet megadni a szimuláció sebességét, és vezérelni a szimulációt.



A szimulációs modell a Reset gombbal állítható alaphelyzetbe.

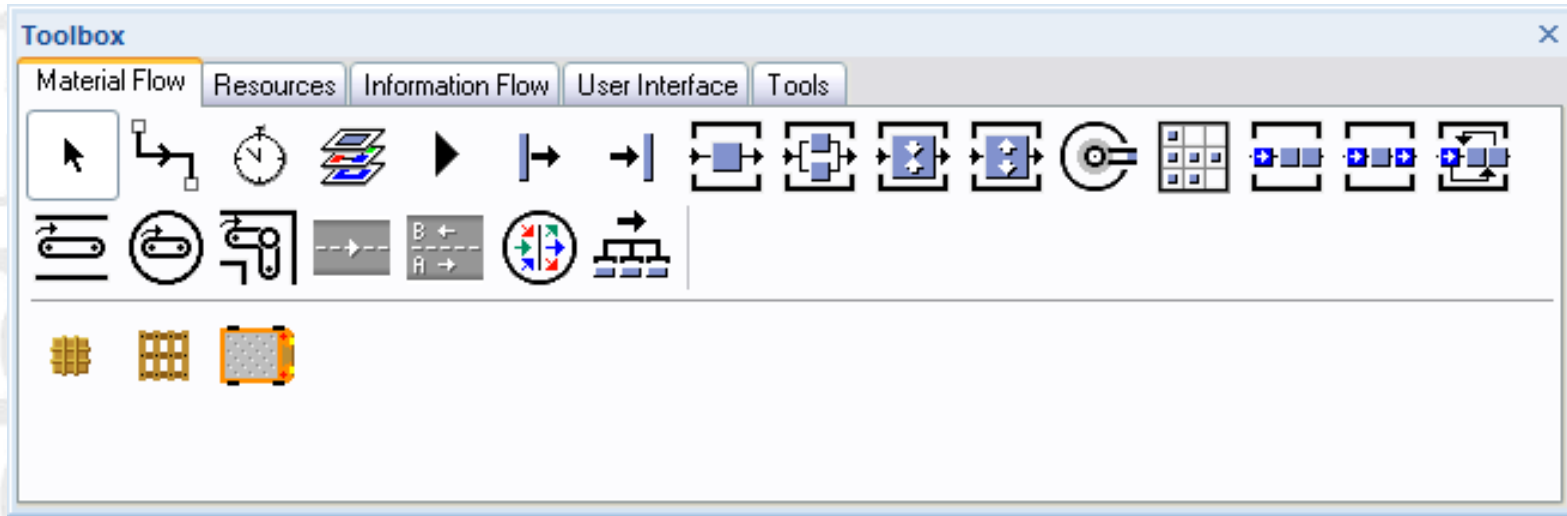
A Plant Simulation meghívja az összes Reset nevű metódust az aktív modellben. Ez törli a fel nem dolgozott eseményeket, visszaállítja a szimuláció idejét 0-ra, alaphelyzetbe állítja a statisztikákat, és eltávolítja a gépek esetleges hibáit.

A szimuláció az Init gombbal inicializálható, ekkor lefut az összes Init nevű metódus.

A metódusok a **Init** gomb lenyomásakor kezdenek el lefutni, és befejeződnek mielőtt a modell inicializálna. A Plant Simulation inicializálja a szimulációs modellt, mielőtt a következő esemény lefutna.

A szimuláció a **Start** gombbal indítható el. Ha ez az első futása a szimulációnak, vagy előtt lenyomta a **Reset** gombot, a Plant Simulation lefuttatja az összes **Init** metódust a modellben. Az aktív események feldolgozása után a szimuláció a **Stop** gombbal állítható meg. Újraindítani a **Start** gombbal lehet.

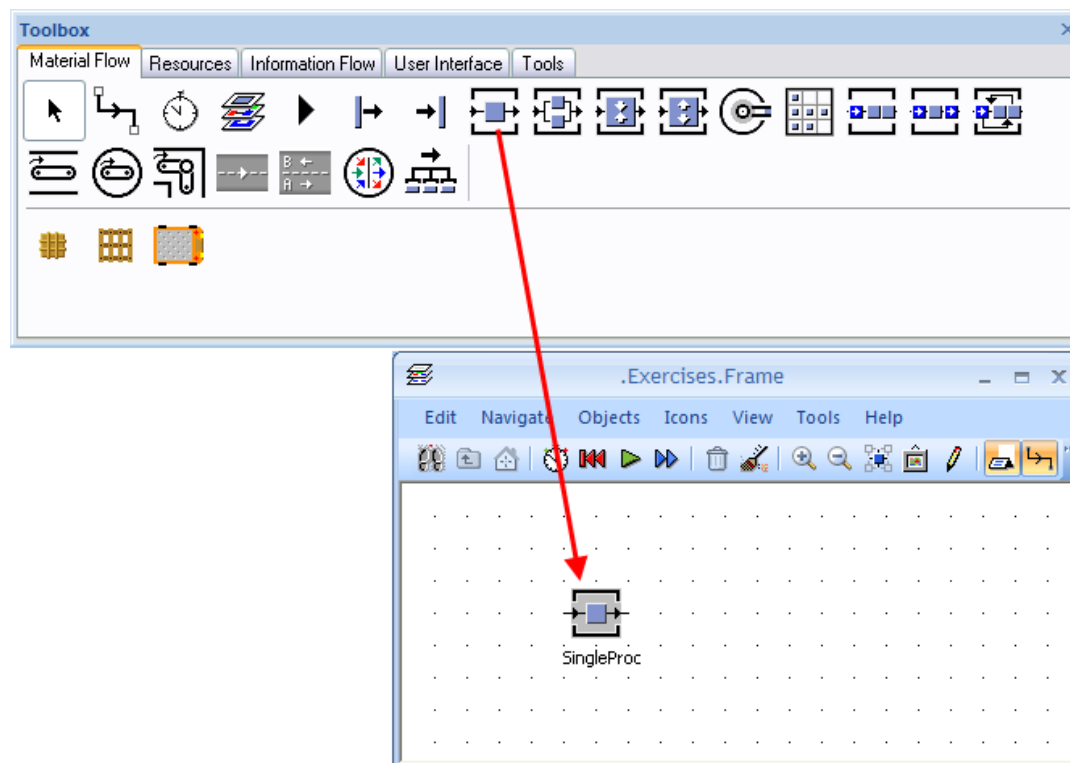
A szimuláció lefuttatható az egyes eseményeken végighaladva lépésről lépésre is (**Step**).



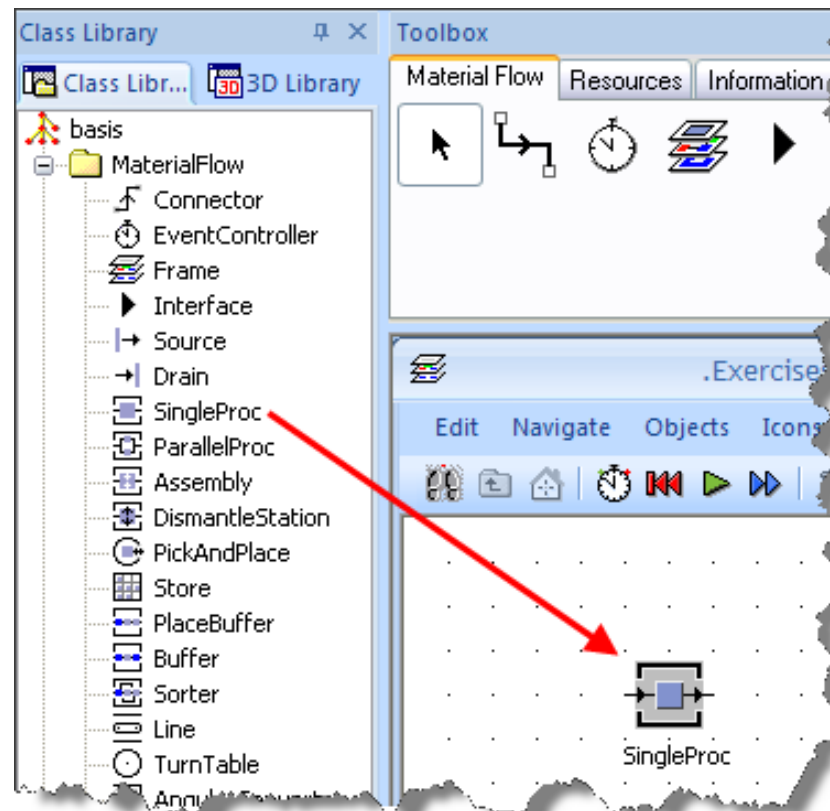
A *Toolbox* azoknak a Plant Simulation eszköztáraknak a tárolója, amelyek a *Class Library* objektumait tartalmazzák.

Objektum beillesztése a palettáról *Frame* - be:

- A megfelelő objektum ikonját tartalmazó fül/eszköztár kiválasztása.
- Ikonjára kattintással a kiválasztása.
- Az egér mozgatás a megfelelő *Frame* - be, és kattintás az objektum elhelyezéséhez.

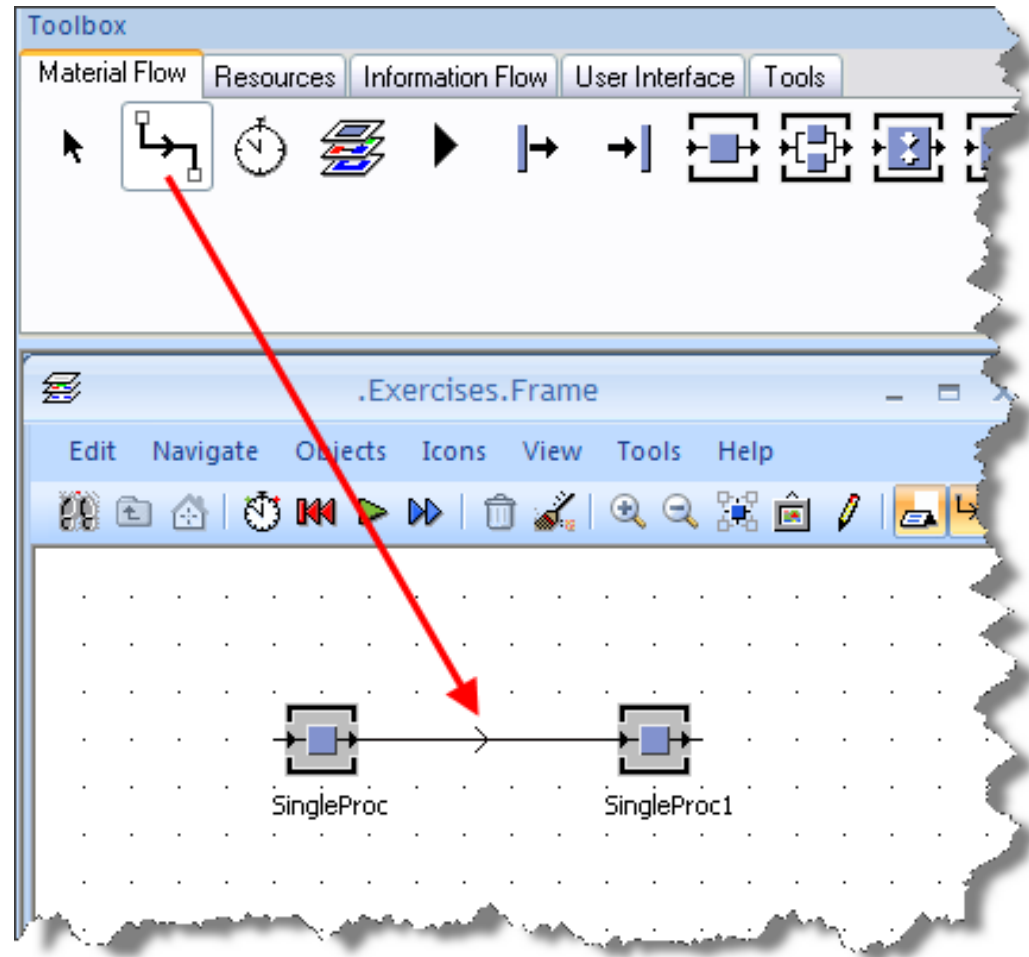


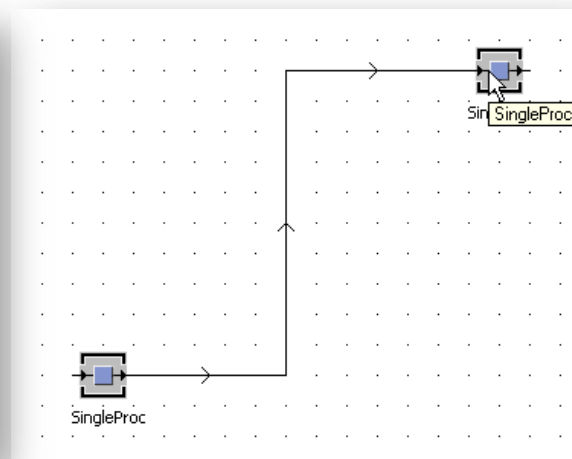
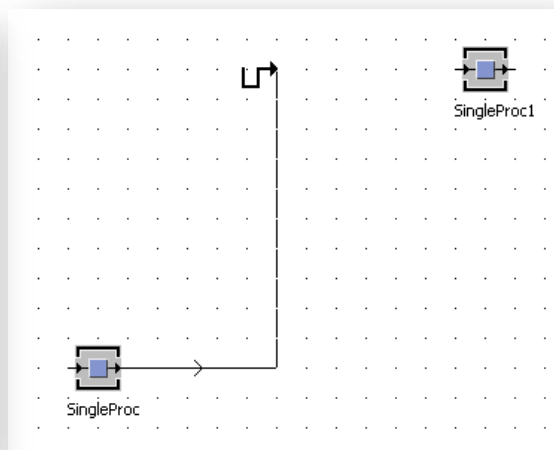
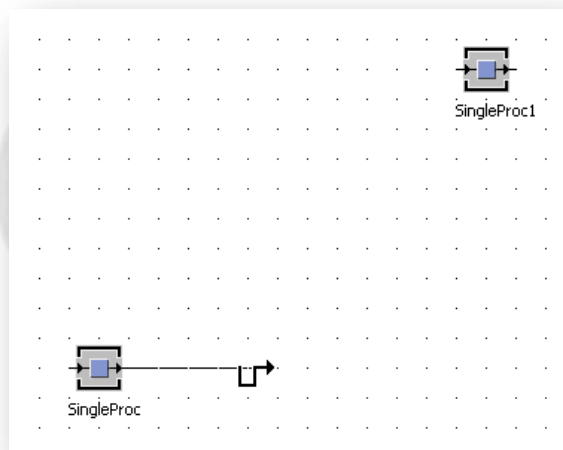
Ha egy objektumból többet kell beilleszteni, akkor az egér kattintás közben nyomva kell tartani a **Ctrl** billentyűt. Ekkor nem kell a palettához visszamenni minden egyes elem beillesztéséhez. Ebből a másolás módból jobb egérkattintással lehet kilépni a *Frame* - en belül.



Az objektumok húzd és ejtsd módszerrel is beilleszthetők a *Frame*-be a *Class Library*-ből áthúzva azokat a *Frame* területére.

A connector (összekötő) objektum különbözik a többi Plant Simulation objektumtól, mert nem használható külön objektumként, csupán más objektumok összekötésére.

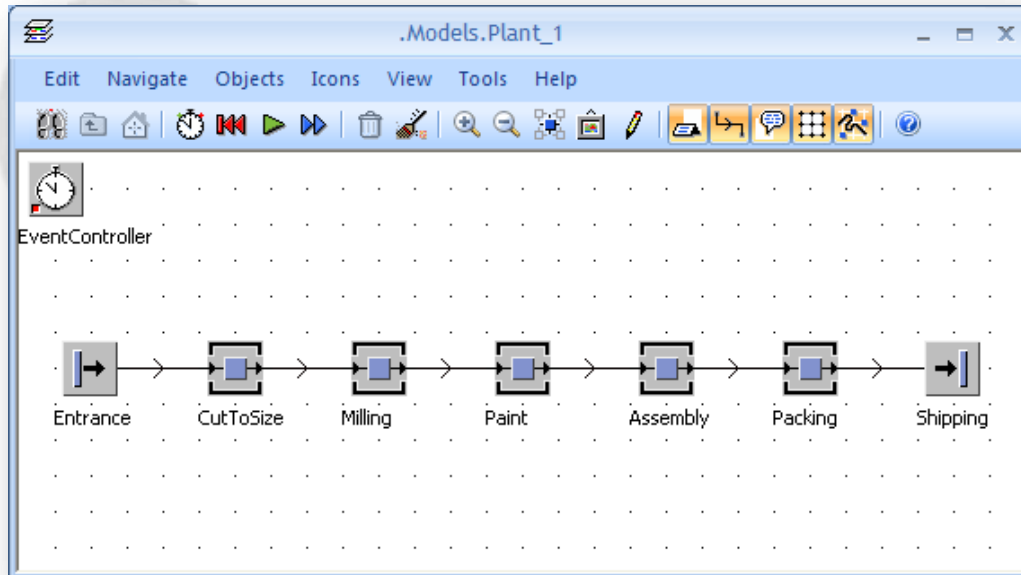




A és B objektum összekötéséhez ki kell választani a *Connector* objektumot a palettán, és kattintani kell először az A, majd a B objektumon.

Nem egyenes összekötés készítéséhez meg kell adni a töréspontokat is. Először kattintani kell az A objektumra, majd a töréspontok helyére és végül a B objektumra. Később a töréspontok helye vonszolással módosítható.

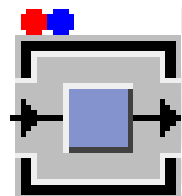
A gyártólétesítmény modellezése



1. Nyissa meg a **Frame** objektumot a **Models** mappában.
2. Nevezze át **Plant_1** névre.
3. Illessze be a szükséges objektumokat és kösse azokat össze.
4. Módosítsa a *Milling* objektum feldolgozási idejét (processing time) 8 percre.
5. Nyissa meg az *EventController* objektumot, és kattintson a **Reset** gombra először, majd a **Start** gombra a szimuláció elindításához.

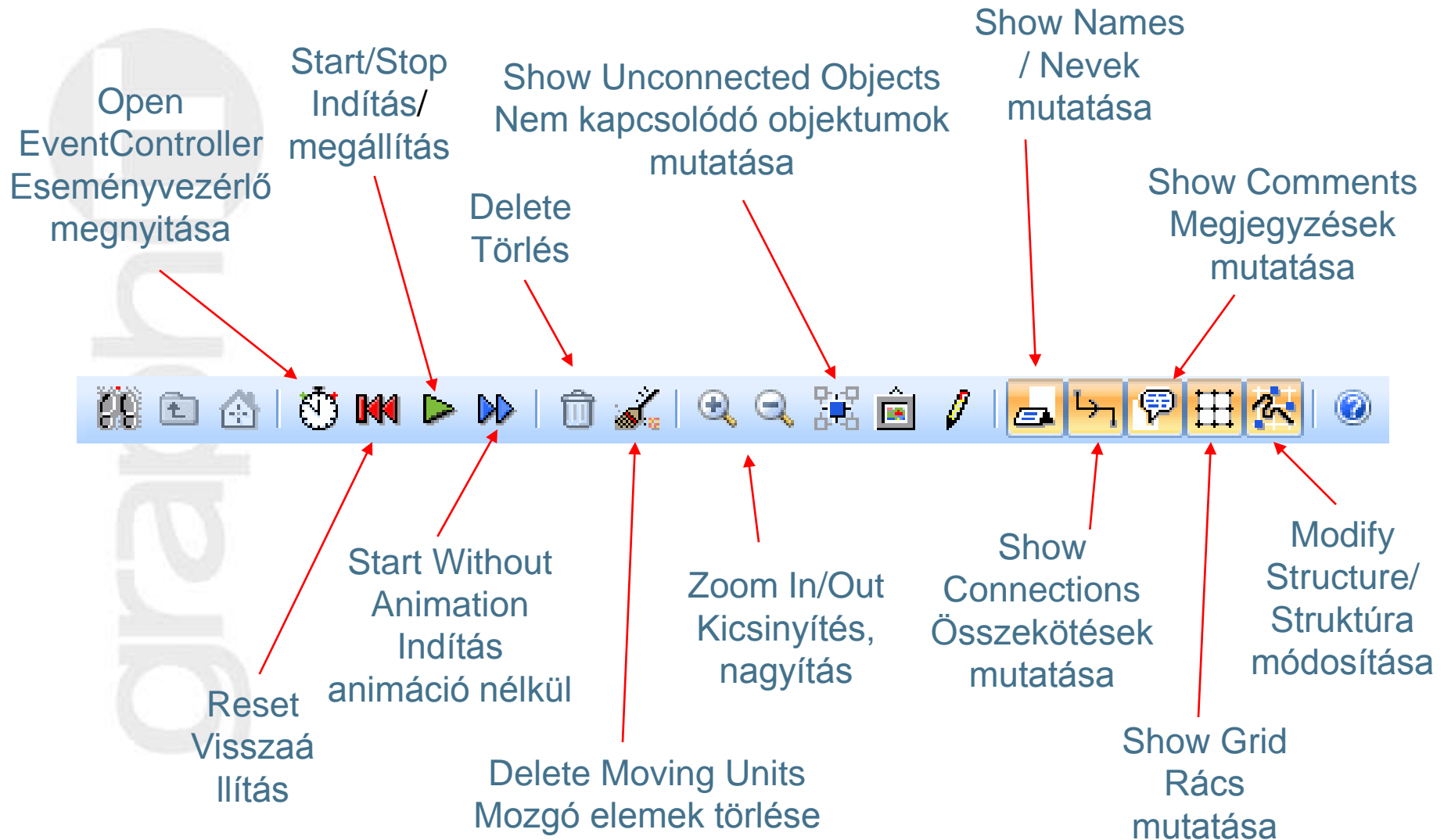
Az anyagáram objektumok LED - del jelzik az ikonjuk felső kerete mentén az adott objektum állapotát. A LED egyszerre többféle állapotot is jelezhet, az állapotjelző ikonok viszont csak egy állapotot tudnak egy időben jelezni..

Az alábbi színek a következő állapotokat jelzik:



- | | |
|--------------------------------|--|
| vörös | az objektum meghiúsult vagy hibás (failed) . |
| kék | az objektum szünetel (paused) . |
| zöld | az objektum működik, dolgozik (working) . |
| sárga | az objektum blokkolt (blocked) . |
| barna | az objektum beállítás alatt (setting-up) (setUp). |
| vil. kék
(no_entry). | az objektum helyreállítás alatt (recovering) |

Az alapértelmezett állapot ikon nem jelenít meg semmilyen színes jelet.



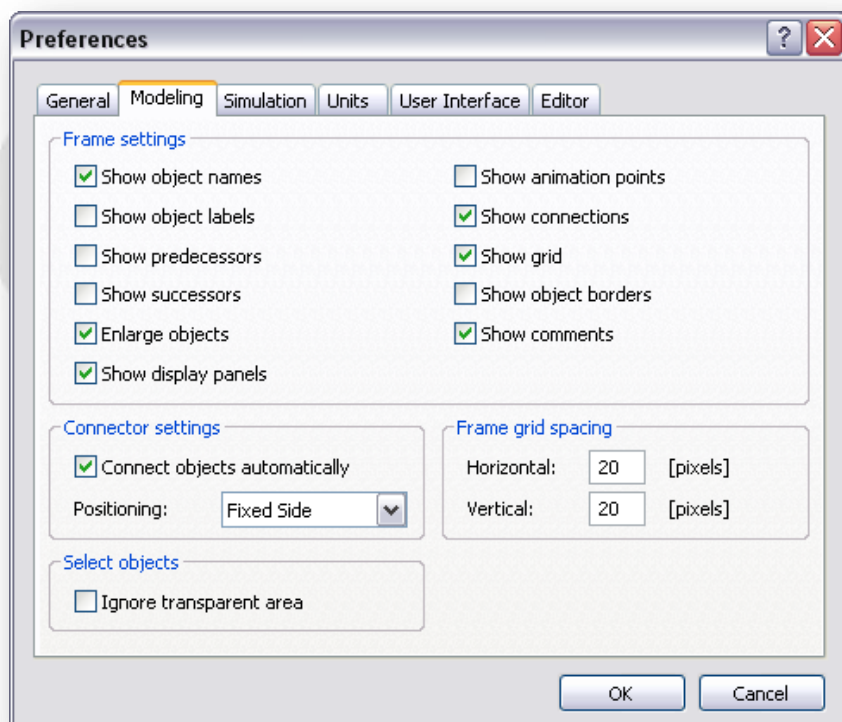
A Frame beállításai a menüben:

Tools > Options > Modeling

(Eszközök > Beállítások > Modellezés)

Ezek a beállítások elérhetők a Frame ablakában a ***View > Options*** (Nézet > Beállítások) menüpont alatt is.

Javasolt a ***Delete objects > Confirm*** (Objektumok törlése > Nyugtázás) opció bekapcsolása, mivel az objektumok törlése nem vonható vissza.



Általánosságban az új beállítások a Plant Simulation újraindítása után érvényesülnek. Bizonyos beállítások azonban, mint például a mértékegységek módosítása azonnal érvényesülnek.

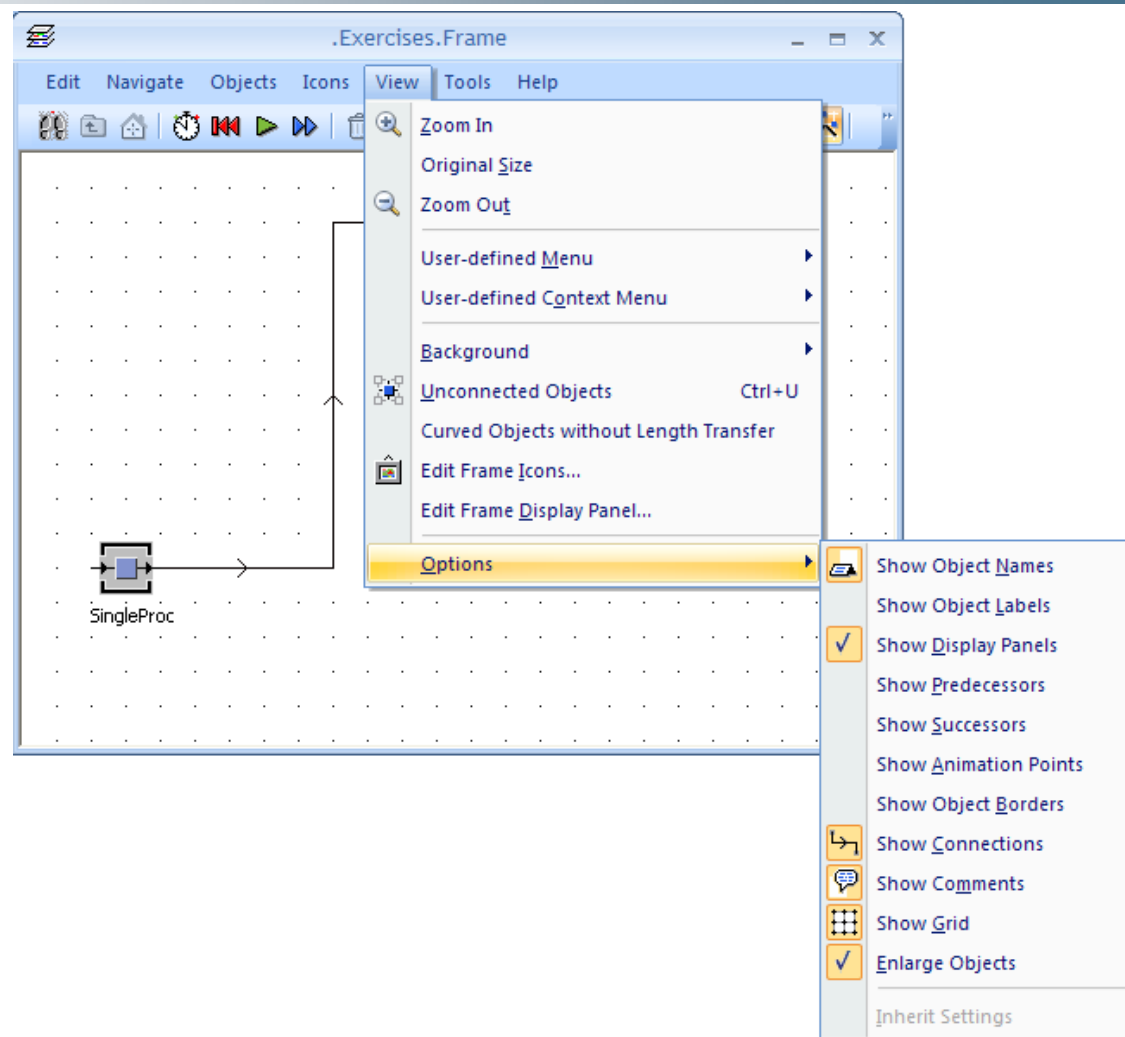
Minden egyes Frame - re külön megadhatók beállítások a menüben:

View > Options (Nézet > Beállítások)

A globális beállításokat pipa jelöli a menüpont bal oldalán.

Emellett a **View** (Nézet) menü parancsokat tartalmaz:

Zooming (Nagyításhoz), és felhasználói menük és helyi menük definiálására a Frame - hez



User Menu (felhasználói menü) és **Shortcut Menu** (helyi menü) és a nem összekötött objektumok megjelenítésére: **Showing unconnected objects**.

Megadható az ikonok helye, a helyzete, a mérete és szerkeszthetők (**Edit icons**) az ikonok.

Az **Icons** (ikonok) menü csak akkor aktív, ha ki van választva objektum a Frame-ben.

Mirror Vertically – Tükrözés függőlegesen

Mirror Horizontally – Tükrözés vízszintesen

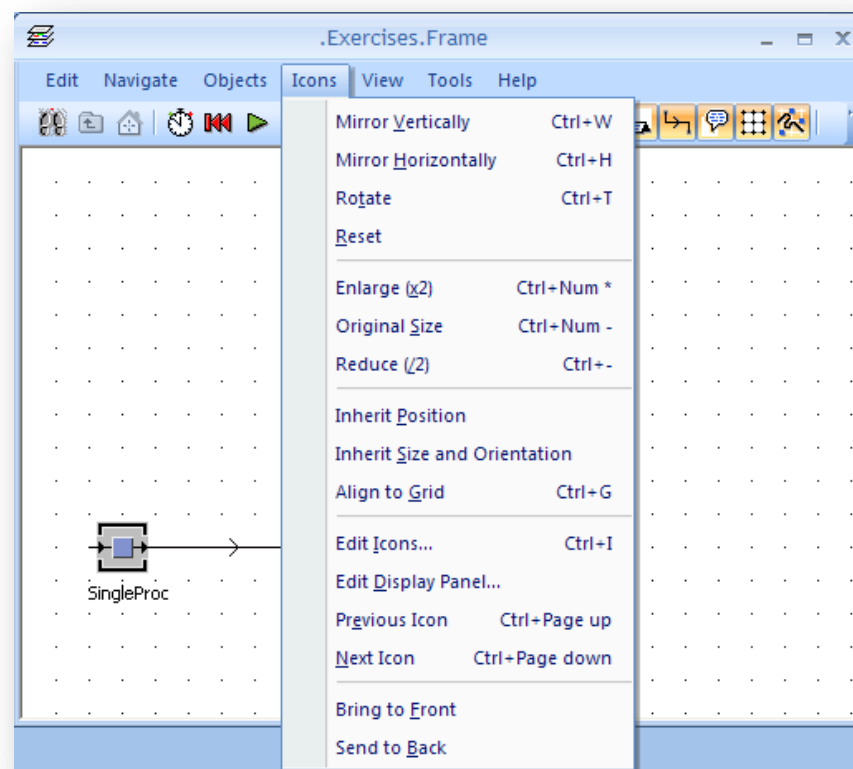
Rotate – Forgatás

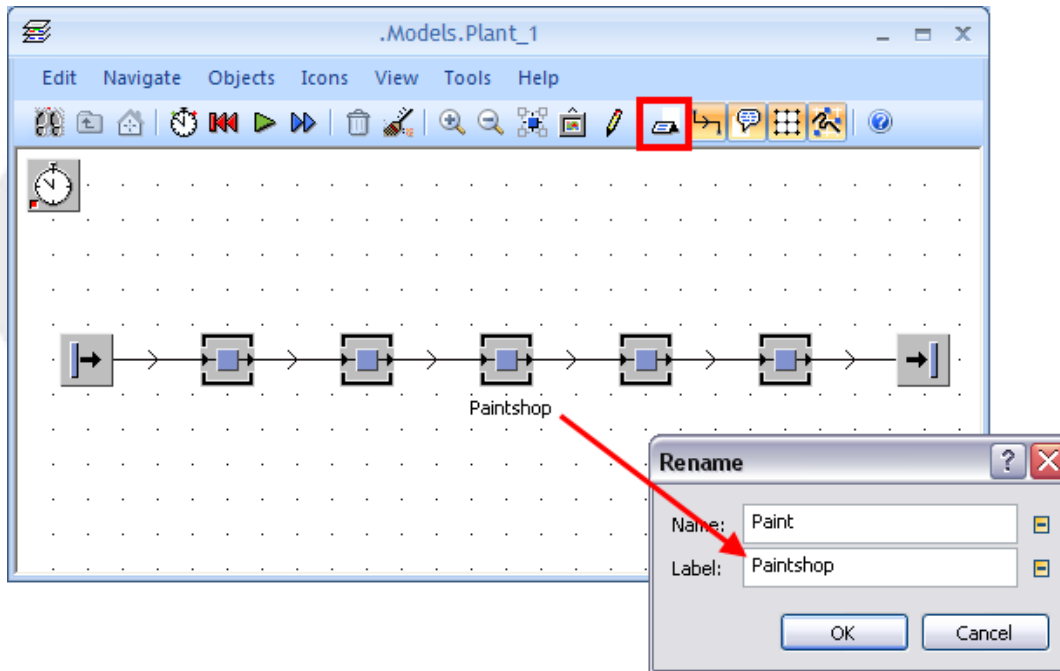
Reset – Visszaállítás

Enlarge (x2) – Duplára növelés

Original size – Eredeti méret

Reduce (/2) – Felére csökkentés





Az objektumokra a nevük (**Name**) mellett egy címke (**Label**) is megadható. A neveknek eleget kell tenni a megnevezési szabályoknak, míg a címkék tetszőleges szöveget tartalmazhatnak.

Az objektumok nevének és címkéjének kiírása be, illetve kikapcsolható a menüből:

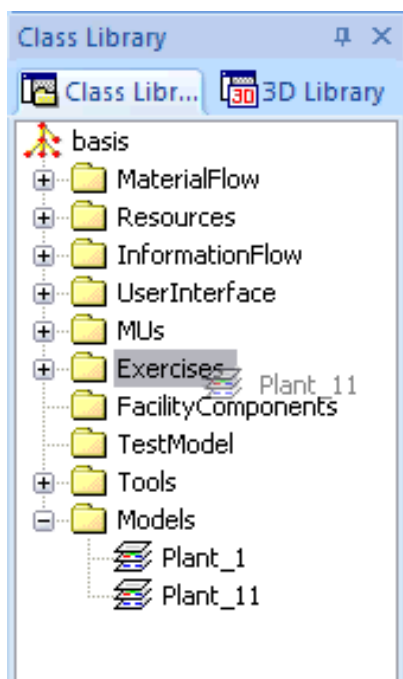
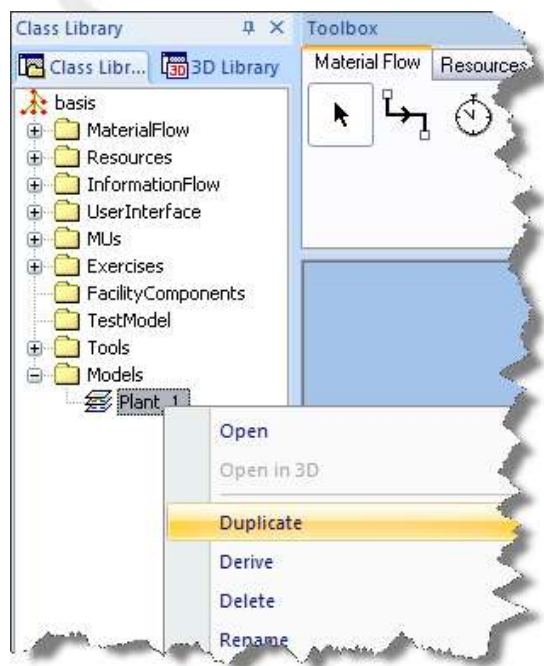
View > Options > Show Object Names (Nézet > Beállítások > Objektum nevek mutatása)

és/vagy

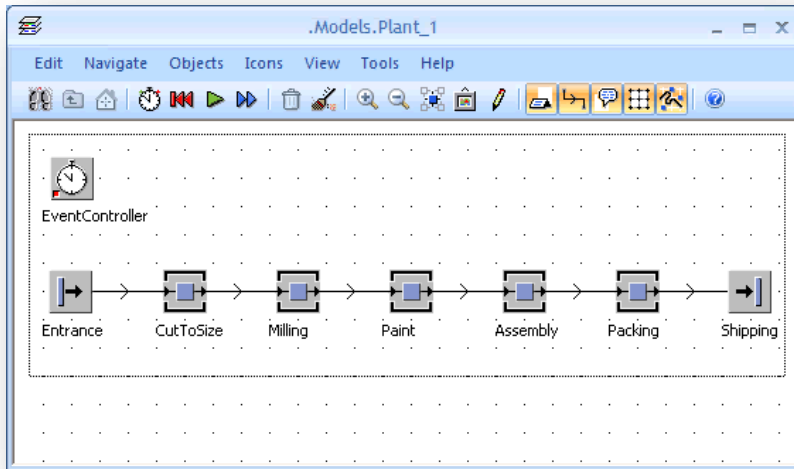
Show Object Labels (Objektum címkék mutatása)

Általában a leckék az előző modellezési lecke eredményéből indulnak ki, például ez a lecke az előzőekben megadott Frame - et használja a kezdéshez.

Hogy leckénként szét tudjuk válogatni a lépéseket, először egy másolatot készítünk az eddig elkészítettekből a Class Library > helyi menü > **Duplicate** (Másolat készítése) parancsával. Ez után a Frame átnevezhető (helyi menü vagy F2 billentyű) és átmozgatható másik mappába a **Shift** billentyűt lenyomva tartva az egérrel vonszolás során.

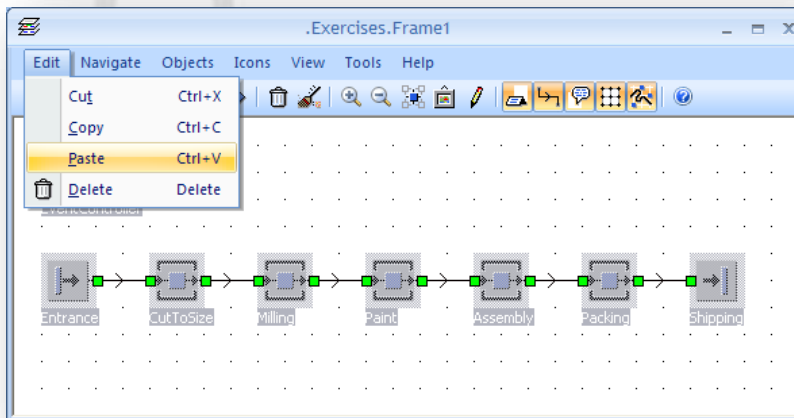


Ez a technika a valós életben is jól használható a projekteknél.



Másolja Vágólapra a Frame egész tartalmát és illessze be ugyanabba és egy másik Frame-be.

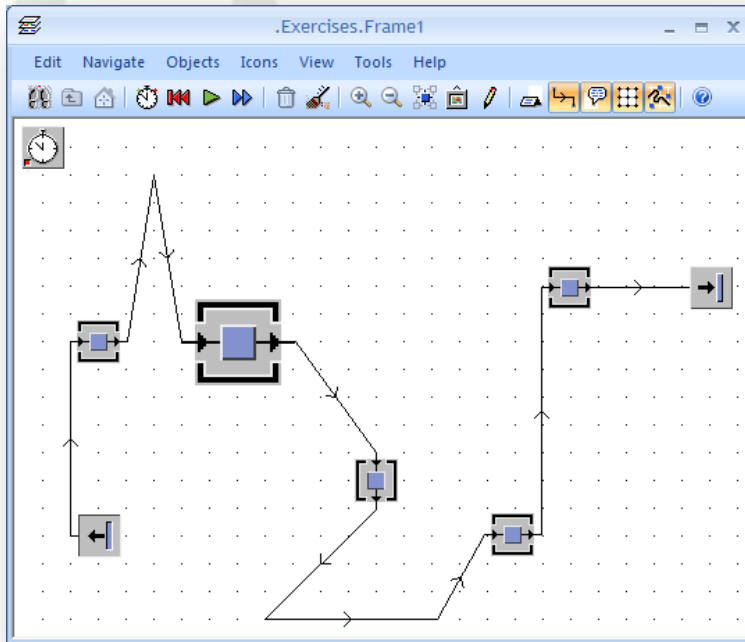
Kattintson az egér bal gombjával és téglalap kijelöléssel válasszon ki tetszőleges objektumokat.



Másolja ezeket az objektumokat a vágólapra az **Edit > Copy** (Szerkesztés > Másolás) (vagy Ctrl+C) paranccsal.

Illessze be a vágólap tartalmát az **Edit > Paste** (Szerkesztés > Beillesztés) (vagy Ctrl+V) paranccsal tetszőleges helyre és tetszőleges *Frame*-be.

Menüparancsok használata objektumok megjelenítésére a Frame - en belül



Másolja át a Plant_1 Frame tartalmát egy új Frame-be az Exercises (Frame_1) mappába.

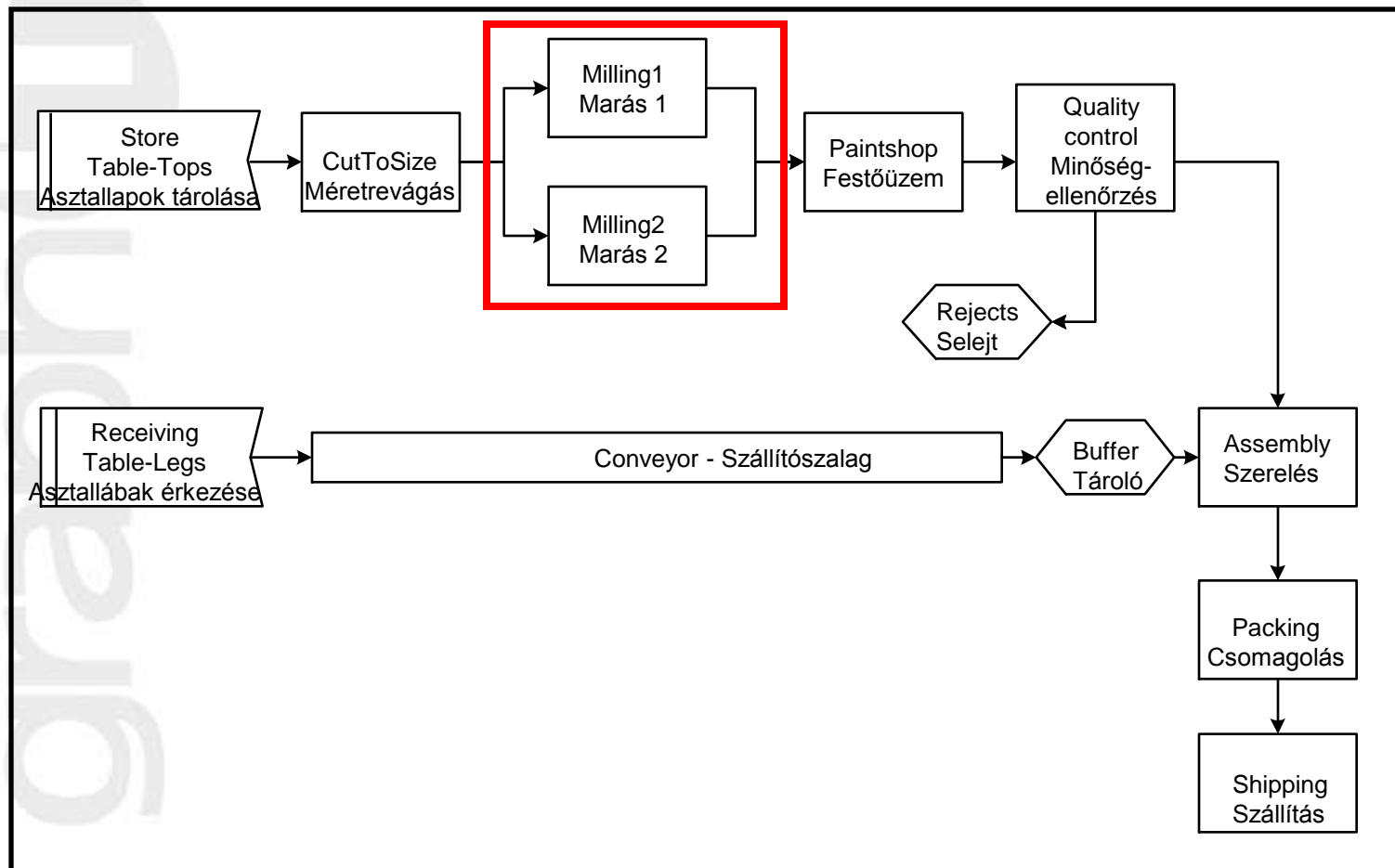
Gyakorolja az objektumok megjelenítésének különböző módjait, a **nagyítást, forgatást és mozgatást, objektumok összekötését** egyenes és nem egyenes összekötő elemekkel, és **objektumnevek és címkék megadását** és megjelenítését.

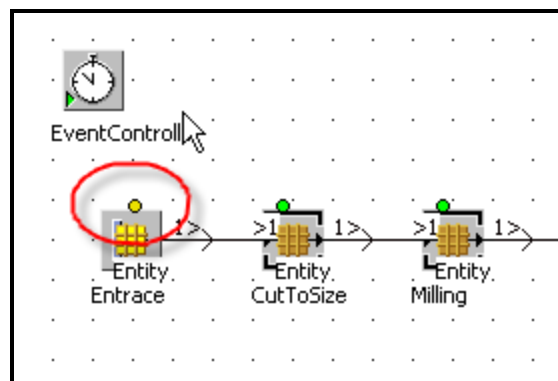
Keresse meg, hogy melyik nyomógomb tartozik a *Frame* eszköztáron az *EventController* objektumhoz.



7. fejezet

Anyagáram megadása, statisztikák,
grafikonok

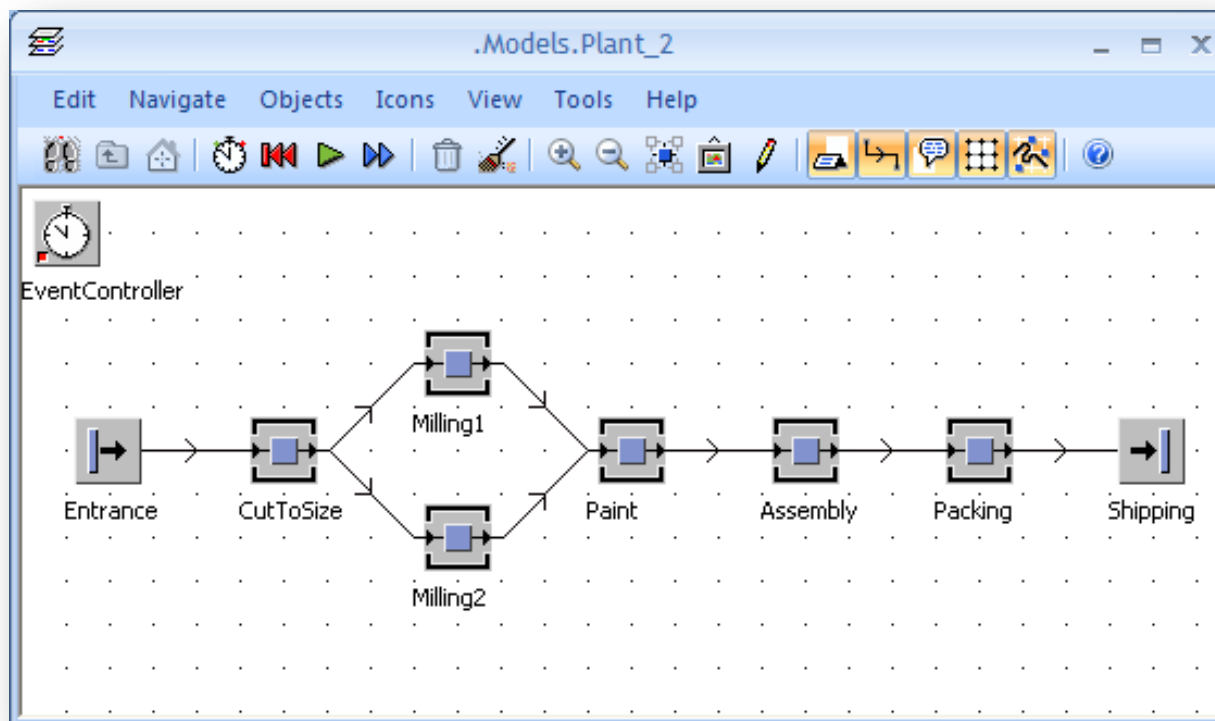




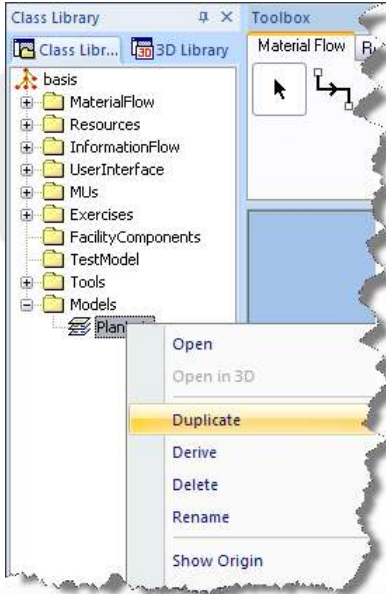
A 8 perces feldolgozási idővel rendelkező marógép az üzemünk legszűkebb keresztmetszete.

Megoldásként hozzáadunk még egy marógépet a sorhoz, és azt az elsővel párhuzamosan helyezzük el. Ennek a gépnek a feldolgozási ideje 3 perc lesz.

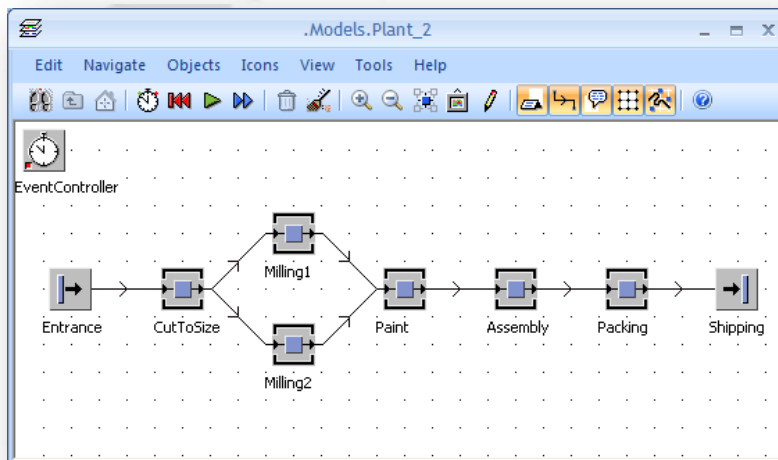
Illesszünk be egy második marógépet (*SingleProc*) a modellbe, és kössük össze azt a többi géppel *Connector* -okkal.



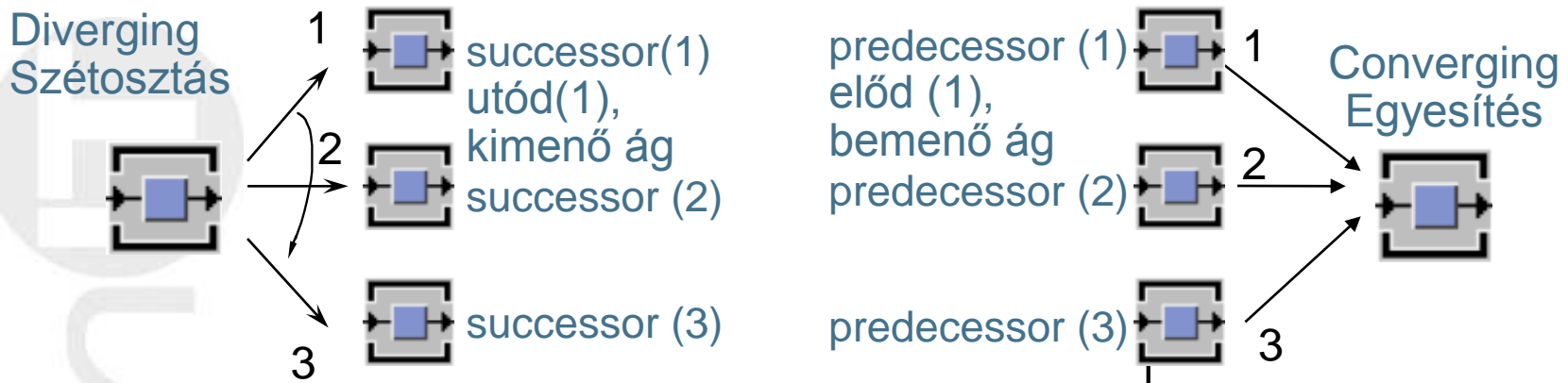
Anyagáram szétosztás és egyesítés



1. Másolja a Plant_1 objektumot és nevezze el **Plant_2**-nek.
2. Készítsen egy második marógépet (*SingleProc*) aminek a neve legyen **Milling2**.
3. Válassza ki a **View > Options > Show Successors** (Nézet > Beállítások > Utódok mutatása) menüpontot a *Frame* menüjében.

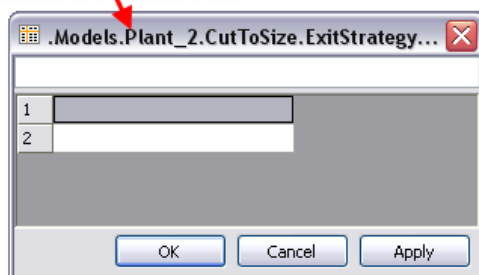
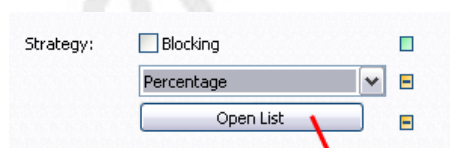
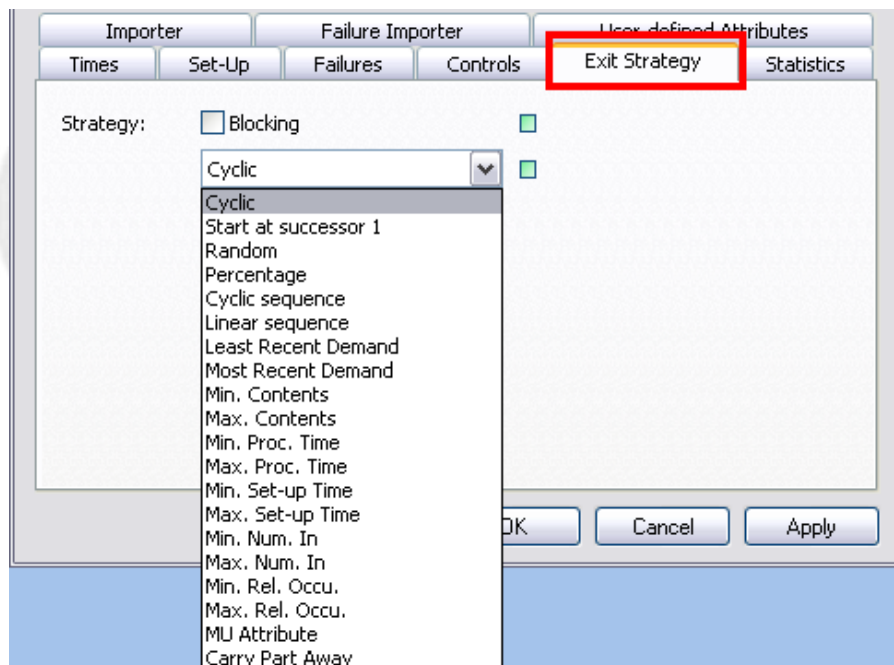


4. Indítsa el a szimulációt és **csökkentse a sebességét**.
5. Figyelje meg, hogyan mozognak az MU-k az egyik állomásról a másikkra.



Az alapértelmezett stratégia anyagáram **szétosztásakor** a nem-blokkolás: a Plant Simulation az MU-kat a **successor** -ok **sorrendjében** küldi tovább. Ha egy successor foglalt az MU vár addig amíg a successor szabaddá válik az MU fogadására.

Amikor egyesül az anyagáram, a Plant Simulation az MU-kat az érkezési sorrendben küldi tovább, amelyik először érkezik az először távozik. Ha az ág foglalt, akkor az MU bekerül a *Forward Blocking* listájába az adott ágnak.




A kilépési stratégia az anyagáram objektum **Exit Strategy** (Kilépési stratégia) fülén választható ki listából.

A kilépési stratégia határozza meg, hogy milyen módon küldi tovább az objektum az MU-kat.

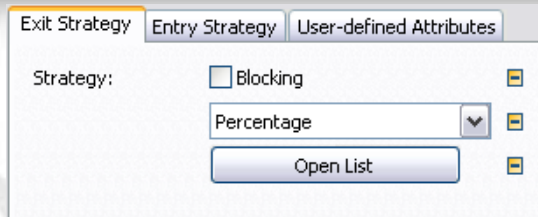
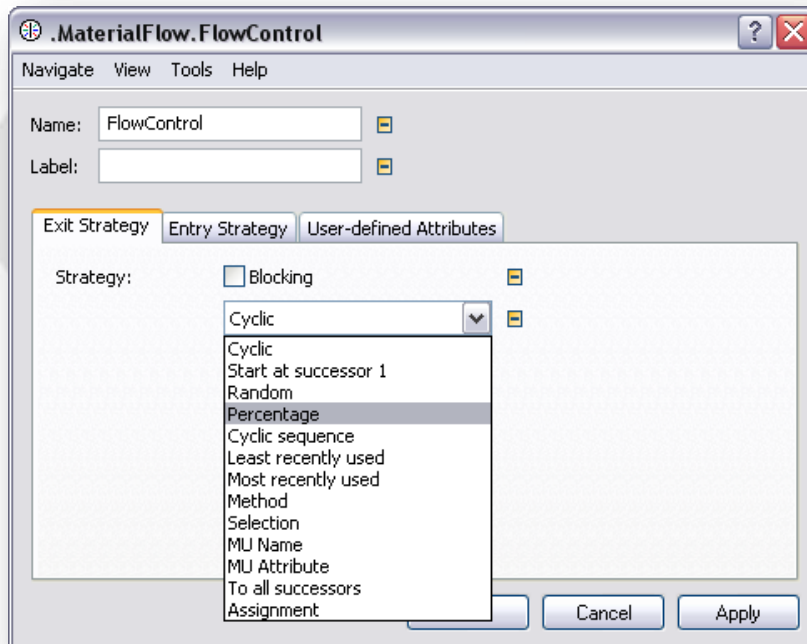
A **Cyclic** (periodikus) stratégia az alapértelmezett. A **Blocking** (blokkolt), esetben csak akkor megy tovább az MU, ha a következő objektum fogadó ága készen áll az MU fogadására.

Ha a **Blocking** opció ki van kapcsolva, akkor bármely fogadó ág szabaddá válásakor tovább mozog az MU.

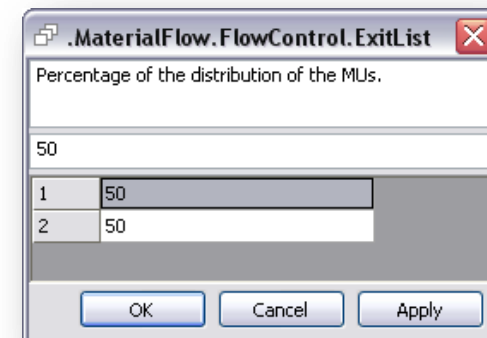
Jellemzők :

- Ikon: 
- Kapacitás: 0
- Anyagáram objektum

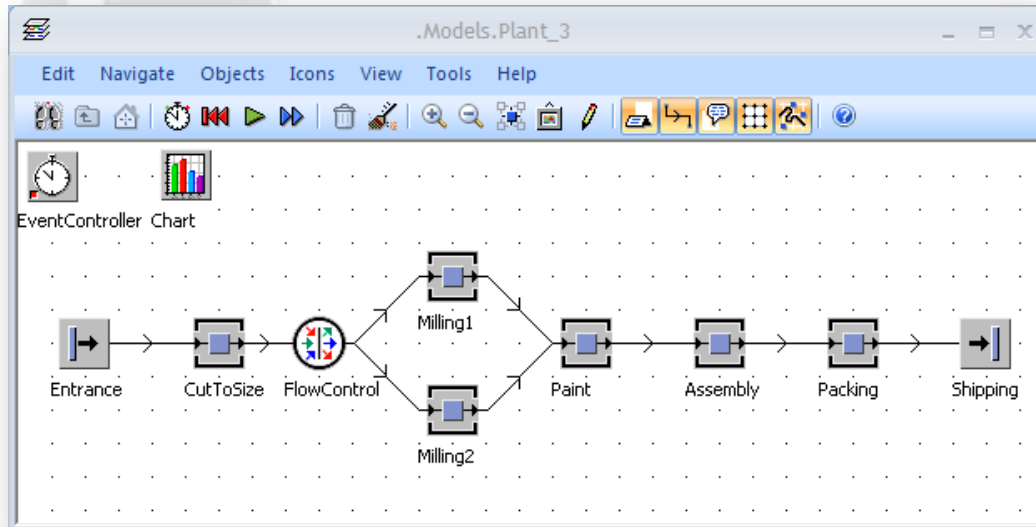
A *FlowControl* szabályozza az anyagáramlást, meg lehet adni egy kilépési stratégiát az áramlás szétosztására, vagy egy belépési stratégiát az anyagáramlás egyesítésére.



Példa: Percentage strategy (százalékos stratégia)



FlowControl, Exit Strategy (kilépési stratégia), Percentage Strategy (százalékos stratégia)



1. Másolja a Plant_2 objektumot és nevezze át **Plant_3** névre.
2. Illesszen be egy **FlowControl** objektumot a *CutToSize* és a két marógép állomás (*Milling1* és *Milling2*) közé.

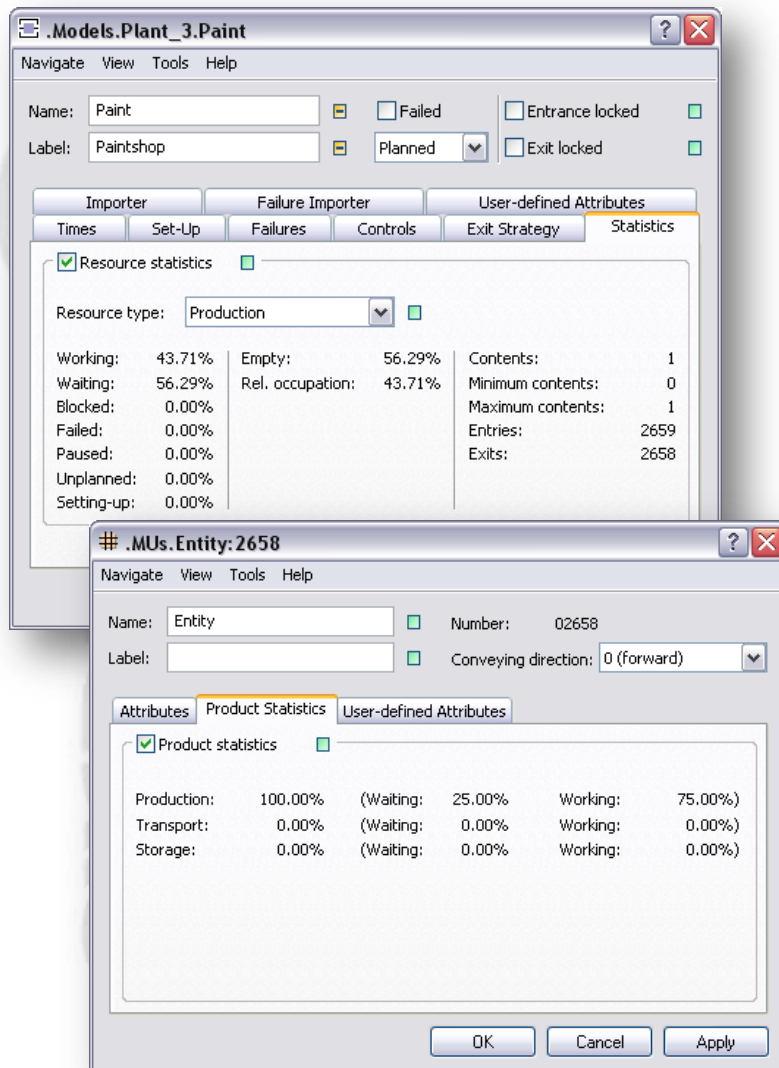
A *FlowControl* kontrollálja az anyagáramlást az ez egyik állomásról a másikkra.

3. Válassza ki a **Percentage** stratégiát és kapcsolja be a **Blocking** opciót:
Milling1 kapja 27%-át, *Milling2* a 73%-át a mozgatott MU -knak.

Statisztikák különböző típusai:

A típusuktól függően az anyagáram objektumok és az MU-k különböző típusú statisztikai értékeket gyűjtenek össze:

- **Resource statistics:** *A következő típusú anyagáram objektumok: Production, Container és Transporter.*
- **Product statistics:** Minden mozgó objektum.
- **Driving statistics:** *Transporter.*
- **Complete statistics:** Minden anyagáram objektum és minden mozgó objektum beírja a statisztikai értékeit a teljes statisztika táblájába.



Minden objektum gyűjt magáról statisztikai értékeket. Ez kikapcsolható a **Statistics** (Statisztika) fül *Resource statistics / Product statistics* (Erőforrás / termék statisztika) opciójával.

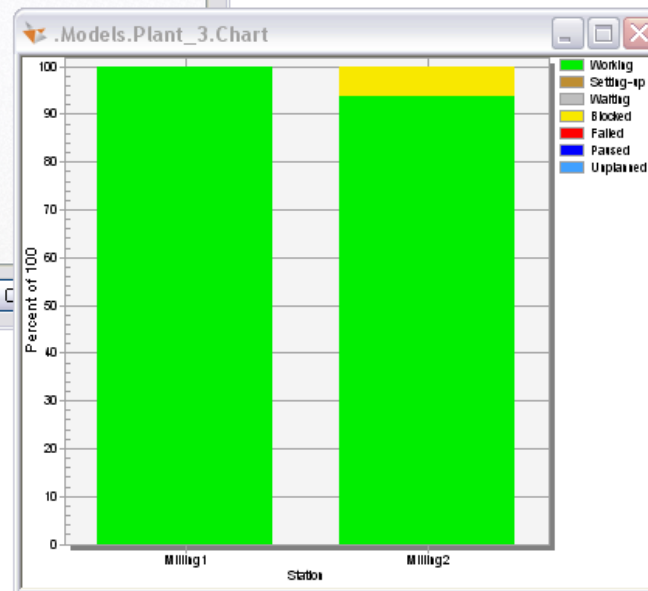
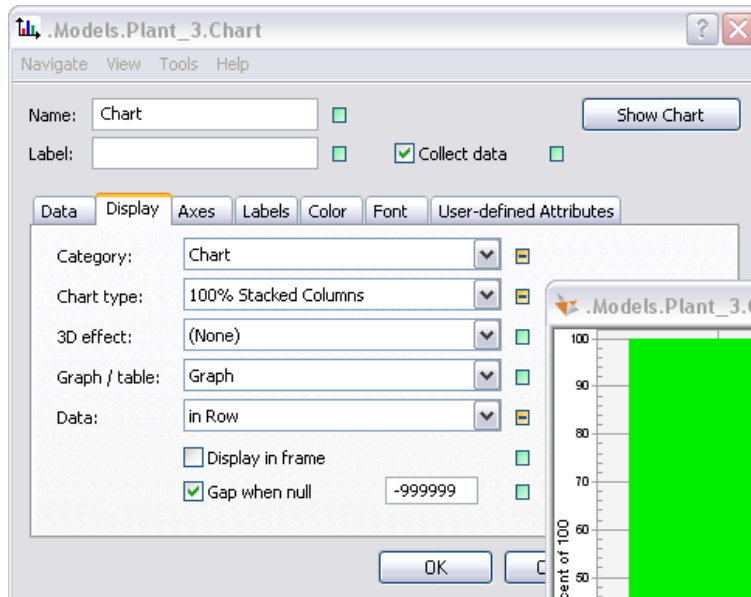
A Plant Simulation a legfontosabb statisztikai értékeket a **Statistics** vagy **Product statistics** fülön mutatja.

Minden anyagáram objektum és minden mozgó objektum beírja a statisztikai értékeit a **Resource Statistics Table**, a **Product Statistics Table** vagy **Complete Statistics Table** táblákba.

Az *EventController* **Reset** gombjára kattinva a Plant Simulation az összes statisztikai értéket nullára állítja vissza.

Rákattintva a **View, Statistics Report** menüpontra gombra, a Plant Simulation megnyit egy táblázatot, amely tartalmazza az összes értéket, amely összegyűlt a szimuláció futása során. A különböző értékek részletes leírását a Súgó tartalmazza a **Statistics Table** téma alatt.

A különálló statisztikai értékek a statisztikai metódusok révén érhetők el, az ilyen függvények neve „stat...” kifejezéssel kezdődik.



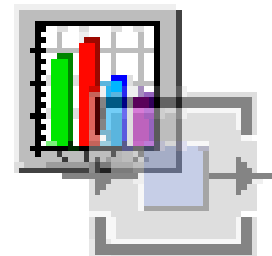
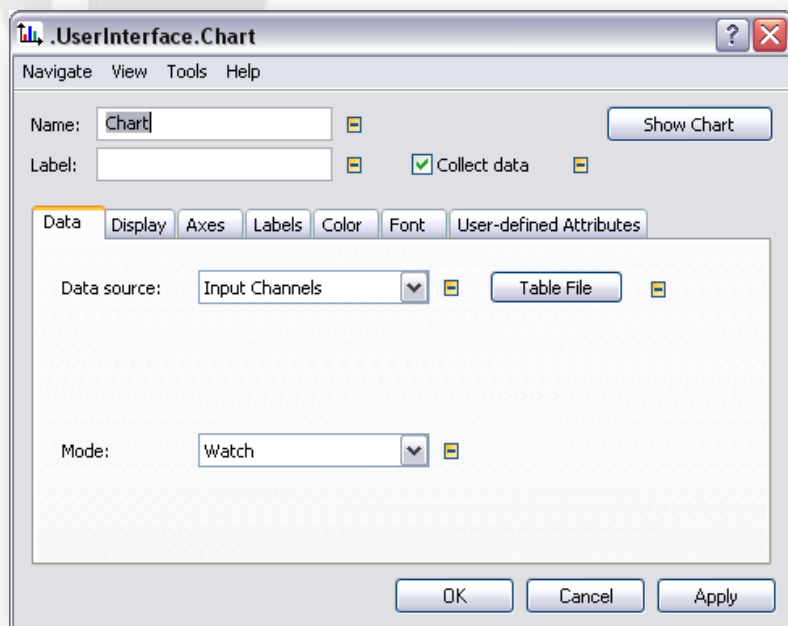
Jellemzők :

- Ikon: 
- Kapacitás: 0
- Felhasználói felület objektum

A *Chart* (grafikon) statisztikai adatok megjelenítésére szolgál, és lehetővé teszi az aktuális értékek és a szimulációs eredmények gyors **elemzését**.

Az állapotok (**working**, **waiting**, **blocked**, **failed**, és **paused**) mellett egyéb statisztikai értékeket is össze tud gyűjteni.

A *Chart* objektumot be kell szúrni a *Frame*-be. Aztán az anyagáram objektumot, amelynek a statisztikáját meg szeretnénk jeleníteni, rá kell dobni a *Chart* objektumra.



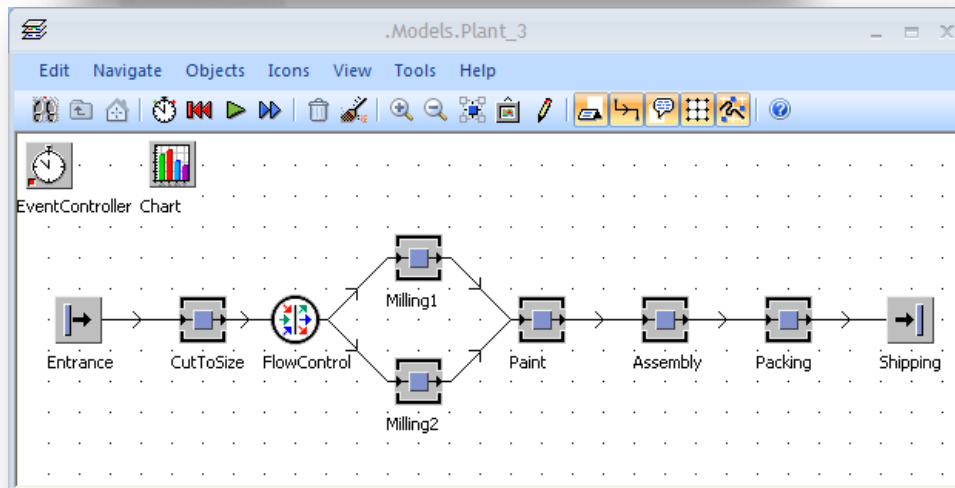
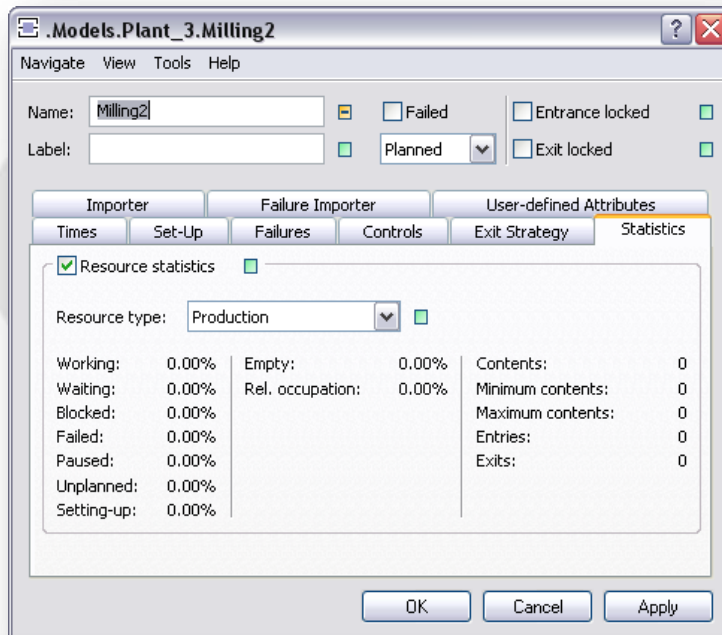
A rádobott objektum típusától függően a grafikon állapotjelzőként (**state chart**) működik, mutatva a megfelelő anyagáram objektum állapotát.

Ha az első rádobott objektum egy buffer (puffer) objektum volt, akkor a grafika **histogram** (oszlopdiaagramm) lesz.

A grafikon **plotterként** is használható.

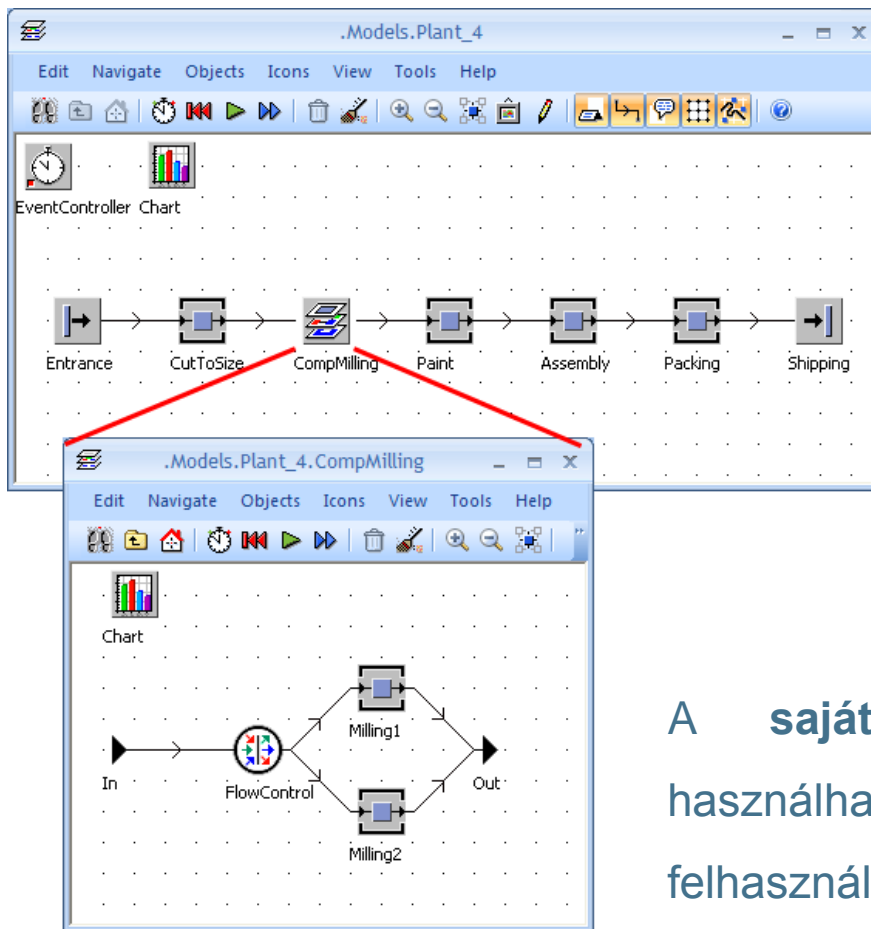
Statisztikák és grafikon

1. Aktivizálja a statisztikák gyűjtését a két marógépre.
2. Illesszen be egy *Chart* objektumot a **Plant_3 Frame**-be.
3. Dobja rá a *Milling1* és *Milling2* objektumokat a *Chart* objektumra.
4. Nézze meg a statisztikát a szimuláció futása során.



8. fejezet

A modell hierarchikus szerkezete

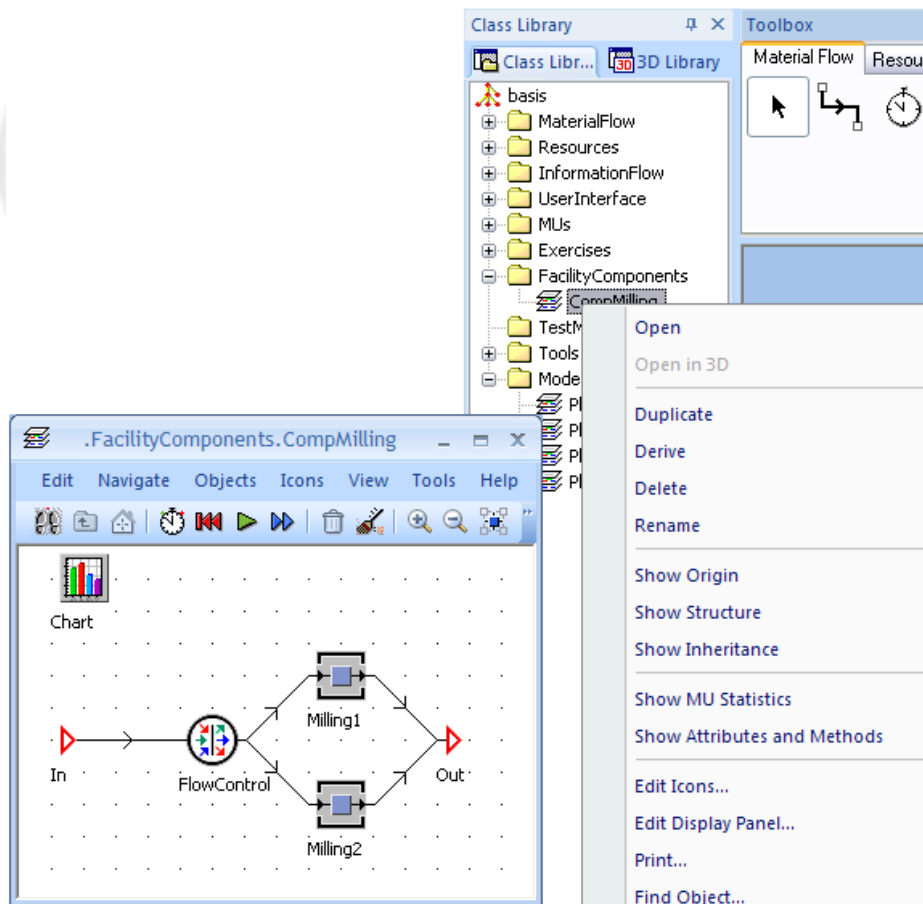


A hierarchia az a koncepció, ahogy a *Frame*-ek egymásba ágyazhatók. Ezzel a módszerrel különálló komponensek modellezhetők és tesztelhetők.

Egy *Frame* kombinálhat egy vagy több komponenst, és a komponensek a modellbe tetszőleges számban beilleszthetők.

A **saját fejlesztésű objektumok** ugyanúgy használhatók, mint a beépített objektumok, és felhasználhatók más modellekben is.

A hierarchikus modellezés könnyen áttekinthetővé is teszi a modellt.



A komponenst a *Class Library*-n belül egy új *Frame*-ként modellezzük, és elhelyezzük rá az objektumokat, amelyekre szükség lesz ehhez a modulhoz.

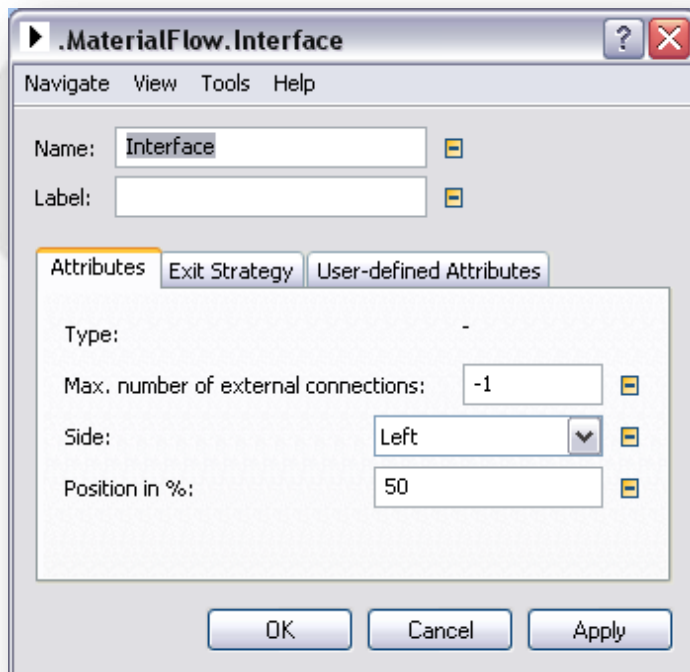
Ez után a komponenst beillesztjük a szimulációs modellbe és úgy használjuk, mint bármely beépített objektumot, akár másik projektekben is.

A komponens elmenthető objektumként, ami más projektekbe betölthető.

Az **Interface** (interfész) objektum adja meg, hogy a *Frame* melyik oldalán lép be az anyagáram és melyiken lép ki.

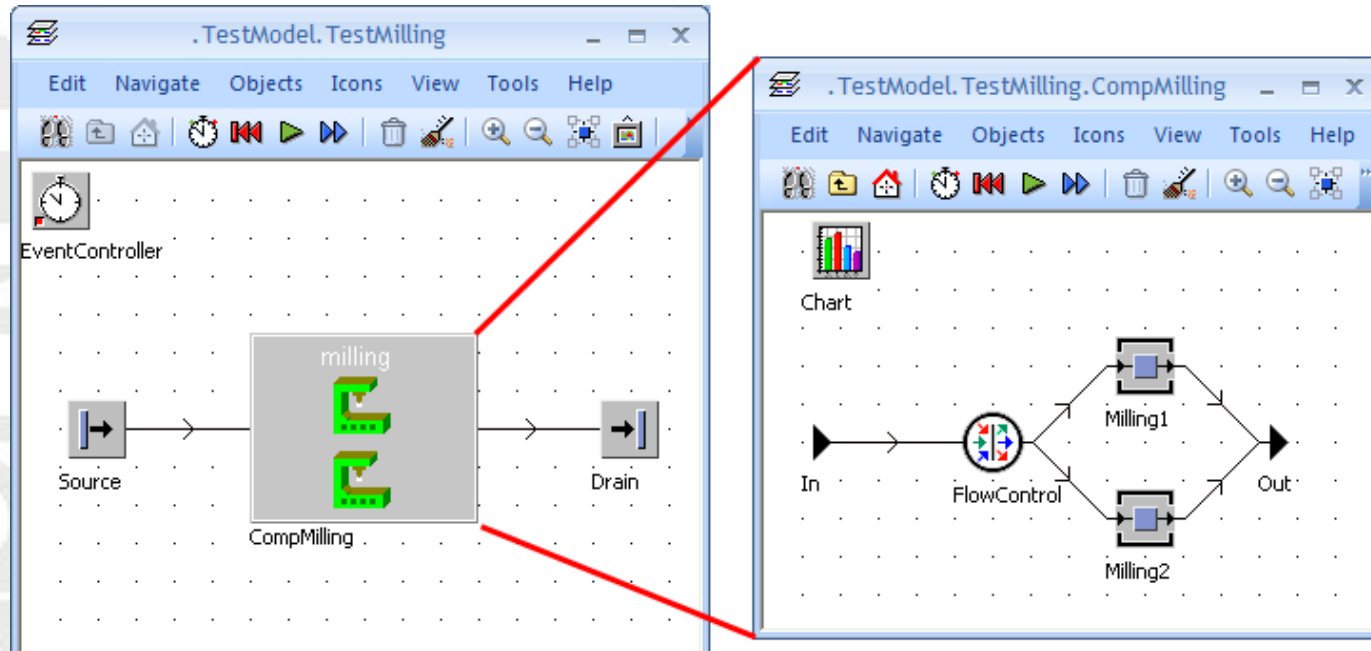
Jellemzők :

- Ikon: ►
- Kapacitás: 0
- Anyagáram objektum



Az **Interface** modellezi a kapcsolatot *Frame* és anyagáram objektumok között. Megadja azt is, hogy melyik oldalon csatlakozik a *Connector* objektum hozzá.

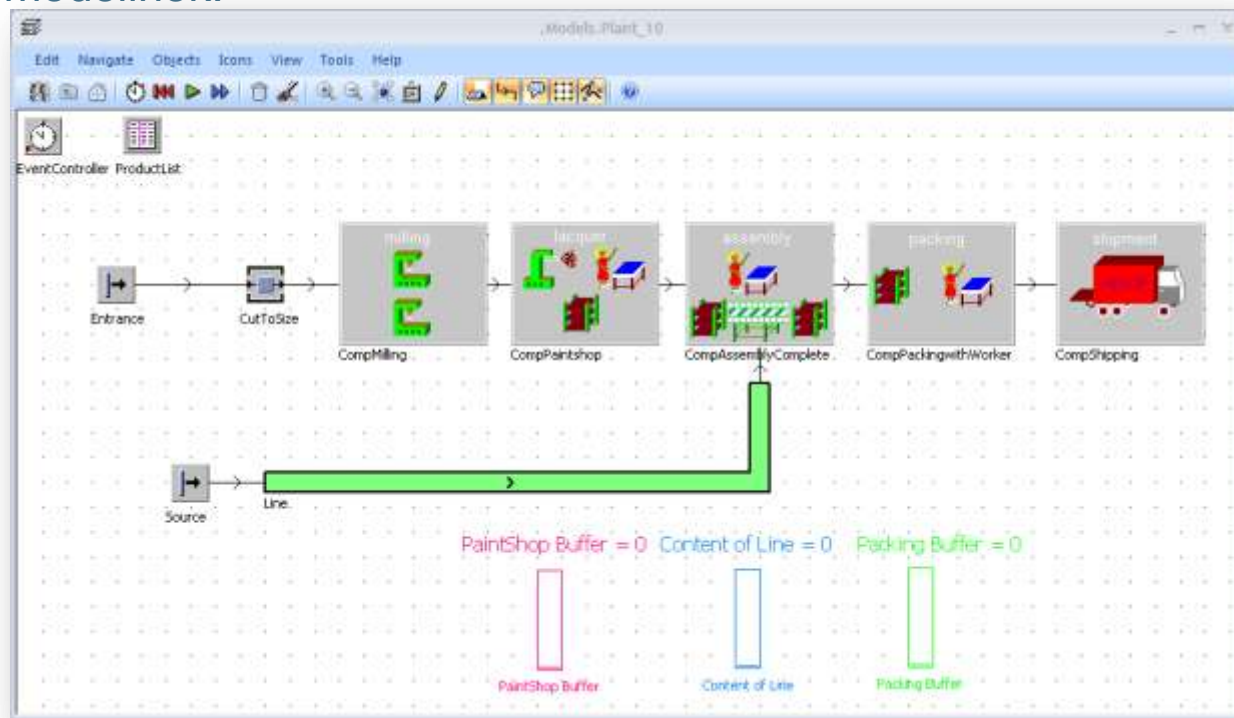
Oldal (Side) kiválasztás: **Left (bal)**, **Right (jobb)**, **Top (felül)** vagy **Bottom (alul)** ahol az ikonon a Plant Simulation megjeleníti a belépést és a kilépést a *Frame* objektumon. Megjegyzendő, hogy a Plant Simulation a **Side** (oldal) beállítást csak a megjelenítés céljára használja.



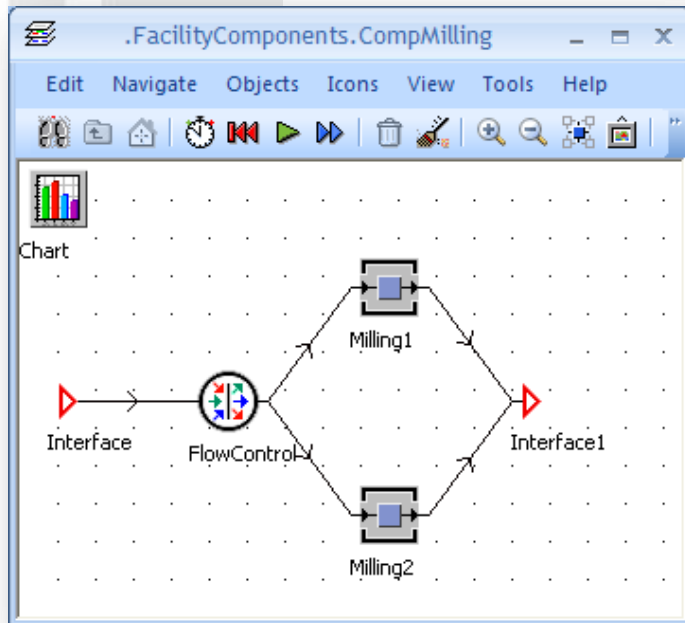
A **komponens** (felhasználói által definiált objektum) funkcióját le kell tesztelni, mielőtt a modellbe illesztésre kerülne.

Felhasználói komponensek segítségével modellezhetők a szimuláció objektumai. A komponensekhez ikonok készíthetők, és az elkészült komponensek beilleszthetők a modellbe, mint bármely anyagáram objektum. Az elkészült komponensek felhasználhatók más szimulációs modellekben is.

A komponensekkel történő modellezés könnyebben áttekinthető szerkezetet ad a szimulációs modellnek.

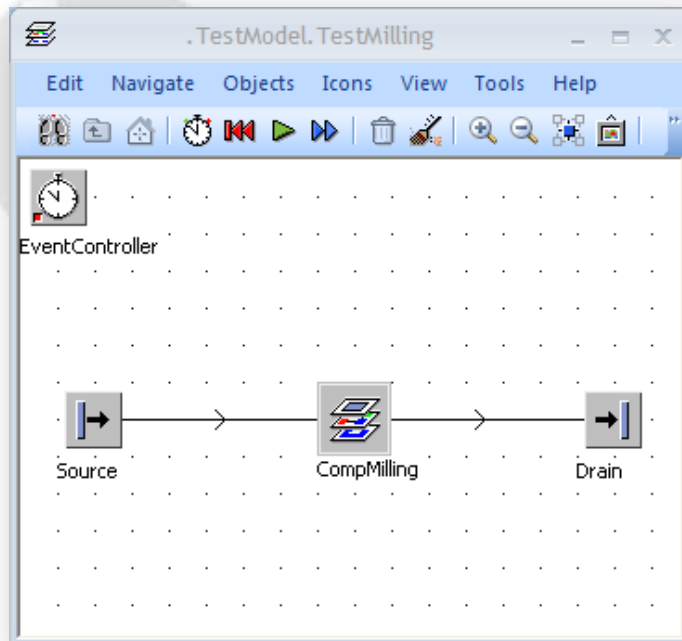


Modul a szimulációs modellben



1. Készítsen egy új Frame-et a **FacilityComponents** mappában, és legyen a neve: **CompMilling**.
2. Másolja át a *FlowControl*, *Milling1* és *Milling2*, és *Chart* objektumokat a **Plant_3** objektumról a *CompMilling* Frame-re.
3. Illesszen be két **Interface** (interfész) objektumot. Válassza ki a megfelelő Side oldalt, ahol a *Connector* objektum kapcsolódik.

A modell/teszt környezet hierarchiája

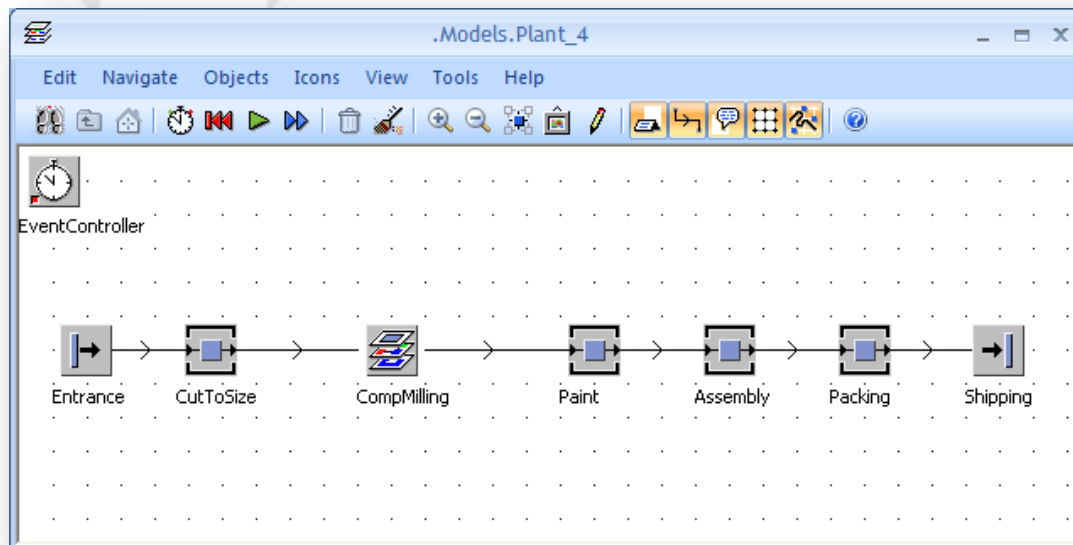


1. Készítsen egy új *Frame* -et a **TestModel** mappában, és legyen a neve: **TestMilling**.
2. Illesszen be egy *Source*, egy *Drain*, egy *EventController* objektumot, dobja be a *CompMilling Frame* -et a *FacilityComponents* mappából a *Source* és a *Drain* közé. Kapcsolja össze őket *Connector* objektumokkal.
3. Tesztelje a **funkcionalitást**. Nyissa meg a *CompMilling Frame* -et és nézze meg mi történik benne.

Ne feledje, hogy a *CompMilling* komponensen változtatásokat csak a **Class Library** - ben végezzen a *FacilityComponents* mappában.

A modell hierarchiája

Miután bebizonyosodott, hogy a *CompMilling Frame* a várt módon működik, be lehet illeszteni a modellbe.



1. Készítsen másolatot a Plant_3-ból és legyen a neve **Plant_4**.
2. Törölje a *FlowControl* és mindkét marógép komponenst, és illessze be a *CompMilling Frame* komponenst a *FacilityComponents* mappából.
3. Kösse össze a *Frame* -et a többi komponenssel, és futtassa a szimulációt.

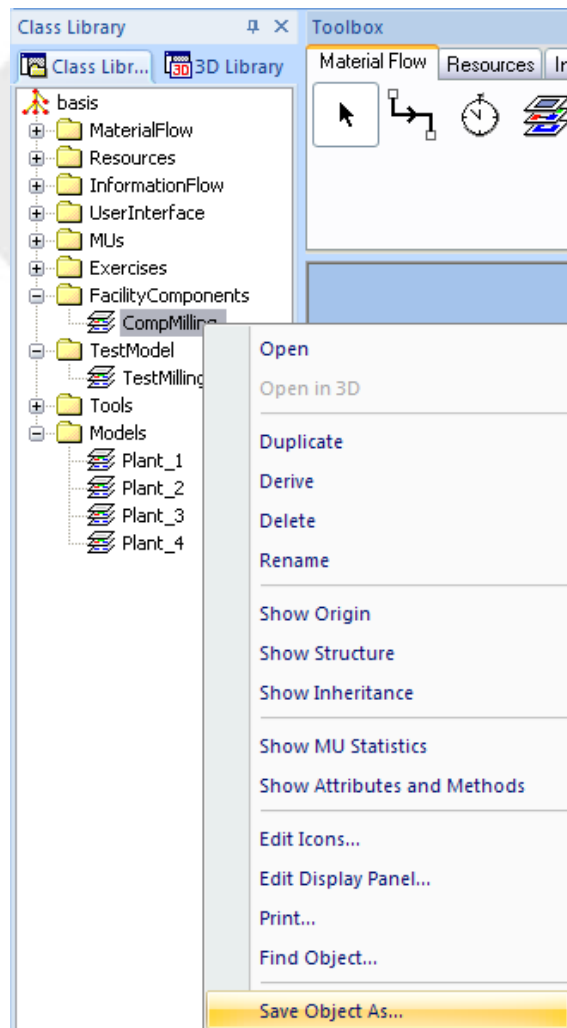
A komponensek vagy mappák objektum fájlként elmentetők *.obj kiterjesztéssel:

Jobb klikk a komponensen > Save Object As
(Objektum mentése másként)

Más projekt osztály könyvtárába (*Class Library*) a komponens beszűrhető:

Jobb klikk > Load Object (Objektum betöltése)

Ha a névvel megegyező nevű objektum már létezik a *Class Library*-ban, akkor megadható, hogy felül kívánja - e írni a meglévő objektumot az újjal, vagy az új objektum más néven kerüljön be a modellbe.

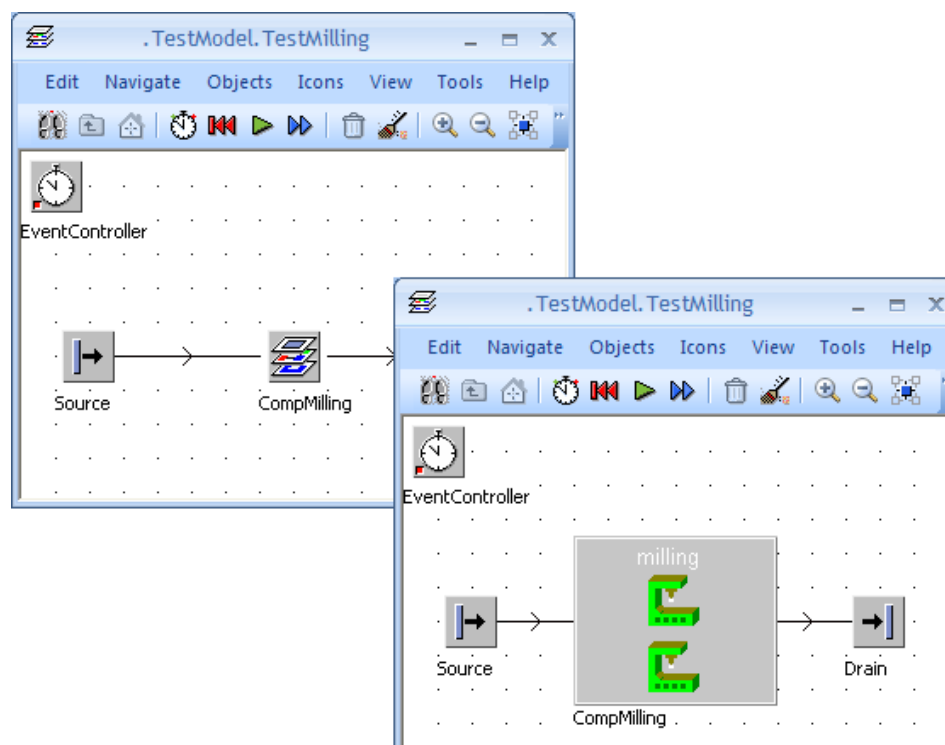




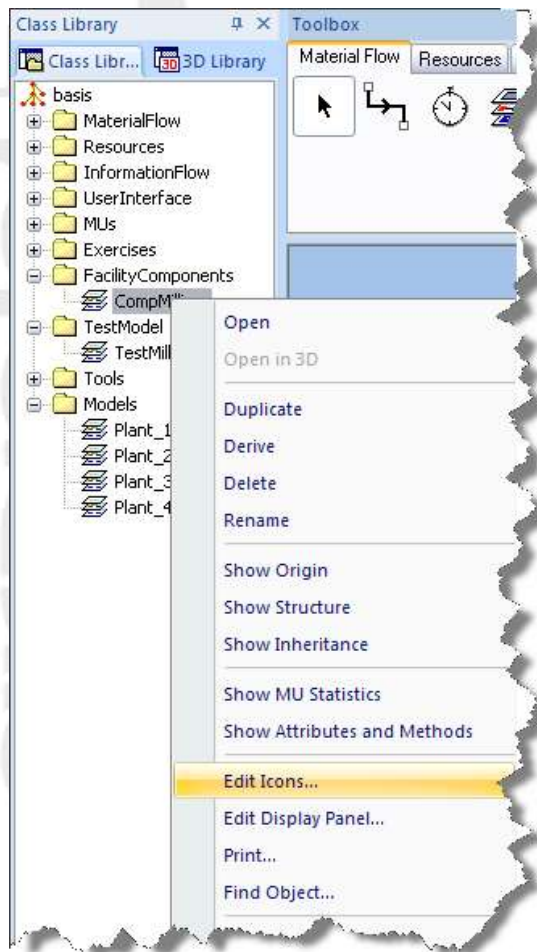
9. fejezet

Az ikonszerkesztő

A Plant Simulation ikont használ az objektum megjelenítésére a következő helyeken: *Class Library*, *Toolbox*, és *Frame*. Az ikonok alkalmasak az objektum műveleti állapotának kijelzésére is, pl: failed, paused, stb. A beépített ikonok módosítása, beleértve az animációs útvonalakat is, és új ikonok készítése az *Icon Editor* (ikon szerkesztő) segítségével lehetséges.



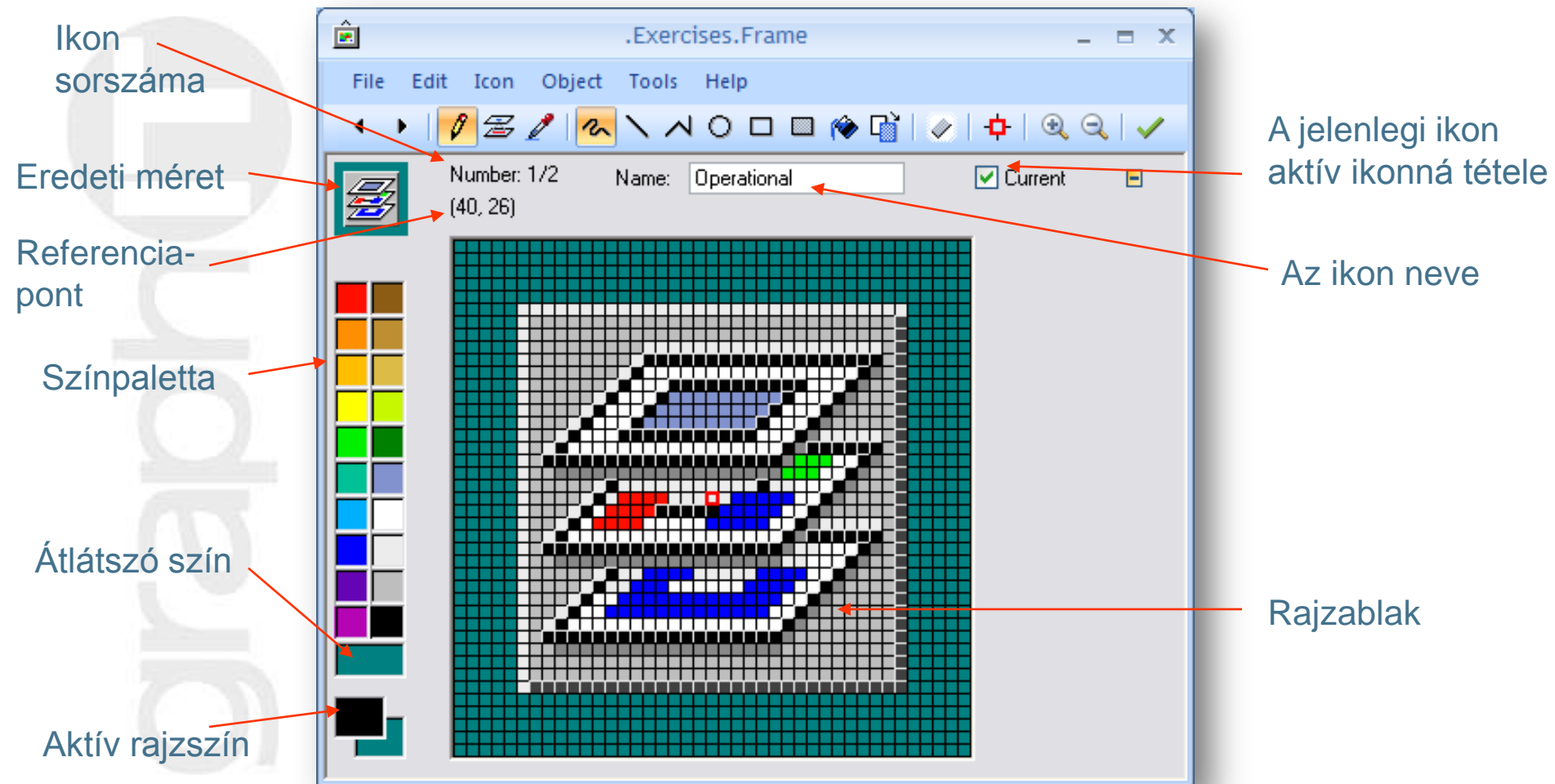
A beépített ikonok módosítása az összes objektumra, és új ikonok hozzáadása. Emellett megadhatók animációs pontok/útvonalak az anyagáram objektumokra, az MU-kra és a Frame-ekre.



Az Icon Editor megnyitása

Helyi menü > Edit Icons (Ikonok szerkesztése)

Az ikon egy osztálytulajdonság, ezért egy objektumnak a *Class Library*-ben és mindenhol máshol, ahol előfordul, azonos ikoncsoportja van. Az ikon módosítása egy helyen a teljes objektum osztályban történő megváltozását okozza!



- Tetszőleges számú ikon definiálható egy objektumra. A Plant Simulation a *Frame*-ben az objektumot az **Active** (aktív) ikont jeleníti meg.
- A Plant Simulation a Toolboxban a **0**-ás és **Default** nevű ikont jeleníti meg 40 x 40 képpont méretben. Ezek az ikonok módosíthatók.
- A *Frame*-hez hozzáadható egy ikon **Background** névvel, amit a Plant Simulation a *Frame* háttérképeként használ.
- Az ikonok maximális mérete 4000 x 4000 képpont.
- Az ikon jellemzők osztályszintű jellemzők.
- Az ikon animáció és az MU animáció aktivizálása csökkenti a teljesítményt.
- A Plant Simulation nem tartalmaz beépített állapotikonokat!

Az anyagáramlás és a mozgó objektumok képesek automatikusan változtatni az ikonjukat az állapotuk változása esetén. Ez csak akkor működik, ha az egyes állapotokhoz vannak ikonok definiálva az alábbi nevekkal:

Anyagáram objektumok:

- **working**
- **setUp**
- **no_Entry**

Anyagáram objektumok és Transporter:

- **failed**
- **paused**

Mozgó objektumok és Transporter:

- **operational**
- **waiting**

Az állapotikonok használata engedélyezhető:

Icons > Use State Icons (Ikonok > Állapot ikonok használata)
az ikonszerkesztőben.

Draw Mode
Rajz mód

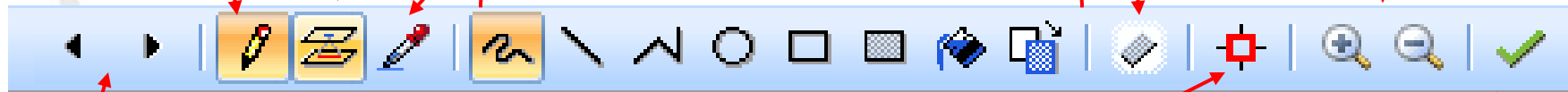
Animation Mode
Animáció mód

Eyedropper
Szemcseppentő

Drawing tools
Rajzeszközök

Delete All
Mindent töröl

Zoom In /
Zoom Out
Nagyítás /
kicsinyítés



Previous / Next Icon
Előző/Következő ikon

Freehand/Szabadkéz
Line/Vonal

Polyline/Polivonal

Ellipse/Ellipszis

Rectangle/Téglalap

Filled Rectangle/Kitöltött téglalap

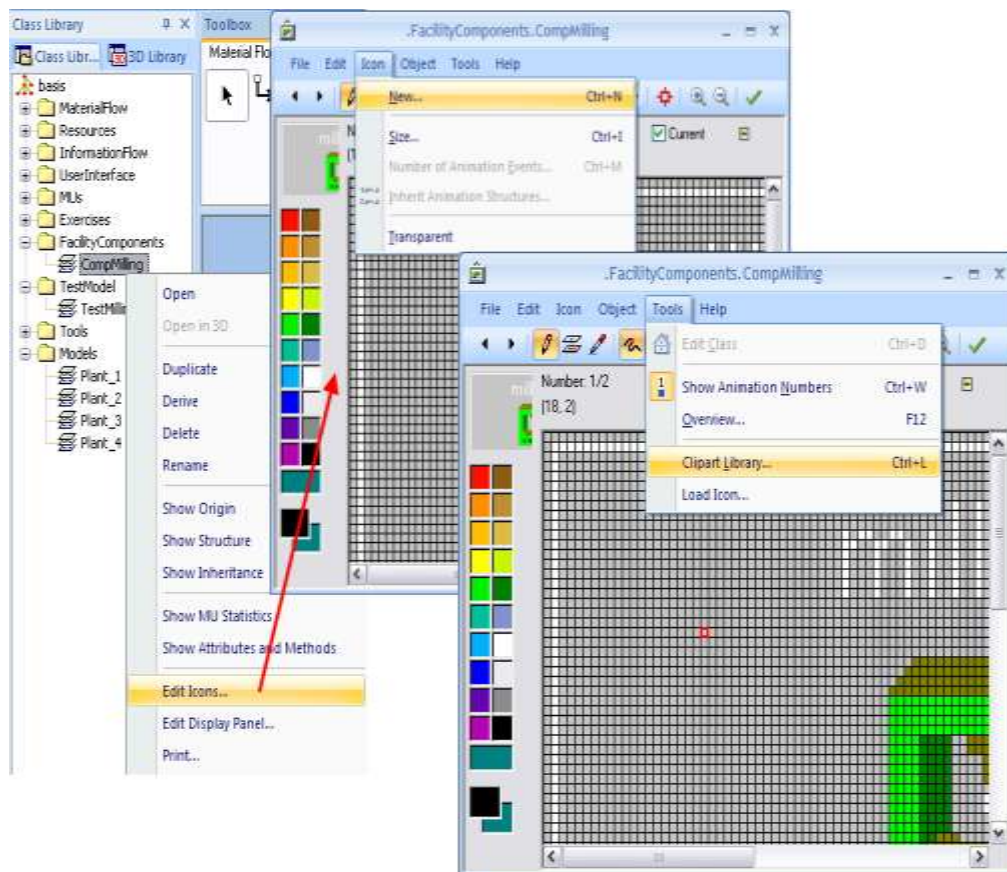
Fill Area/Terület kitöltése

Copy Area/Terület másolása

Set Reference Point
Referenciapont
megadása

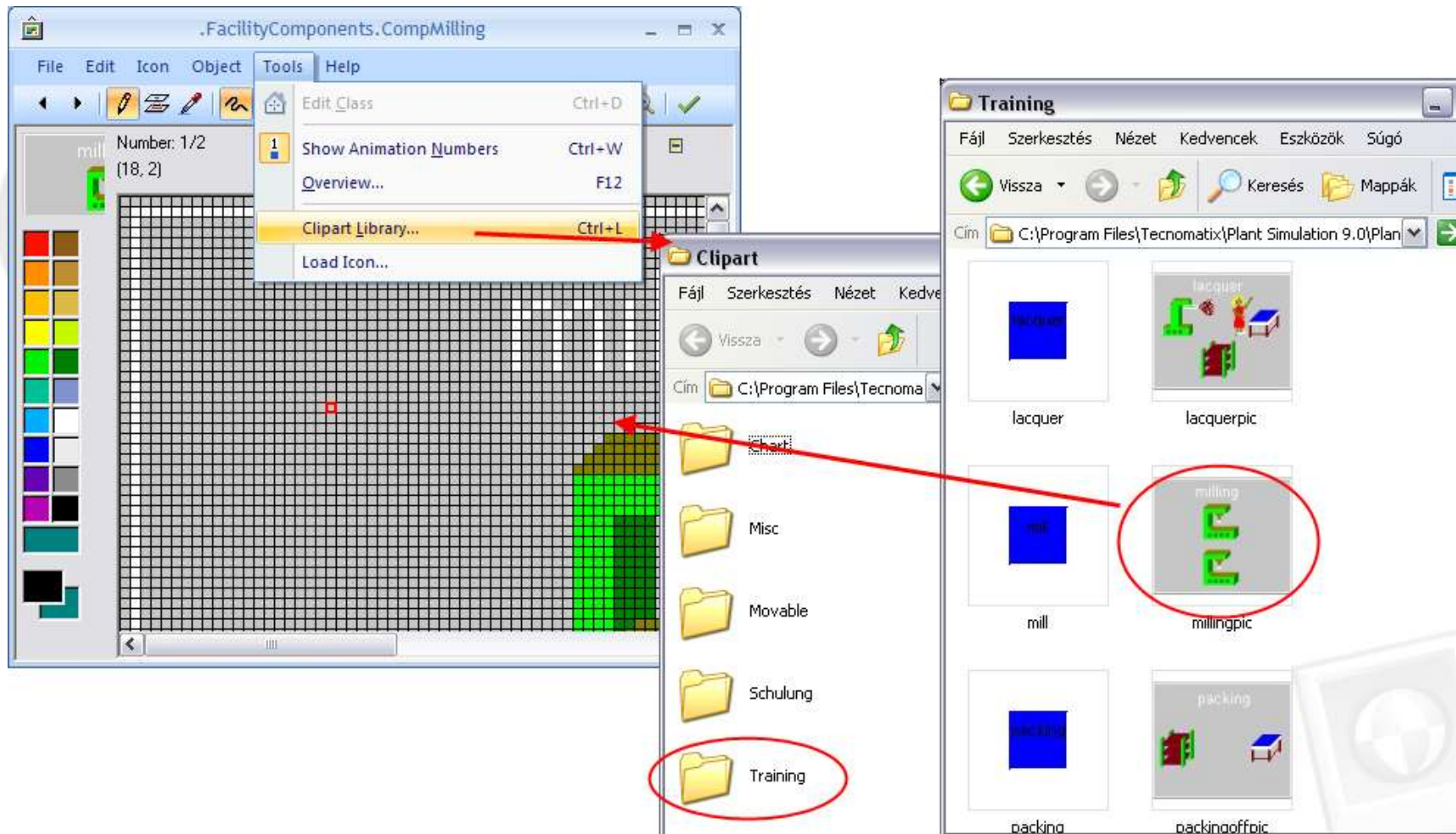
draw/delete
rajzol/töröl

Apply Changes
Változtatások
alkalmazása

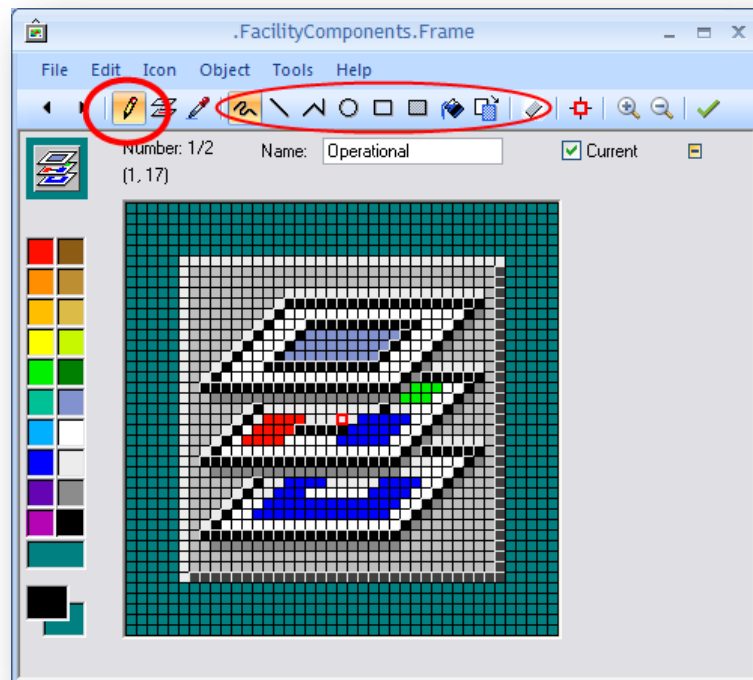


1. **Icon Editor** megnyitása (helyi menü a *Class Library*-ban vagy a *Frame*-ben az **Edit Frame Icons** parancs).
2. Új üres rajzterület megnyitása az új ikon számára
Icon > New (Ikon > Új)
(Ha szükséges, definiálja az ikon méretét (Size)).
3. Nyissa meg a Clipart Library-t:
Tools > Clipart Library
(Eszközök > Képelem könyvtár).





Illessze be ezt az ikont a *Training* könyvtárból húzd és ejtsd módszerrel.



A színpalettáról ki lehet választani egy színt az előre definiált színek közül.

Freehand drawing (szabadkézi rajz)

A **Freehand** (szabadkézi) funkcióval a rajzterületen kattintva pontok rajzolhatók, amik az ikont meghatározzák.

Alakzatok rajzolása

Kattintson a rajzolni kívánt alakzat nyomógombjára, kattintson egyszer az alakzat alappontjának megadásához, tartsa lenyomva az egér gombját, és mozgassa, amíg az alakzat a kívánt méretet ölti fel.

Fill Area (Terület kitöltése)

Kattintson a **Fill Area** gombra, majd a kitölteni kívánt területen belül az aktív rajzolási színnel történő területkitöltéshez.

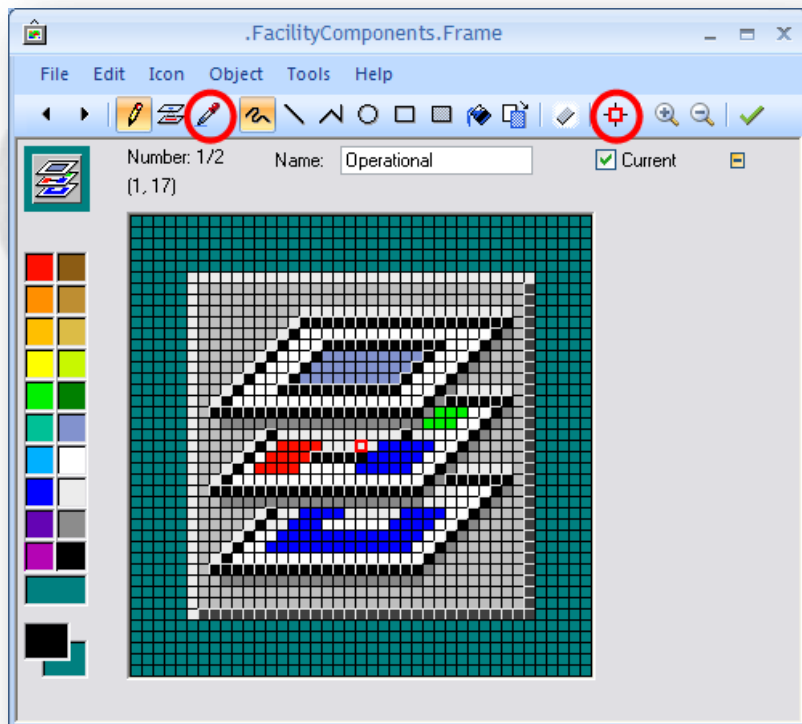
Referenciapont megadása/mozgatása

Minden ikonnak van egy referenciapontja.

A Plant Simulation ezt használja az objektum pozícionálásához, amikor egy *Frame*-be beillesztésre kerül.

Alapértelmezett helye a referenciapontnak a bal felső sarka az ikonnak.

A referenciapont máshova mozgatásához a **Set Reference Point** gombra, majd a rajzablakban a referenciapont új helyére kell kattintani.



Eyedropper (szemcseppentő)

A rajzolási szín beállítható az **Eyedropper** eszközzel. A kiválasztása után arra a színre kell kattintani a rajzablakban, amelyet rajzszíneként kíván használni.

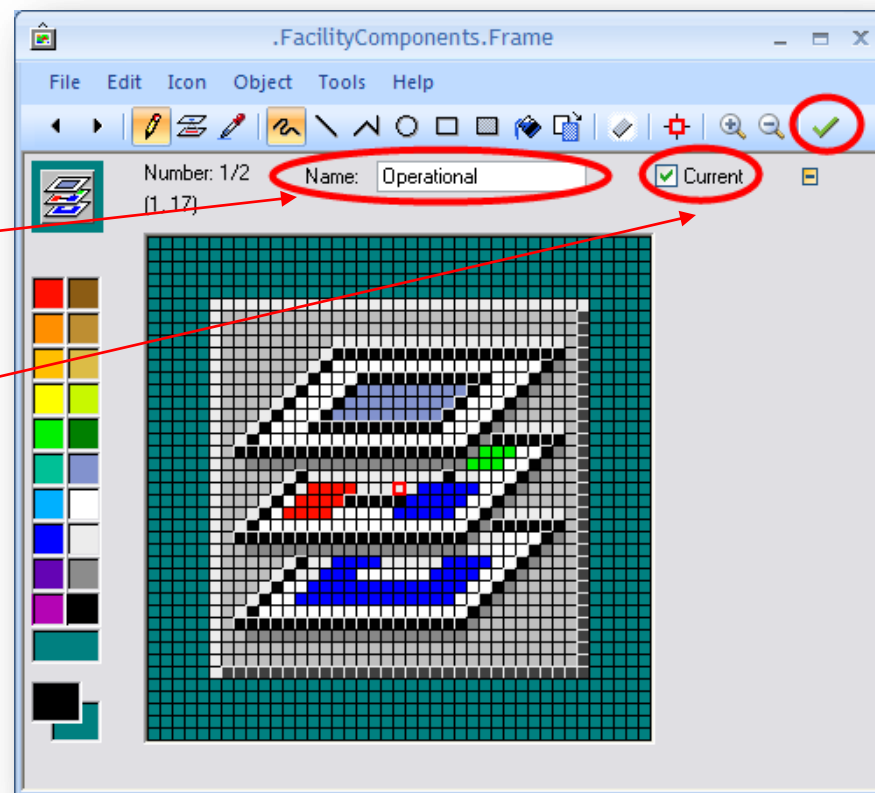
Mielőtt a változásokat alkalmazza az új vagy módosított ikonra, győződjön meg, hogy megadta a megfelelő

Name (Név)

Tulajdonságot az ikonra és bejelölte a

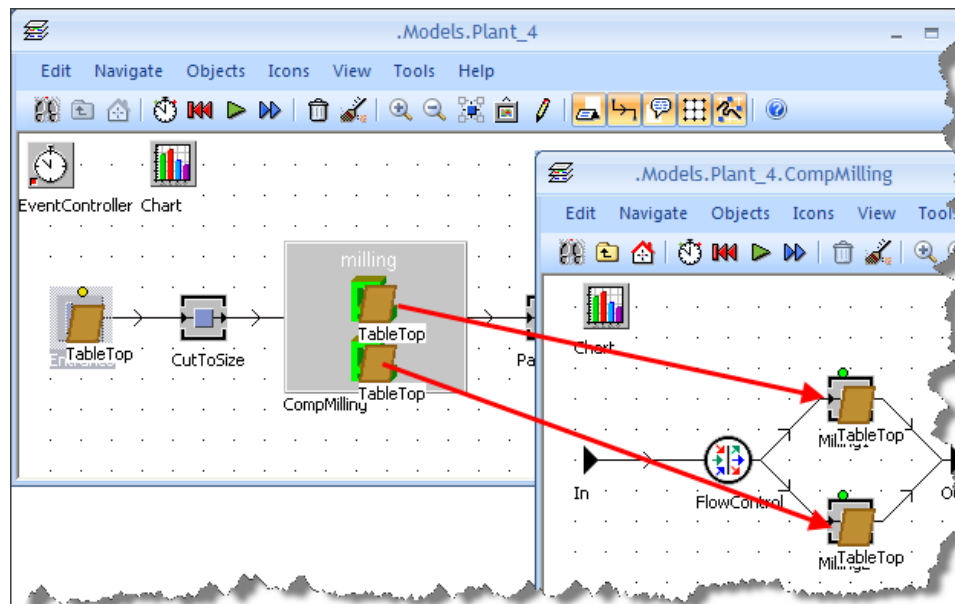
Current (Jelenlegi)

Tulajdonságot, hogy a Plant Simulation ezzel az ikonnal jelenítse meg az objektumot.



Közvetlenül kilépve az ikonszerkesztőből a változások mentésére **Yes** (Igen) választ kell adni.

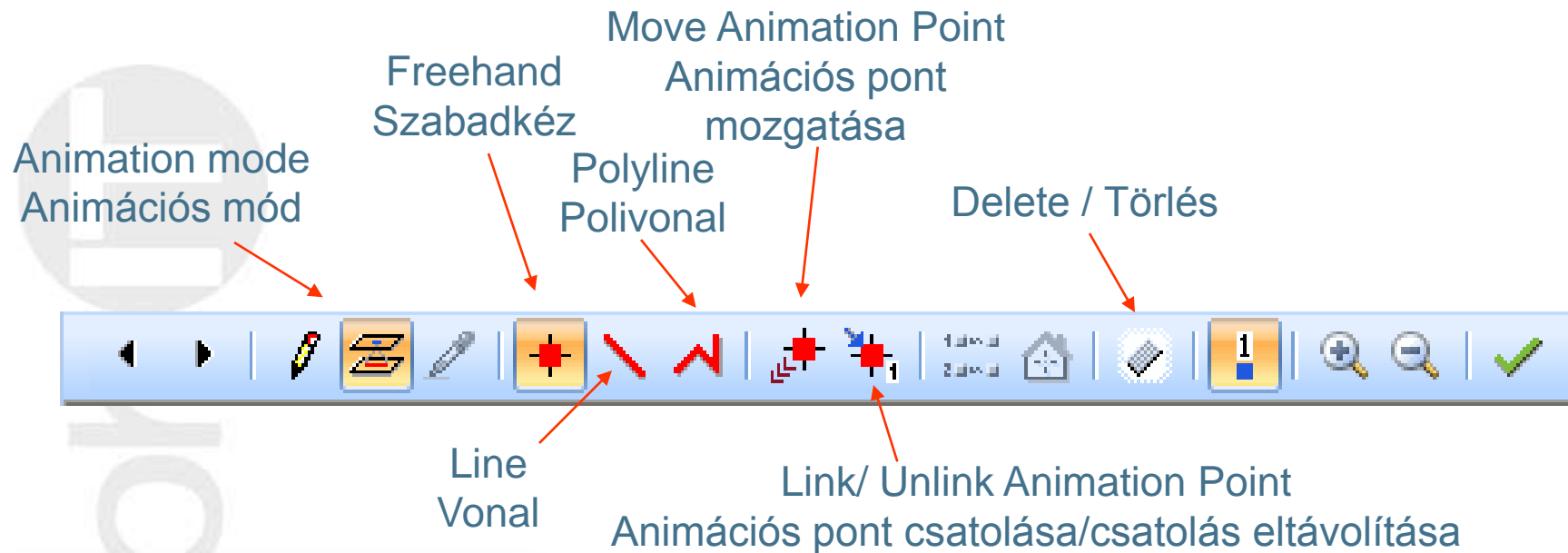
Az ikon elmenthető Eszközfüggelten bitképként (.dib), és Hordozható hálózati képként (.png) is a **File > Save** vagy **File > Save As** menüparancsokkal.



Ahhoz, hogy az anyagáram objektum ikonokon az MU-k ikonjai is megjelenjenek, animációs pontokat (**animation point**) kell megadni.

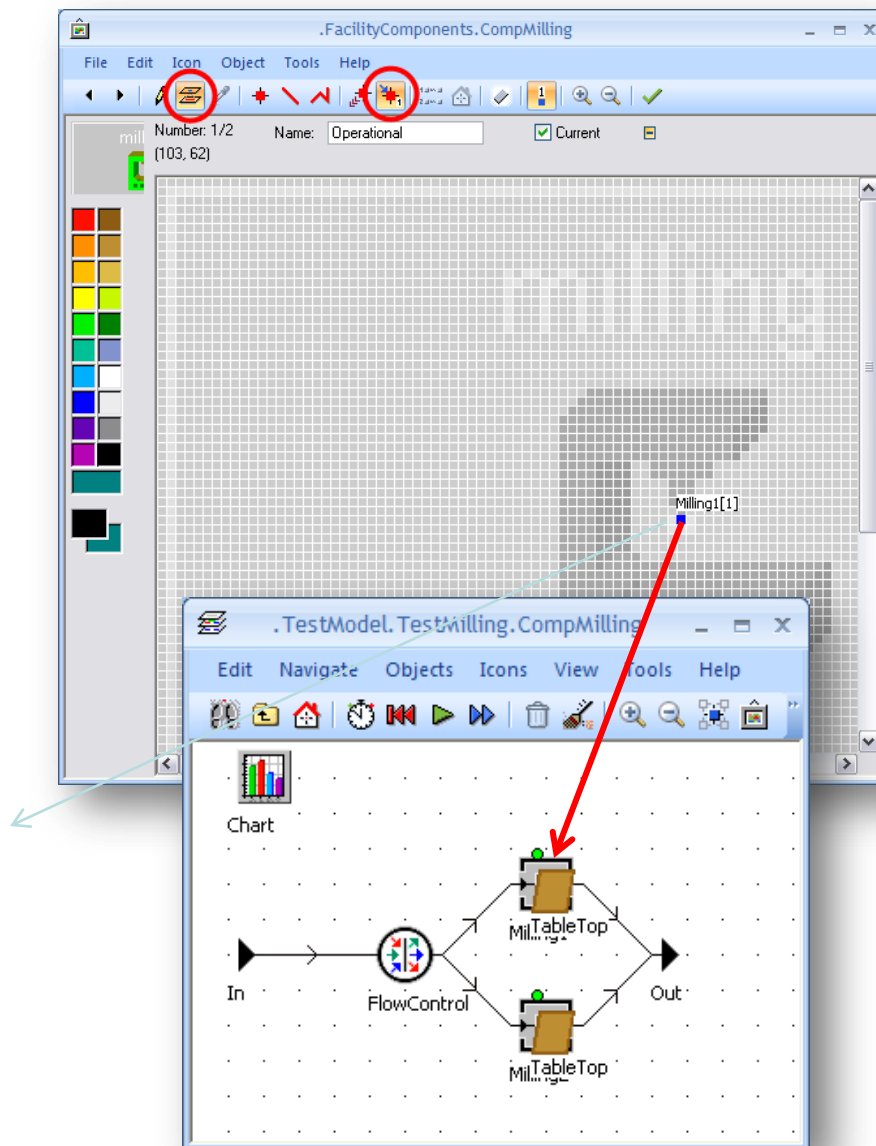
Az animációs pont egy olyan referenciapont, amihez az MU illeszti a saját referenciapontját amikor az objektumon megjelenik, vagyis ahol az MU megjelenik az animáció során. A hossz-orientált objektumokra **animációs vonal** is megadható.

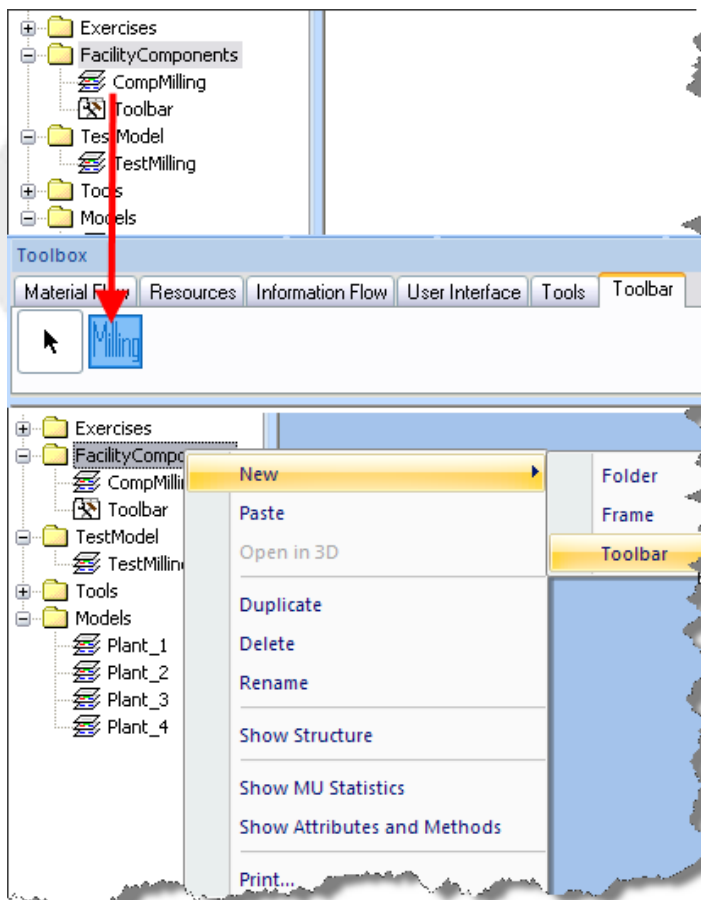
Az animációs pontok/vonalak beillesztéséhez az ikonszerkesztőben az **Animation mode** (Animációs mód) gombra kell kattintani az eszköztáron.



- A rajzeszközök használhatók az animációs pontok és animációs vonalak megadásához.
- A *Connector* objektum használható az objektumok és a komponenseket tartalmazó *Frame*-ek összekötésére.
- A törlés eszközökkel lehet az animációs pontokat és vonalakat törölni.

1. Kattintson az **Animation Mode** gombra.
2. Rajzoljon animációs pontot vagy vonalat.
3. Kattintson a **Link/ Unlink Animation Point** gombra és kattintson az animációs pontra vagy vonalra. Ekkor megnyílik a kapcsolódó *Frame*.
4. Válassza ki azt az állomást, amelyhez hozzá kívánja rendelni az animációs pontot. A Plant Simulation az állomás nevét címkeként hozzárendeli az animációs ponthoz.





A **Toolbox** (paletta) ikon módosításához a 0-ás sorszámú és *Default* nevű ikont kell szerkeszteni (maximum 40 x 40 képpont méretű!).

A definiált komponensekhez új eszköztárak adhatók

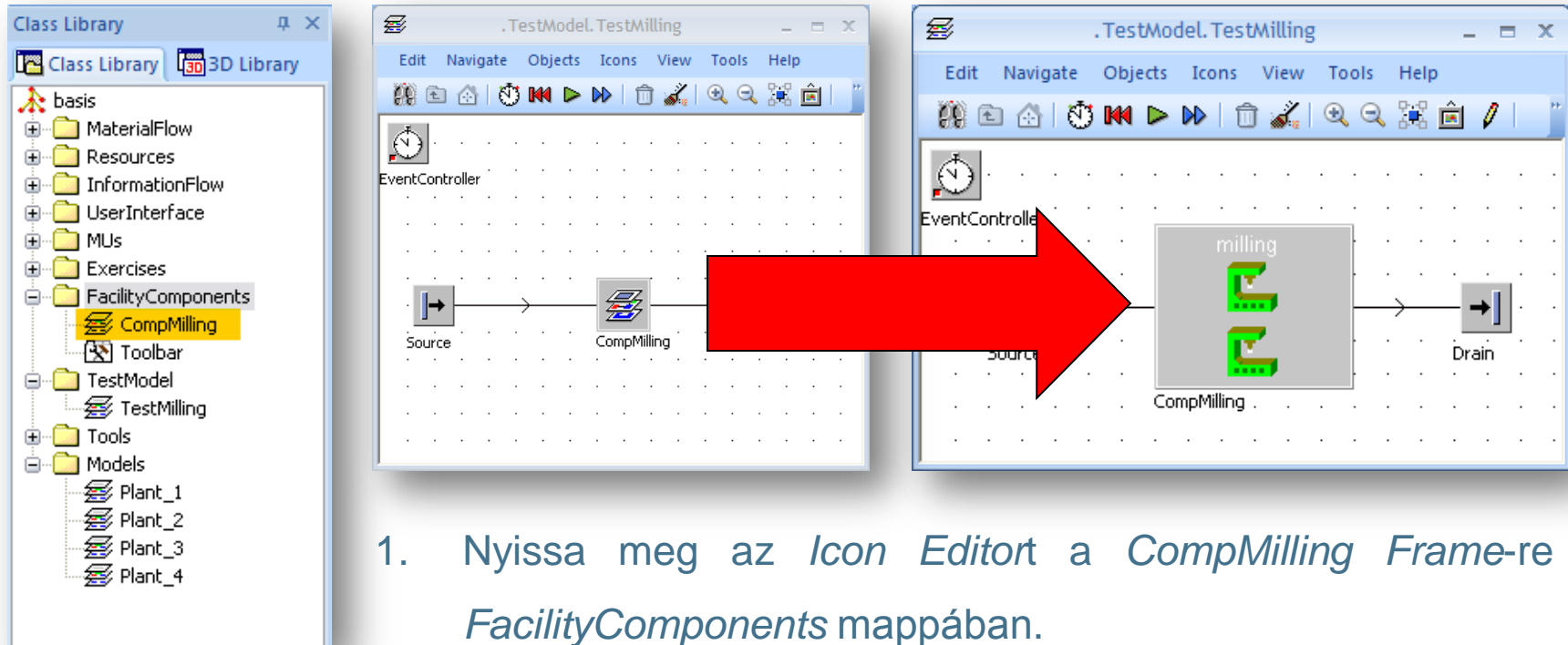
Helyi menü > **New** > **Toolbar** (Új > Eszköztár)

A beépített eszköztárak kibővíthetők vagy módosíthatók:

Tartsa lenyomva a **Shift** billentyűt, és húzza az objektumot az eszköztár ikonra és engedje fel az egérgombot.

Objektum vagy osztály megnyitható, eltávolítható az eszköztárról a helyi menüjéből.

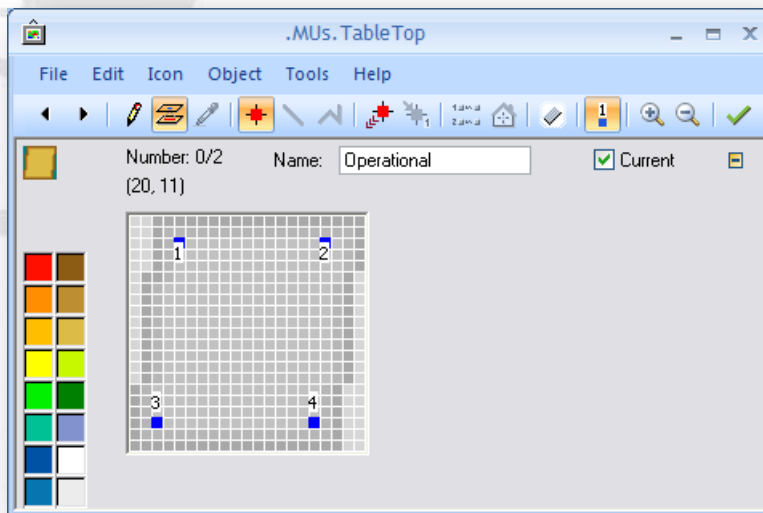
Ikonok módosítása, animációs pontok megadása



1. Nyissa meg az *Icon Editort* a *CompMilling Frame*-re a *FacilityComponents* mappában.

2. Válassza ki az **Icons > New** menüpontot és a marógép ikont a *Clipart Library* -ből.
3. Kattintson az **Animation Mode** gombra és adjon meg két animációs pontot a gép ikonján. Kattintson a **Link/Unlink Animation Point** gombra, és csatolja az animációs pontokat a *Milling1* és *Milling2* objektumokhoz a *Frame*-ben.

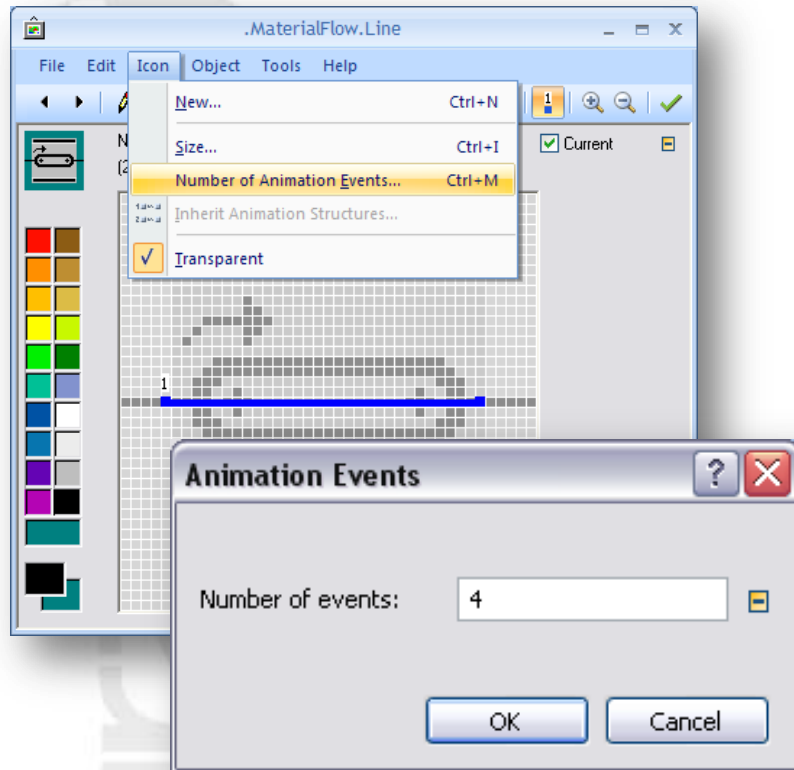
Alkatrészek animálása felhasználó által definiált MU ikonokon



A Plant Simulation az MU-k referenciapontját **hozzailleszti** az anyagáram objektumok referenciapontjához.

Saját MU ikon készítésekor, ha azok más MU-kat is tartalmaznak, meg kell adni animációs pontokat, hogy a Plant Simulation megfelelően tudja animálni azokat.

Animáció a hossz-orientált objektumokon



Annak megmutatására, hogy miképpen mozog az MU a hossz-orientált objektumokon (Line, Track, Buffer, PlaceBuffer), meg kell rajzolni az animációs vonalat az ikonon.

Ez után meg kell adni az animációs lépések számát a menüben:

Tools > Number of Events (Eszközök > Események száma)

menüpont alatt. Az animációs események száma megadja, hogy a Plant Simulation milyen sűrűn mutatja az MU-t a hossz-orientált objektumon a szimuláció futása során.



10. fejezet

Osztályok, példányok és öröklődés

Az objektum orientált programok egyik legfontosabb jellemzője az **Öröklődés** (Inheritance).

Minden Plant Simulation objektum a *Class Library*-ben egy osztály.

Ezek az osztályok kibővíthetők, és készíthetők új osztályok, amelyek jellemzőket örökölhetnek a szülő osztályuktól.

Osztály objektumokat használva a szimulációs modellben, használható az öröklődés mivel az osztály összes jellemzője átöröklődik az objektumokba, amelyek belőle származnak.

Ha egy jellemző az osztályban módosul, akkor az összes az osztályból származtatott objektumban is megváltozik. Ez sokkal biztonságosabb és hatékonyabb, mint egyesével megváltoztatni az adott tulajdonságot minden előfordulásban.

Osztály: Minden objektum a *Class Library*-ben osztály. Ezek sablonokként működnek azon példányokhoz, amelyeket a *Frame* - ekbe be lehet illeszteni.

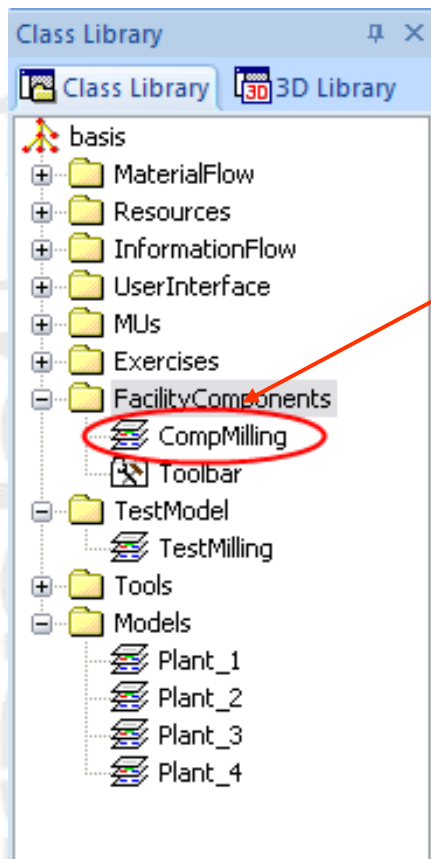
Alosztály: Olyan objektum a *Class Library*-ben, amely a legfontosabb jellemzői más osztálytól származtatódnak. Minden egyéb tulajdonság erre az alosztályra jellemző.

Példány: A példány vagy előfordulás egy a szimulációba illesztett objektum a *Class Library* - ből vagy a Toolbox - ből.

Öröklődés: Az osztály az összes jellemzőjét átadja a belőle származtatott alosztálynak vagy példánynak. Az öröklődés csak az osztálytól az alosztály és példány irányába működik, a másik irányba nem!



Ha egy tulajdonságot az összes példányban módosítani kell, akkor az osztály objektumot érdemes módosítani.

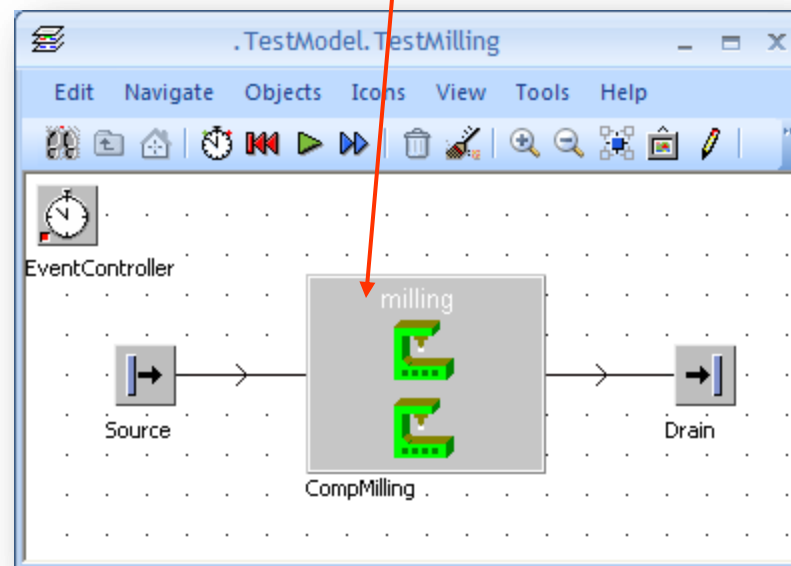


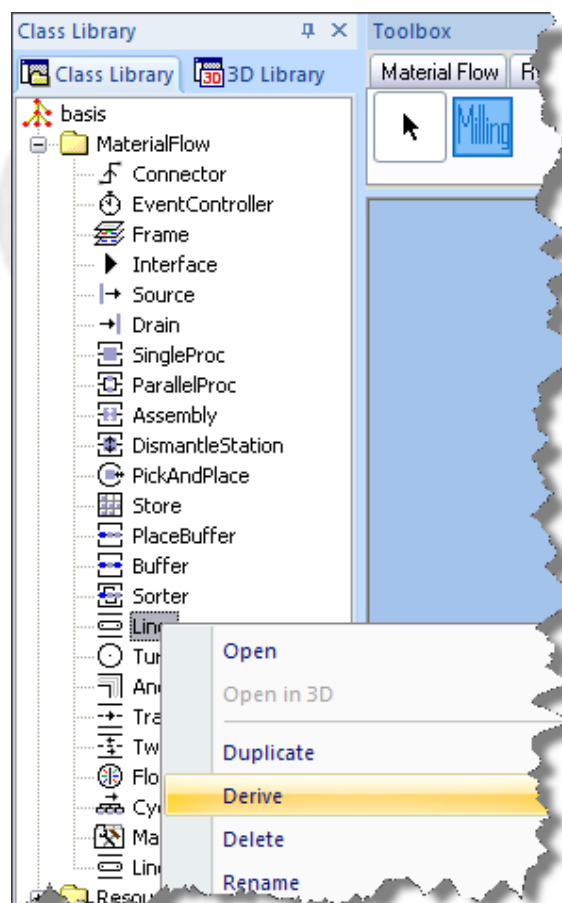
A *Milling* osztály a *Class Library*-ban.

**Class
Osztály**

A belőle származtatott példány beillesztve
a *TestMilling Frame*-be.

**Instance
Példány**

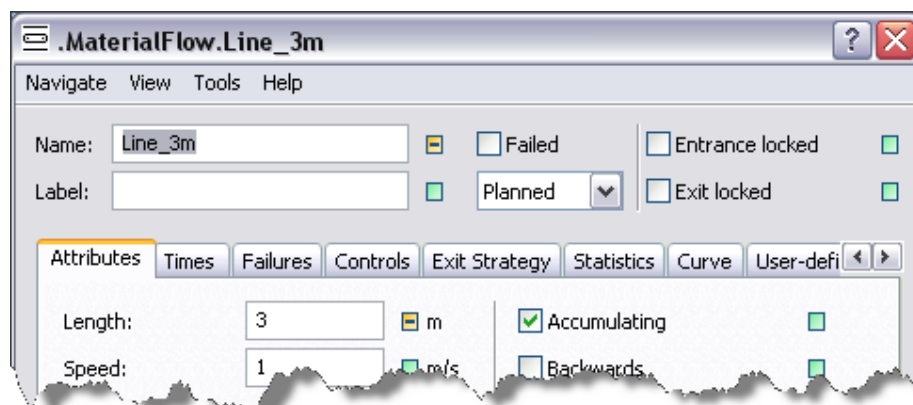


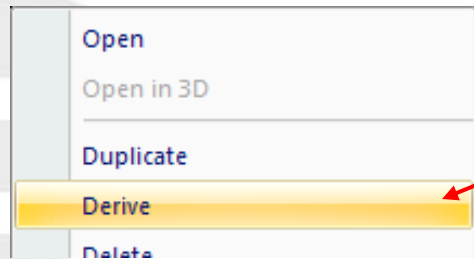


A *Line* osztály megadja az objektum jellemzőit, mint például a hossz, a végsebesség, a kapacitás, stb.

Készíthető két további hasonló objektum, amelyeknek a hossza különbözik. Minden más jellemző ugyanaz marad.

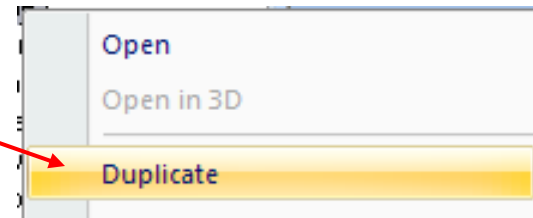
A hossza (*Length*) ki kell kapcsolni az Öröklődés (*Inheritance*) gombot, kikapcsolt állapotban nem öröklí a tulajdonságot. Ezen kívül minden más jellemzőt a *Class Library Line* osztályától öröklí.

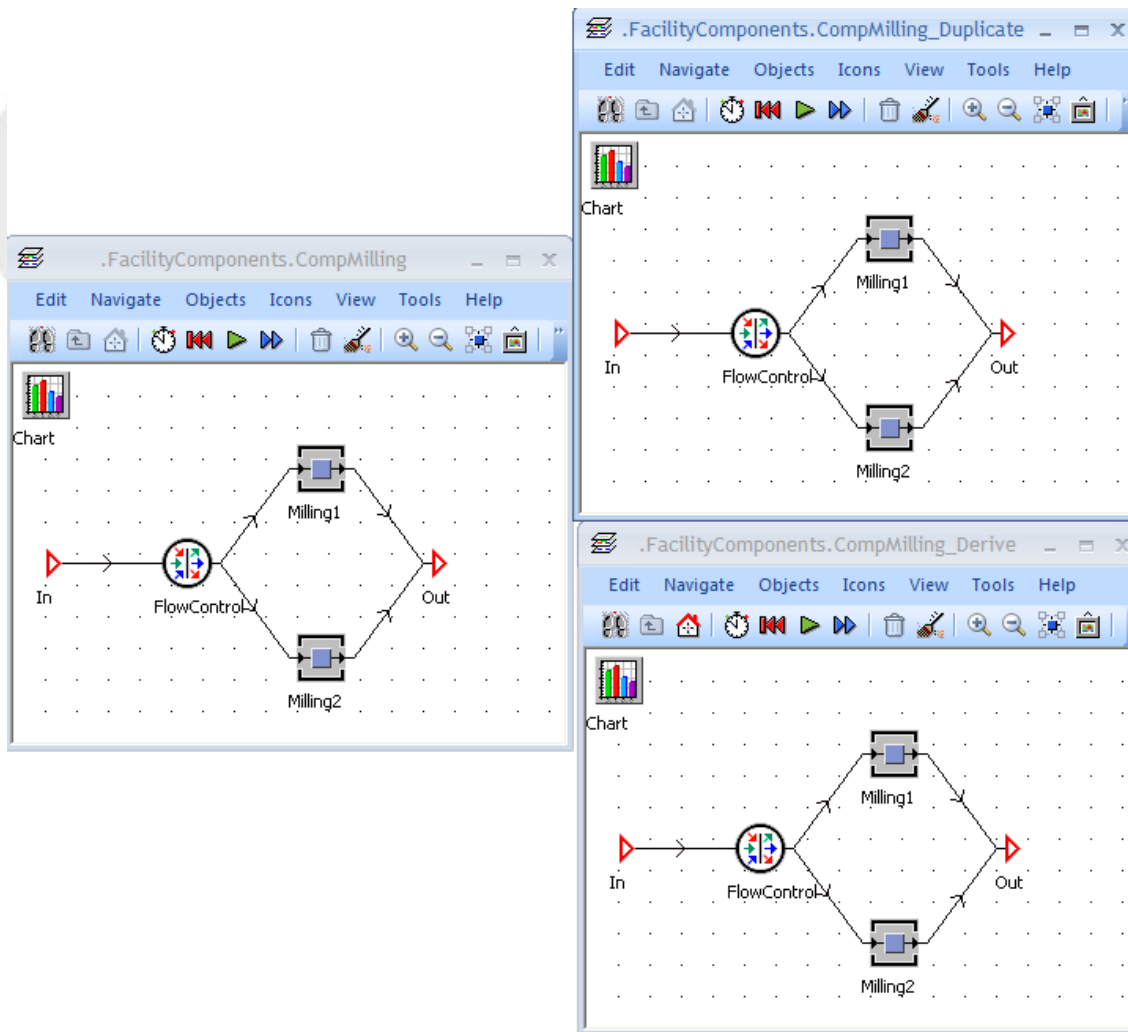




A Line objektum helyi menüjéből a **Derive (származtatás)** parancsot kell kiválasztani az osztály egy alosztályának elkészítéséhez. Az alosztály az osztálytól örökli a jellemzőinek nagy részét.

A Line objektum helyi menüjéből a **Duplicate (másolás)** parancsot kell kiválasztani egy új osztály készítéséhez. Minden öröklődési kapcsolat az osztály és a másolat osztály között megszűnik, csak átmásolódnak a jellemzők.

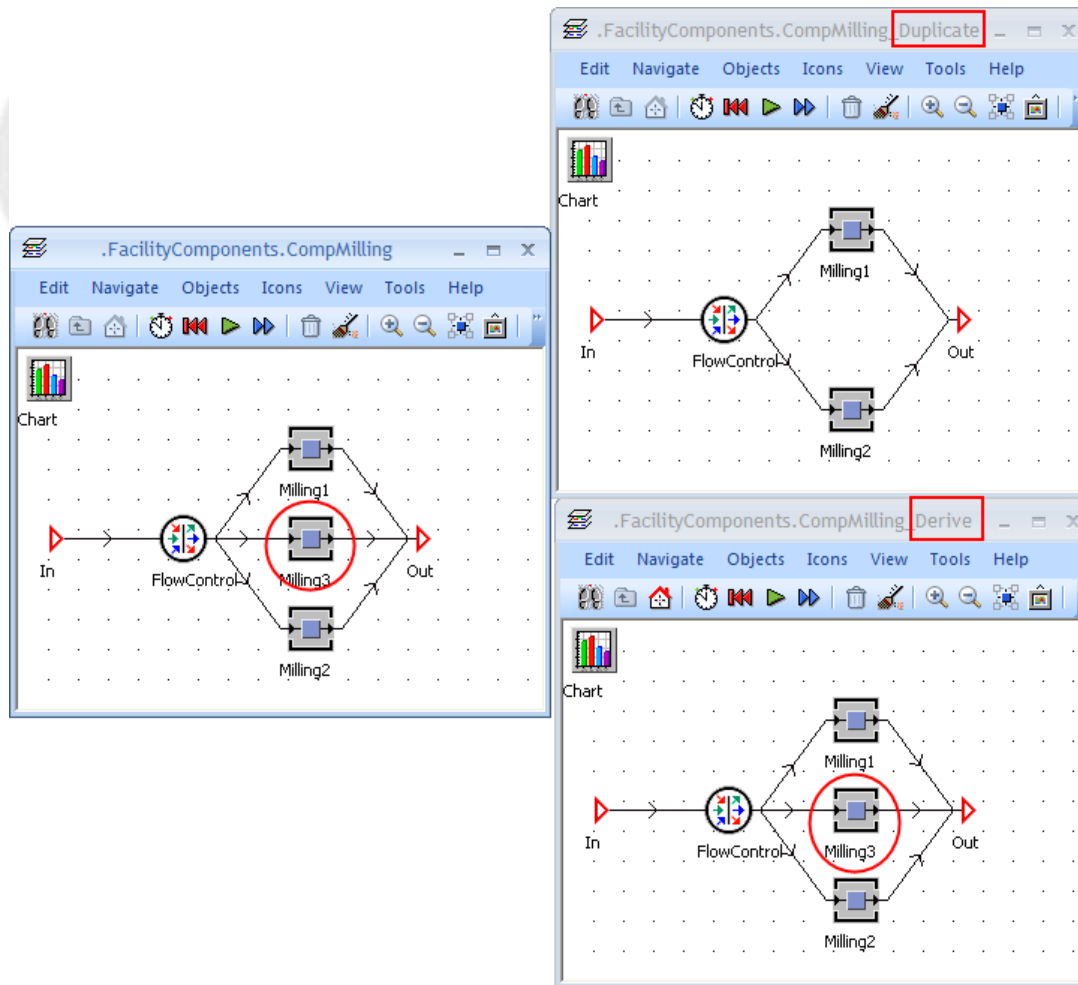




Készítsen egy másolat *Milling* és egy alosztály *Milling* osztályt a *Frame Milling* osztályból a *Class Library*-ben.

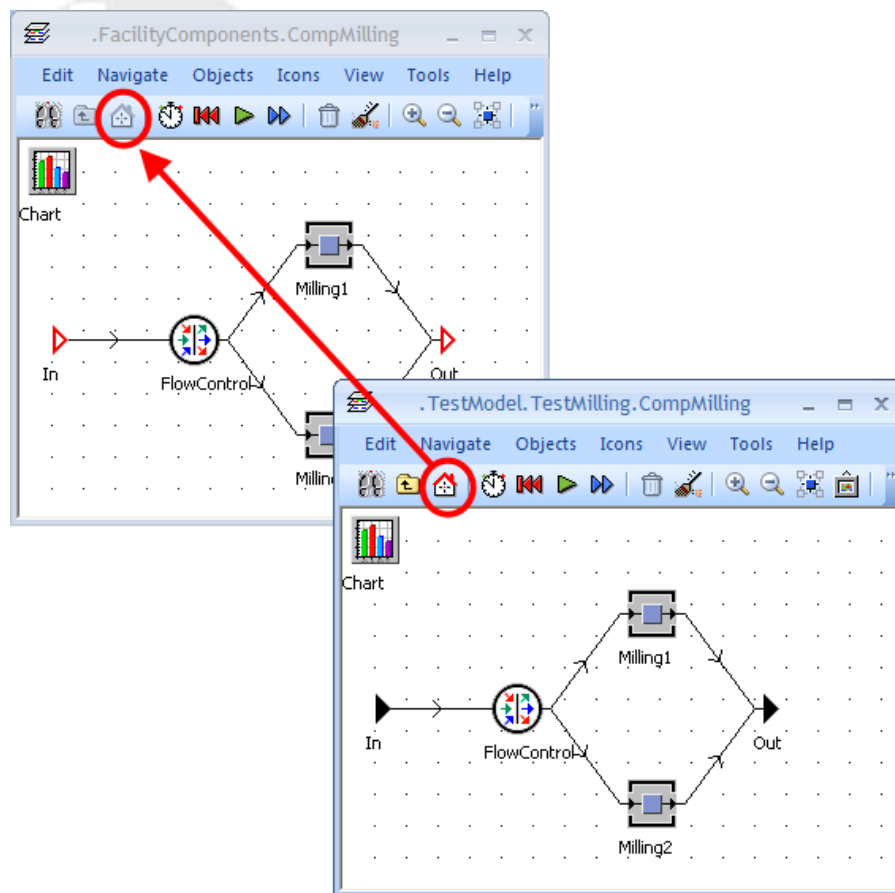
Ez a három objektum azonos struktúrát képvisel.

Most módosítsa a *Milling* osztályt, még egy marógép hozzáadásával.



Mivel az öröklődés aktív, minden változás, ami az osztályban történik, öröklődik az alosztályba is.

A másolat viszont nem érintett ezek által a változtatások által, mivel az öröklődés a másolat esetében nem érvényesül.

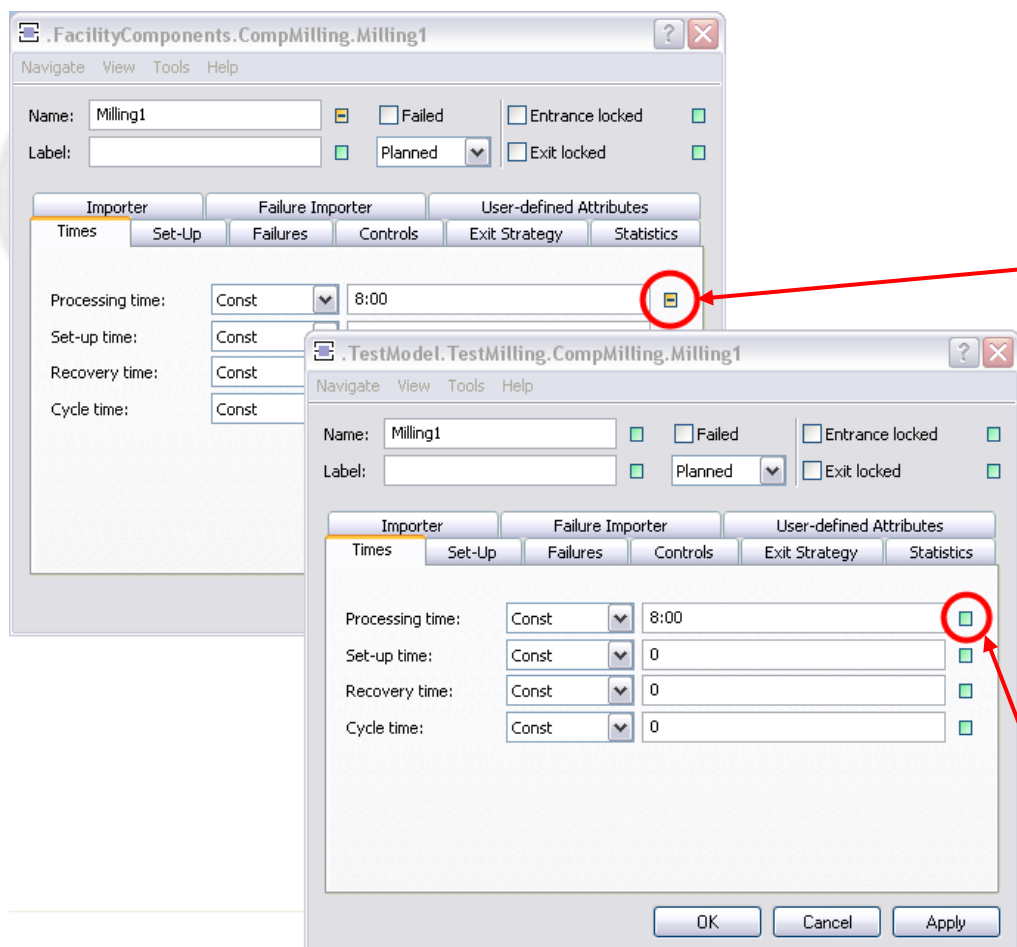


Az **Open Origin** (Eredet megnyitása) akkor aktív a *Frame* eszköztáron, ha az objektumnak van szülője.

A gombra kattintva a Plant Simulation megnyitja a *Frame* szülőjét.

Ez azt jelenti, hogy a Plant Simulation az ezen a szinten végrehajtott módosításokat az innen származtatott hierarchikus szintekre továbbvezeti.

A Plant Simulation az *Interface* objektum ikonját kitöltött fekete színben mutatja, ha *Connector* objektum kapcsolódik a *Frame*-hez.

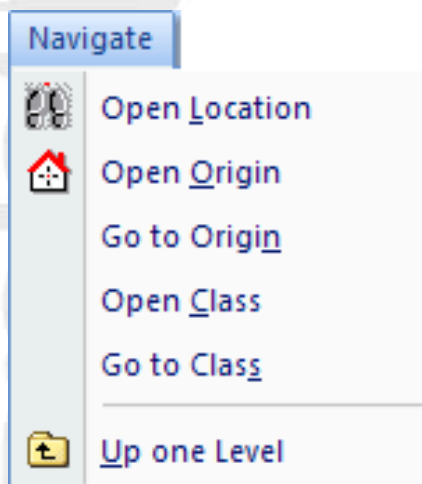


A párbeszédablak elem melletti kapcsológomb állapota mutatja, hogy az öröklődés aktív vagy nem.

A mínusz jeles gomb azt jelenti, hogy az öröklődés nem aktív. A megadott értékek csak a jelenlegi objektumra vonatkoznak, az osztály példányára, és nem az egész osztályra.

A mínusz jel nélküli gomb azt jelenti, hogy az öröklődés aktív. Ekkor az objektum öröklí a szülőjétől az értéket.

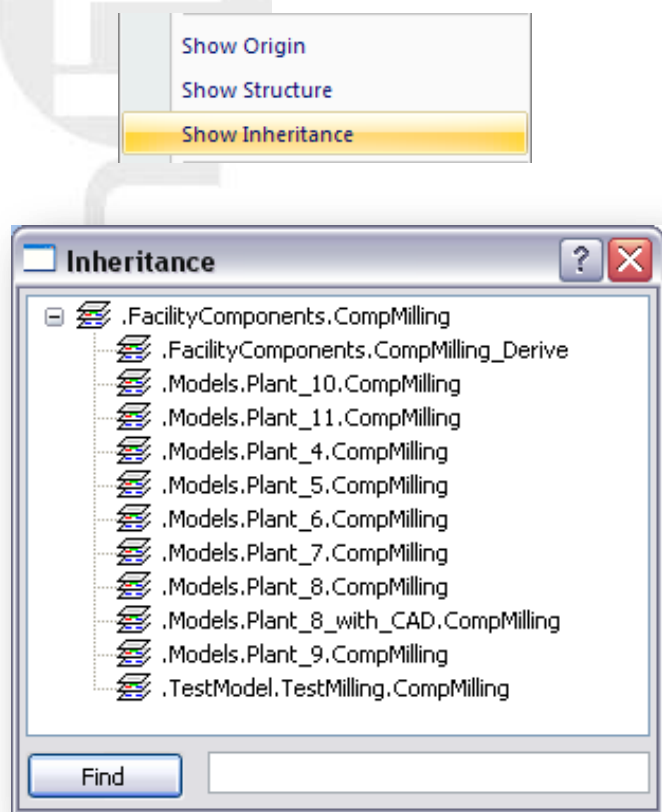
A **Navigate** (Navigálás) menü alábbi menüpontjai használhatók az öröklődési struktúrán történő végighaladásra.



- **Open Location** (Hely megnyitása): annak a mappának a megnyitása a *Class Library*-ben, ahol az objektum található.
- **Open Origin** (Eredet megnyitása): az objektum szülőjének megnyitása, ami a következő szint fölfelé az öröklődési struktúrában.
- **Go To Origin** (Eredetre váltás): a szülő objektum megnyitása és a példány bezárása.
- **Open Class** (Osztály megnyitása): az objektum osztályának megnyitása
- **Go To Class** (Osztályra váltás): az objektum osztályának megnyitása, és a példány bezárása.

Ha nincs magasabb szintű öröklődési elem akkor csak az **Open Location** menüparancs aktív.

Az alábbi parancsok valamelyike használható az öröklődési struktúra ablakban történő megjelenítéséhez.



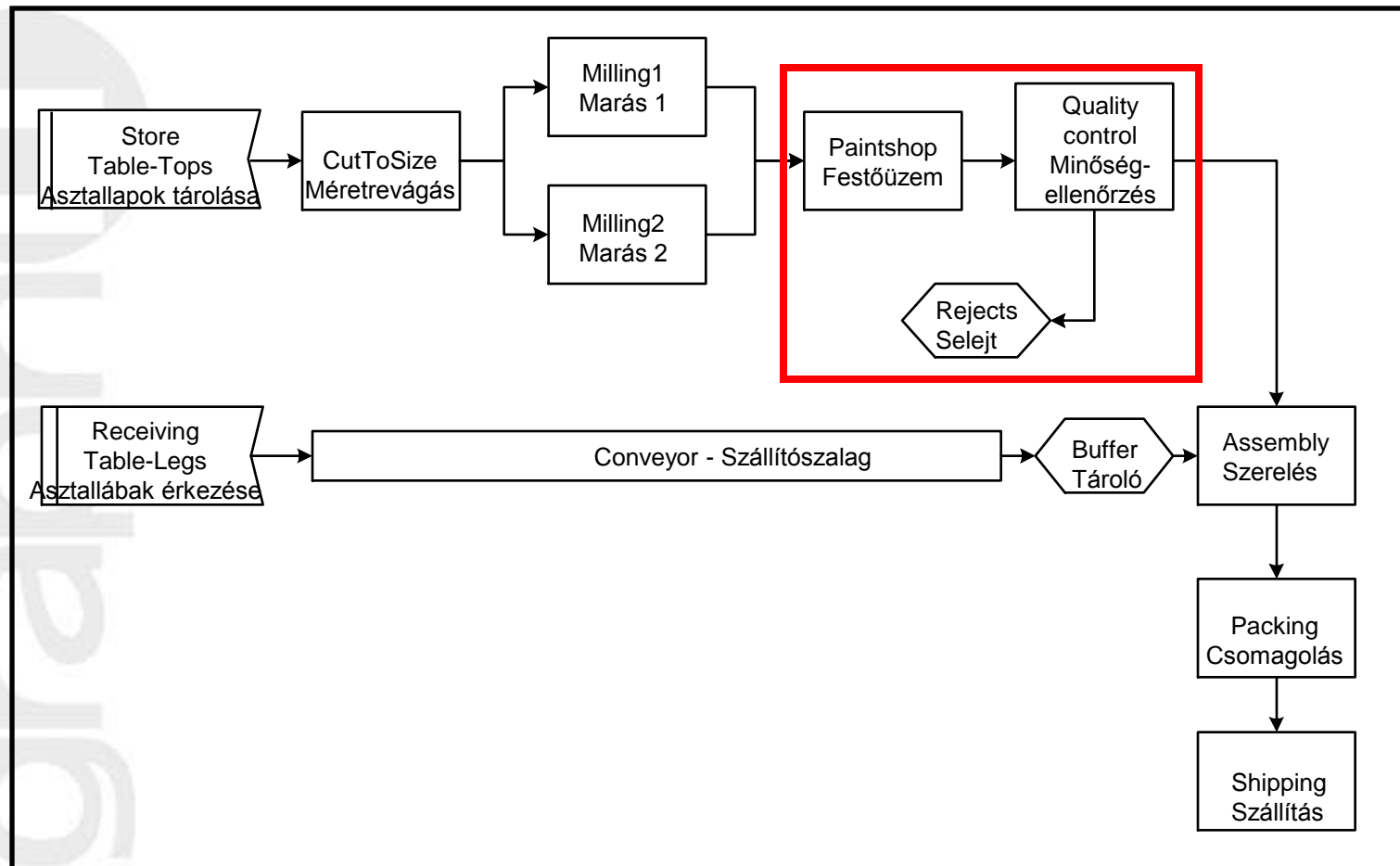
- **Show Origin** (eredet mutatás): a kiválasztott objektum szülőjének mutatása.
- **Show Structure** (struktúra mutatása): a **Structure** ablak megnyitása, ami a kiválasztott Frame struktúráját mutatja az összes benne lévő objektum nevével.
- **Show Inheritance** (öröklődés mutatása): az **Inheritance** (öröklődés) ablak megnyitása, ami mutatja az objektumok útvonalát, amelyek ebből az objektumból származnak.

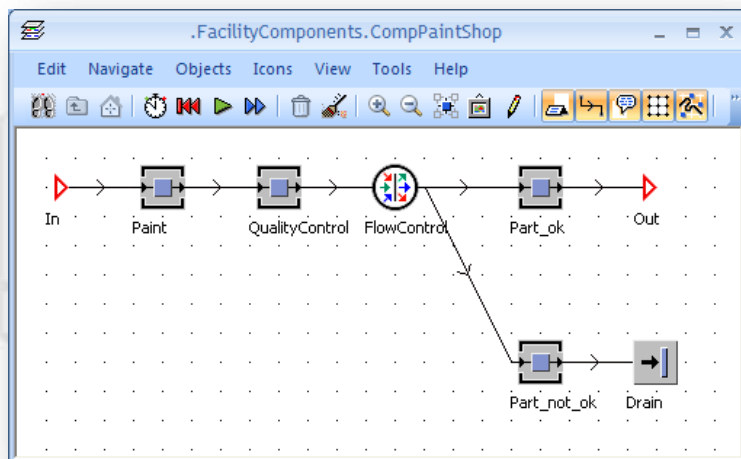
* A *Frame* is egy objektum.



11. fejezet

Adattípusok,
Egyéni jellemzők,
Táblázatok





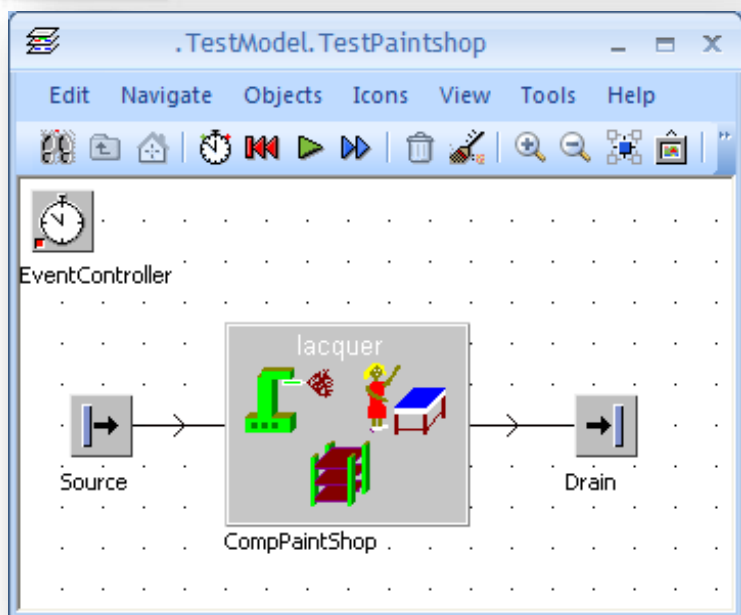
A festőüzem az asztallapokat a festés minősége szerint válogatja szét: jó vagy rossz kategóriába.

Szimulálni kellene a 10%-os visszaküldési arányt.

Ebből a célból az asztallapokhoz hozzá kell adni egy jellemzőt, ami a festés minőségét jelzi: minden tízből egy asztallap rossz kell, hogy legyen.

A festőüzem ellenőrzi ezt a jellemzőt, és ettől függően küldi tovább az asztallapot.

Ehhez többet kell tudnunk az **adattípusokról**, **táblázatokról**, és a *FlowControl* objektumról.



Minden Plant Simulation objektumnak vannak beépített **attribútumai** (pl.: hossz, sebesség, kapacitás, név, stb.) és **metódusai** (pl.: createAttr, closeDialog, moveToFolder, stb.).

Az **attribute** (attribútum) adja meg egy objektum viselkedését, vagy állapotát. A **method** (metódus) információt gyűjt ki az objektumról, és egy értékkel tér vissza. Értéket számít ki, vagy módosítja az objektum viselkedését.

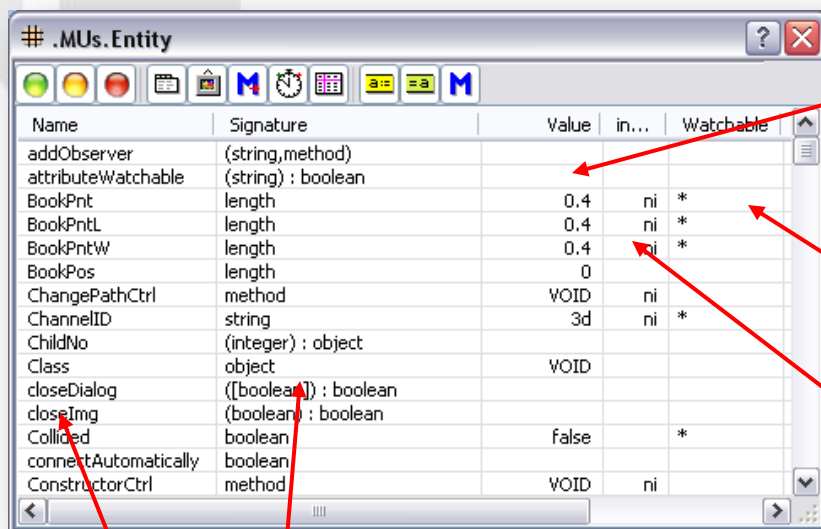
A **Show Attributes and Methods** (Attribútumok és metódusok mutatása) ablak az objektum beépített attribútumait és metódusait jeleníti meg. Az objektum helyi menüjéből nyitható meg

- **Show Attributes and Methods** paranccsal vagy a
- **Show Attributes and Methods** paranccsal a *Frame Objects* menüjében.

Megjeleníthető egy objektumot kiválasztva és **F3** billentyűt nyomva is.

Emellett tetszőleges számú felhasználói attribútum is megadható. A Plant Simulation ugyanúgy kezeli ezeket, mint a beépített attribútumokat.

Duplán kattintva az attribútum nevéen megjelenik egy ablak, ahol módosítható az attribútum értéke.



Name	Signature	Value	in...	Watchable
addObserver	(string,method)			
attributeWatchable	(string) : boolean			
BookPnt	length	0.4	ni	*
BookPntL	length	0.4	ni	*
BookPntW	length	0.4	ni	*
BookPos	length	0		
ChangePathCtrl	method	VOID	ni	
ChannelID	string	3d	ni	*
ChildNo	(integer) : object			
Class	object	VOID		
closeDialog	([boolean]) : boolean			
closeImg	(boolean) : boolean			
Collided	boolean	false		*
connectAutomatically	boolean			
ConstructorCtrl	method	VOID	ni	

Value (érték). Csak az attribútumoknak van értéke.

Azt jelzi, hogy a Plant Simulation tudja-e mutatni az attribútum értékét vagy állapotát (pl. gyakran változó értékeket nem)

Azt mutatja, hogy az érték örökölt (i) vagy nem örökölt (ni)

Az attribútum adattípusa, vagy a paraméterek adattípusa

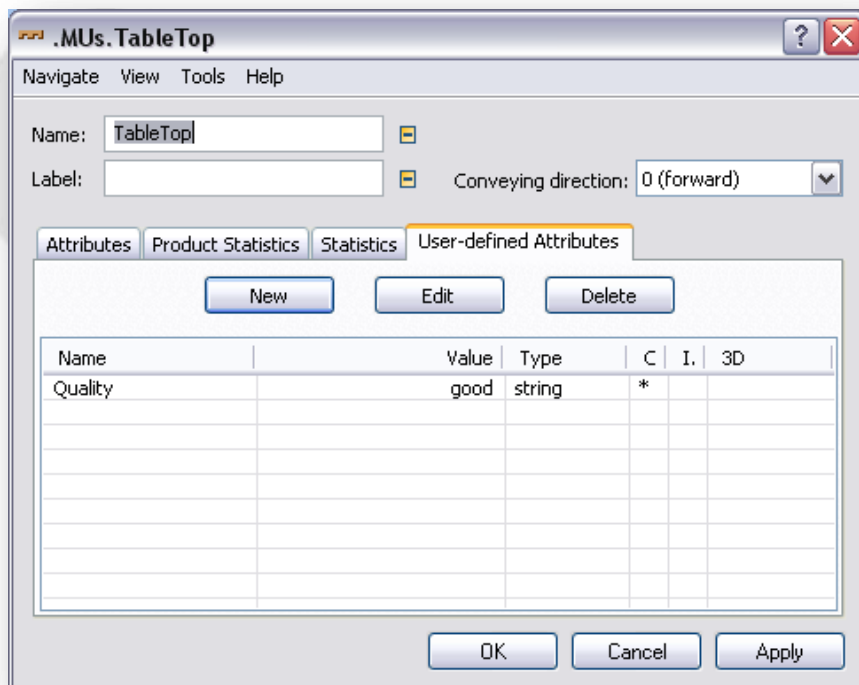
Az attribútum vagy metódus neve

boolean	TRUE vagy FALSE
integer	egész érték
real, length, weight, speed, money	lebegőpontos szám
string	karakterek, számok és speciális karakterek
date	dátum kifejezés (yyyy/MM/ dd)
time	idő kifejezés (hh:mm:ss.ss)
datetime	dátum kifejezés, az időt is tartalmazva (yyyy/MM/dd mm:ss)
list, stack, queue	egy oszlopos lista
table	lista kettő vagy több oszloppal
object	referencia szimulációs modellre vagy objektumra

Egyéni attribútumok definiálhatók kiegészítendő az objektumokat olyan tulajdonságokkal, amelyekre a modellezés során szükség van. A szimuláció során a Plant Simulation le tudja kérdezni, módosítani tudja ezeket az attribútumokat, akár csak a beépített attribútumokat.

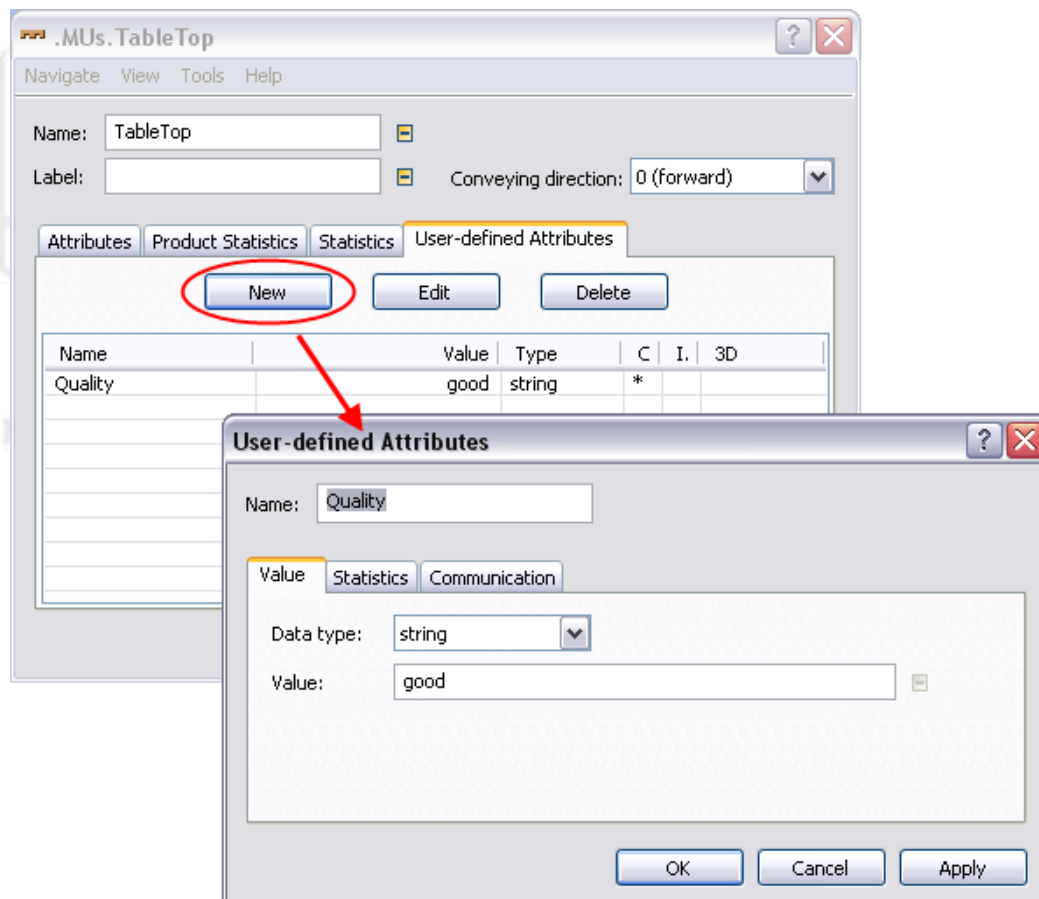
Definiálható egyéni attribútum például színre, méretre, megrendelési számra stb. Tetszőleges számú egyéni attribútum definiálható.

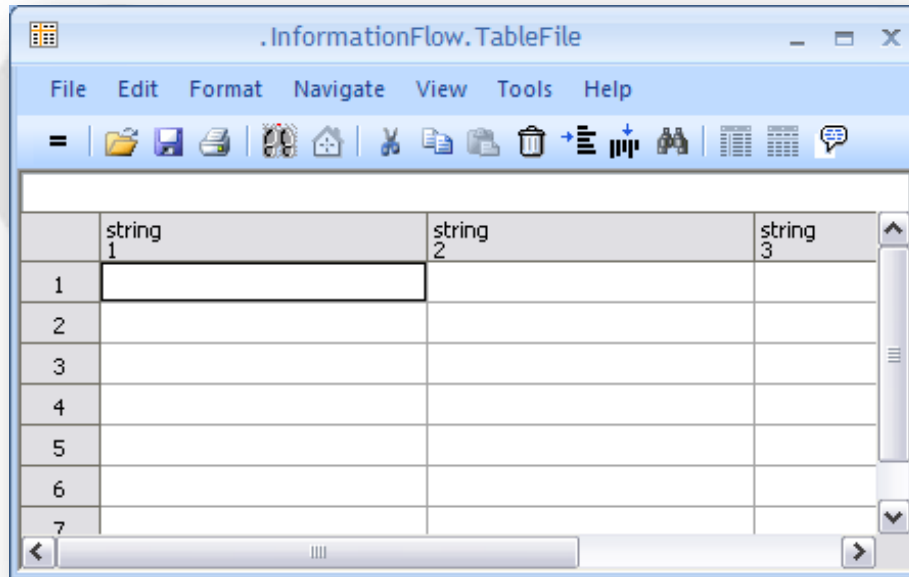
Az egyéni attribútumnak megadható a neve (**Name**), az adattípusa (**Data Type**), és az értéke (**Value**). Az érték hozzárendelhető az egyéni attribútum készítésekor, vagy módosítható a szimuláció futása alatt is.




Az egyéni attribútumok a **Custom Attributes** fülön definiálhatók az objektum ablakában. A listamező tartalmazza a már definiált egyéni attribútumokat.

Kattintson a **New** (Új) gombra, adja meg az adattípust, az értéket és persze az attribútum nevét.





Jellemzők :

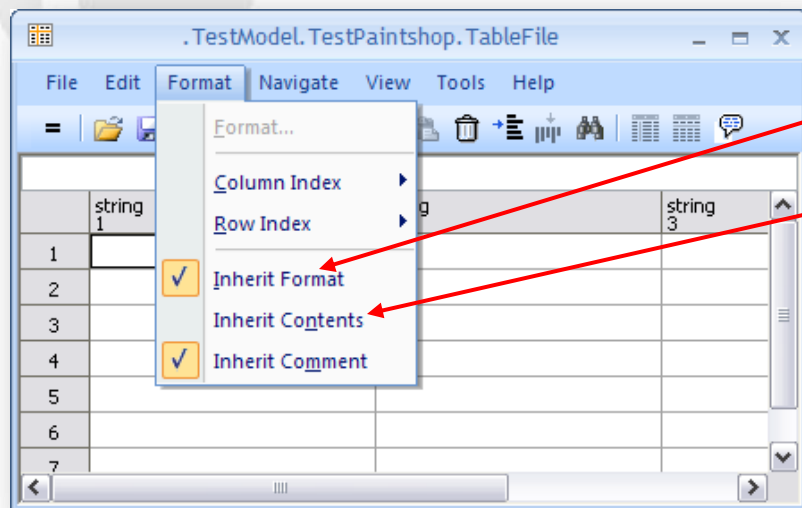
- Ikon: 
- Kapacitás: 0
- Információs áram objektum

A *TableFile* egy többoszlopos lista.

A *TableFile* az igényeknek megfelelően formázható: Megadható az **Oszlopok/sorok száma** (Number of columns/rows), az **adattípusa** (data type) az oszlopoknak, az értéktartomány, és hozzáadható Oszlop vagy Sor index.

Ezeket a beállításokat kell kikapcsolni mielőtt értékbevitel történik a táblázatba, vagy módosul az oszlopok adattípusa:

Ezek az opciók a **Format** (Formátum) menü:



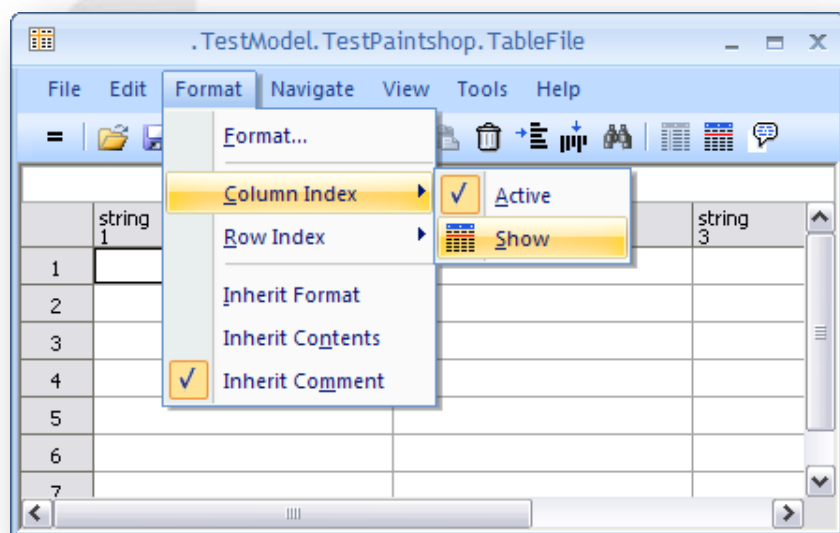
Inherit Format (Formátum öröklése) vagy

Inherit Contents (Tartalom öröklése) parancsokkal kapcsolhatók ki.

Az **Inherit Contents** opció kikapcsolása után módosítható a tábla tartalma.

Az **Inherit Format** opció kikapcsolása után módosítható a tábla formátuma. Ez az opció kikapcsolja az **Inherit Contents** opciót is.

Az oszlop- és sorindexek aktivizálásával beszédes nevek adhatók meg az oszlopokra és a sorokra, ezzel könnyebben megérthetővé téve a táblázatot. A Plant Simulation egy vastag vonallal választja el az index sort és oszlopot és a nulladik sorba, oszlopba helyezi el.



Show (mutat) vagy **Hide** (elrejt) az

Column Index (oszlopindex)

és a

Row Index (sorindex)

a menüparancsokkal lehetséges a

Format (Formátum) menüben.

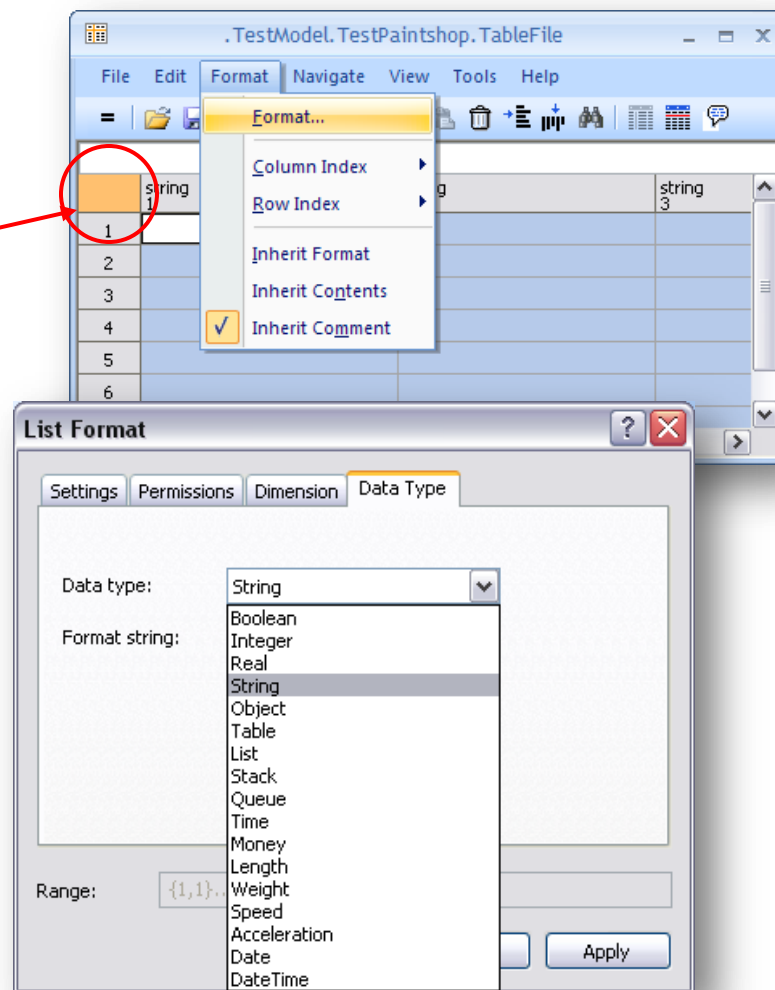
Oszlop beállításainak módosításához ki kell választani az oszlopot az oszlopindex sorban.

A teljes TableFile jellemzőinek módosításához a **Select All** (mindent kijelöl) gombra kell kattintani a TableFile bal felső sarkában.

Kiválasztás

Format > Format (Formátum)

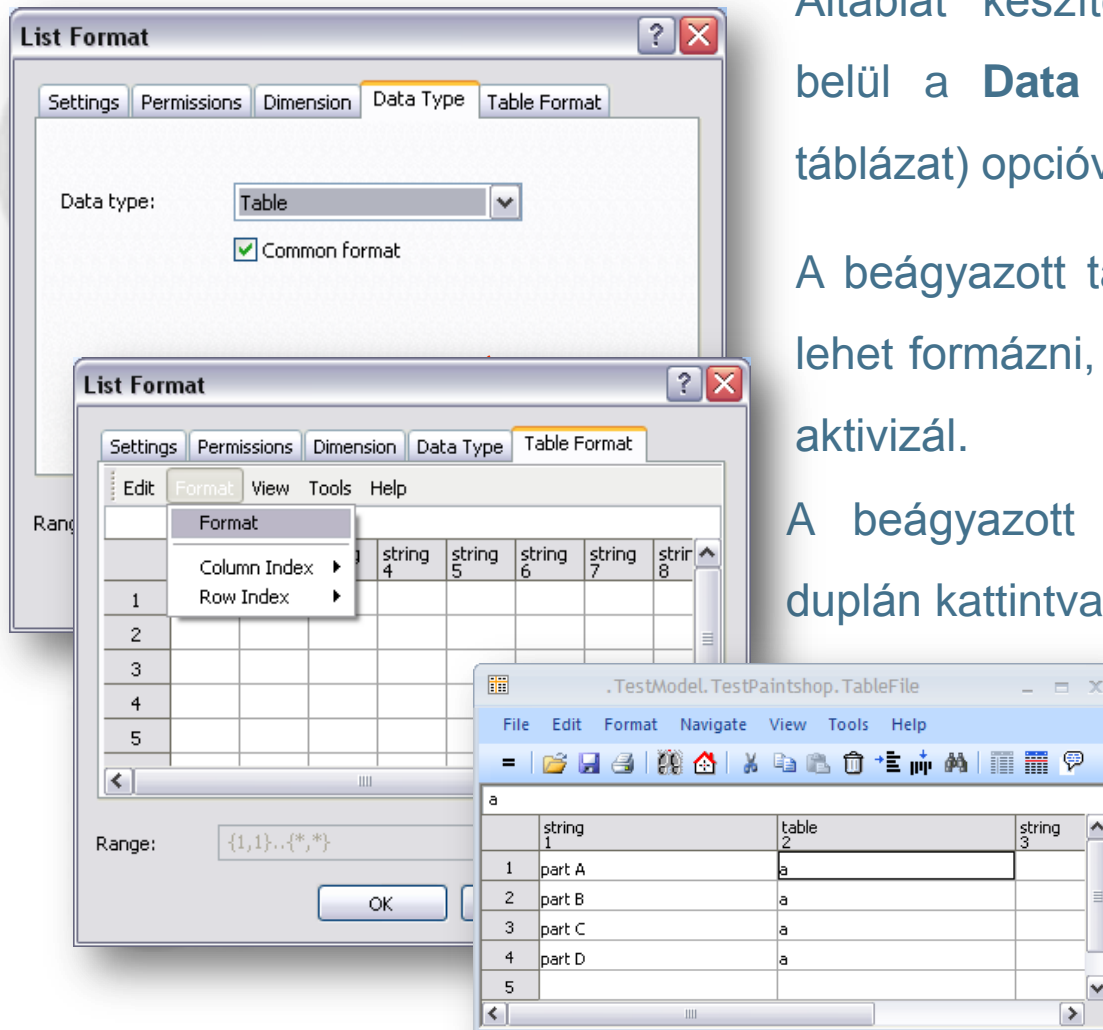
és meg lehet adni a legfontosabb jellemzőit a TableFile objektumnak.

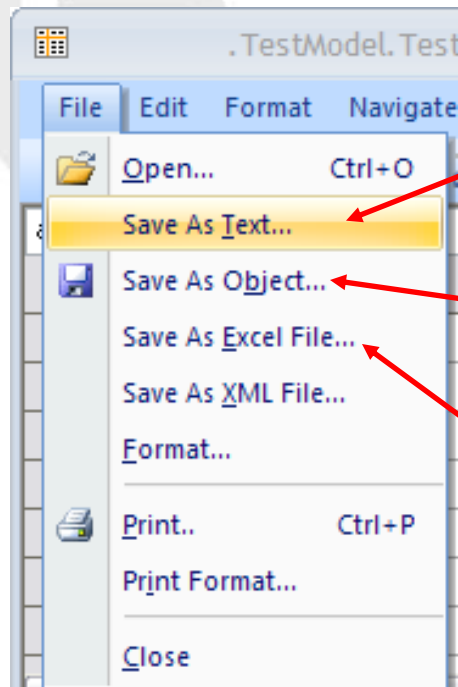


Altáblát készíteni a *TableFile* objektumon belül a **Data type > Table** (Adattípus > táblázat) opcióval lehet a **Data type** fülön.

A beágyazott táblázatokat a **TableFile** fülön lehet formázni, amit a Plant Simulation ekkor aktivizál.

A beágyazott táblát a *TableFile* cellára duplán kattintva lehet megnyitni.



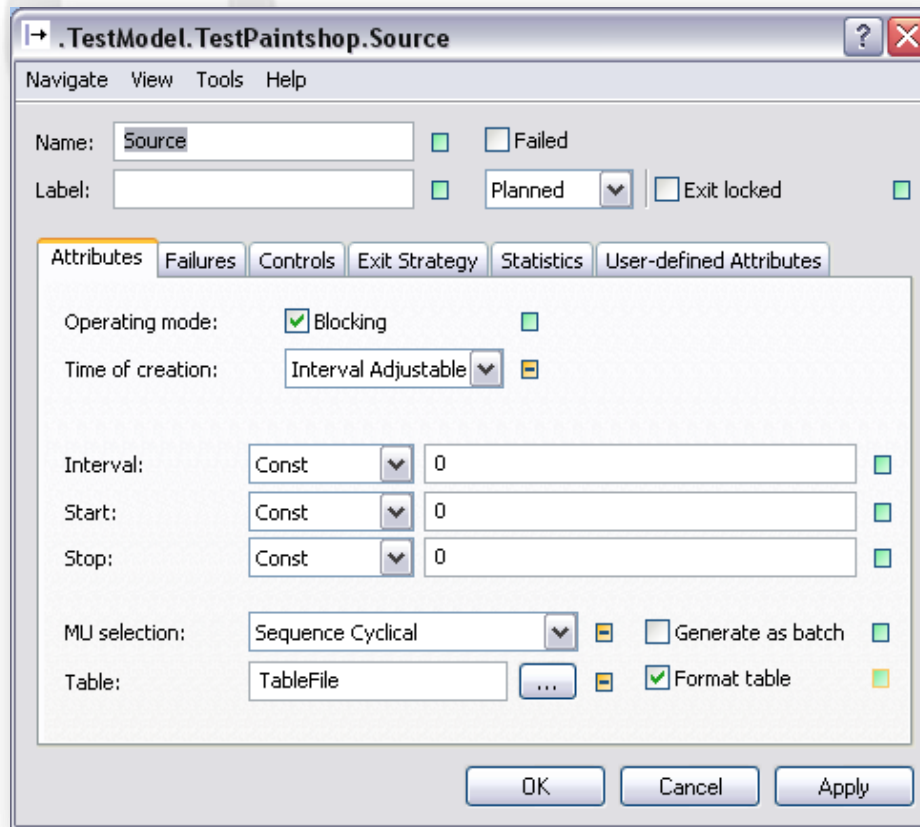


A táblázatban lévő adat tárolható szövegfájlban, ami más alkalmazásokba beolvasható.

A teljes táblázat tárolható objektumként, és másik szimulációs modellben felhasználható.

Adatok Microsoft Excel formátumban történő importálása és exportálása támogatott.

A Source objektumban az **MU selection** > **Sequence** vagy **MU selection** > **Sequence cyclical** opciót kiválasztva:



- Adja meg az útvonalát *TableFile* objektumnak, ahol a különböző MU típusok meg vannak adva, a neveikkel és az attribútumaikkal.
- Válassza ki a **Format table (táblázat formázása)** opciót, akkor a Plant Simulation automatikusan formázza a táblázatot.

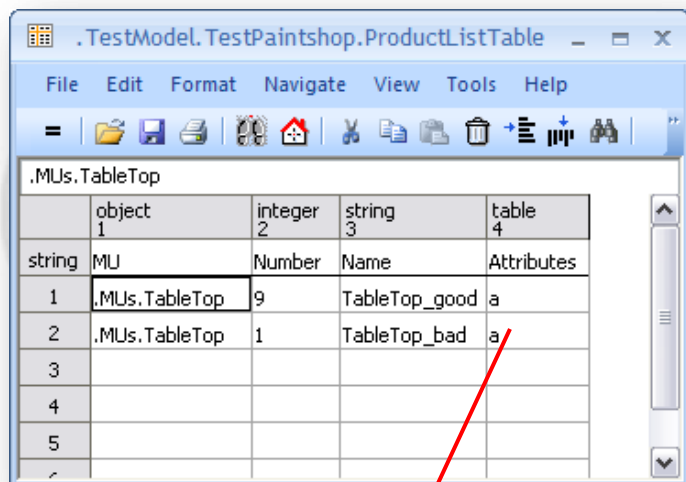
Nyissa meg a *TableFile* objektumot és adja meg vagy válassza ki a következőket:

MU: Húzza az MU típust az MU mappából a Class Library -ből az MU cellába.

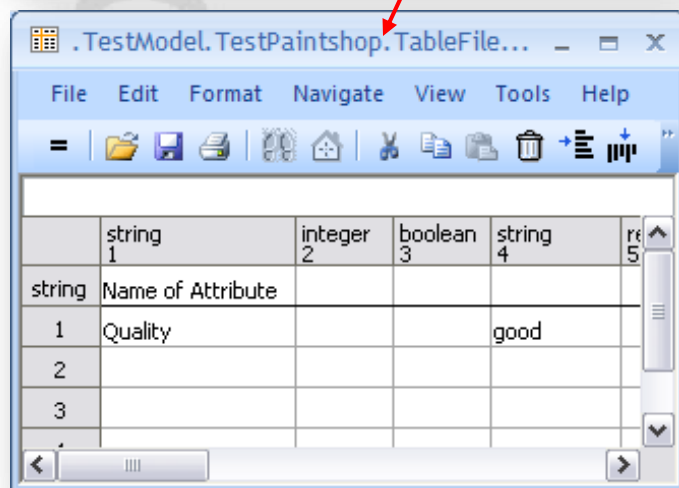
Number (szám): Egy egész szám megadása.

Name (név): Az MU-hoz itt név rendelhető.

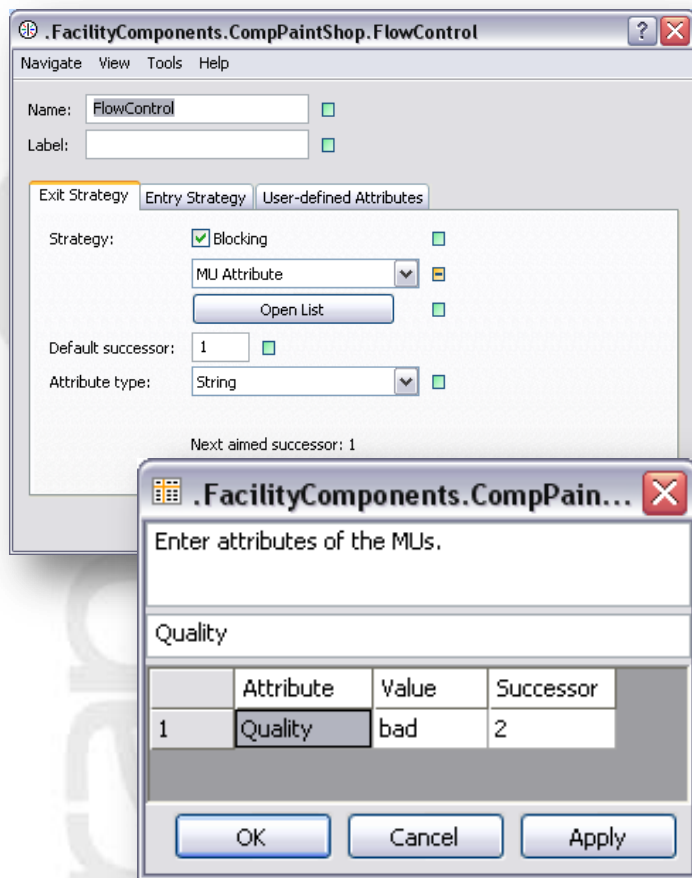
Attribute (attribútum): Az altáblázat nevét kell megadni, és megnyitni az altáblát kétszer kattintva, és közben a **Shift** billentyűt nyomva. Meg lehet adni a beépített és az egyéni attribútumok nevét, értékét és adattípusát. Az altábla neve bármi lehet, az attribútum nevét az első oszlopban, az értéket a negyedikben kell megadni.



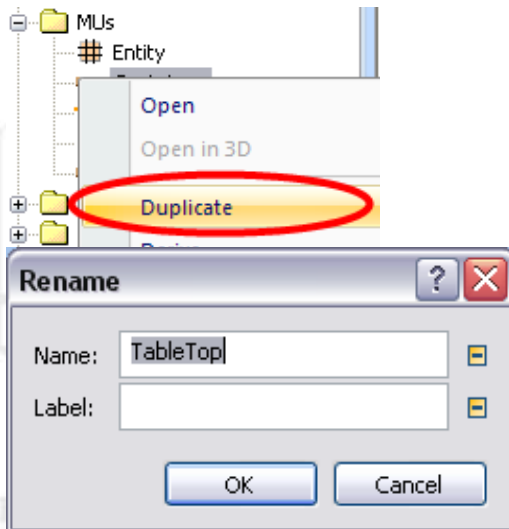
	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attributes
1	.MUs.TableTop	9	TableTop_good	a
2	.MUs.TableTop	1	TableTop_bad	a
3				
4				
5				



	string 1	integer 2	boolean 3	string 4	re 5
string	Name of Attribute				
1	Quality			good	
2					
3					

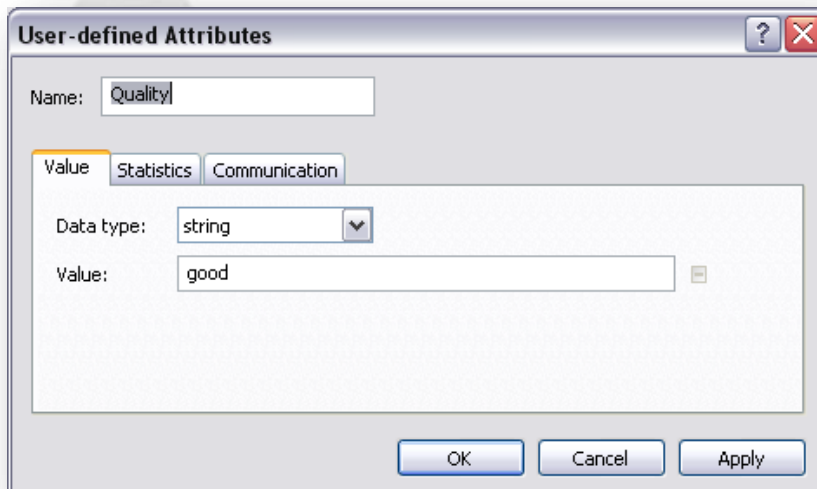


- Válassza ki az **Exit Strategy > Strategy > Attribute** értéket, és a *FlowControl* objektum az MU-kat a megadott attribútum értékének megfelelően továbbítja.
- Ez után ki kell választani az egyéni attribútum adattípusát az **Attribute type** listából, és **Apply** (Alkalmaz).
- Kattintson az **Open List** (lista megnyitása) gombra, és adja meg a keresendő attribútum nevét, és értékét, valamint a kilépési ág sorszámát.
- Az alapértelmezett kilépési ág az 1-es sorszámú. Ha az attribútum ezzel a névvel nem létezik, akkor a Plant Simulation az MU-t az 1-es vagy a szövegmezőben megadott (**Default successor**) sorszámú kilépési ágra küldi tovább.



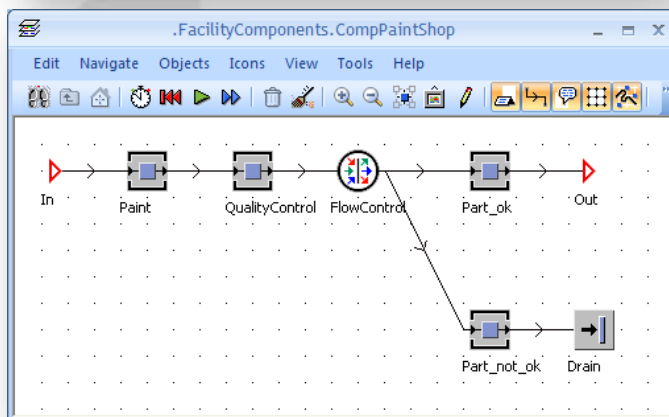
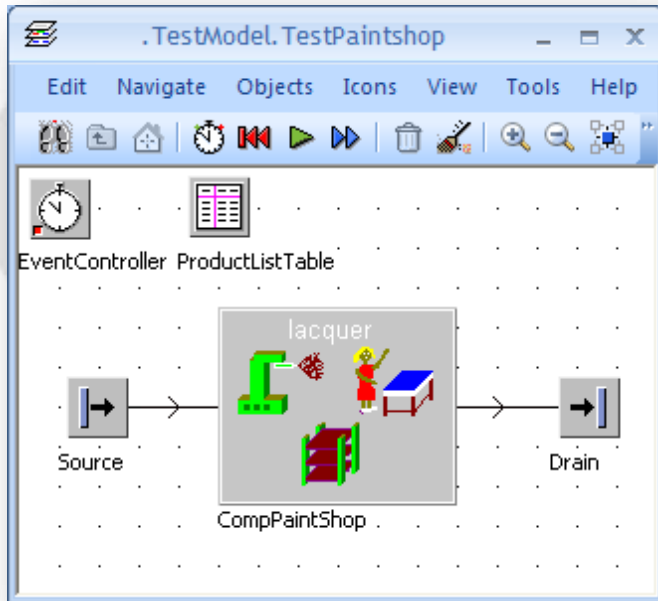
Új MU típus létrehozása az asztallapok számára:

- Container -> Duplicate
- Rename -> TableTop
- Új egyedi jellemző felvétele a TableTop MU - ra, Quality néven, string típus, good alapérték.



Új MU típus létrehozása az asztallábak számára:

- Entity → Duplicate
- Rename → TableLeg



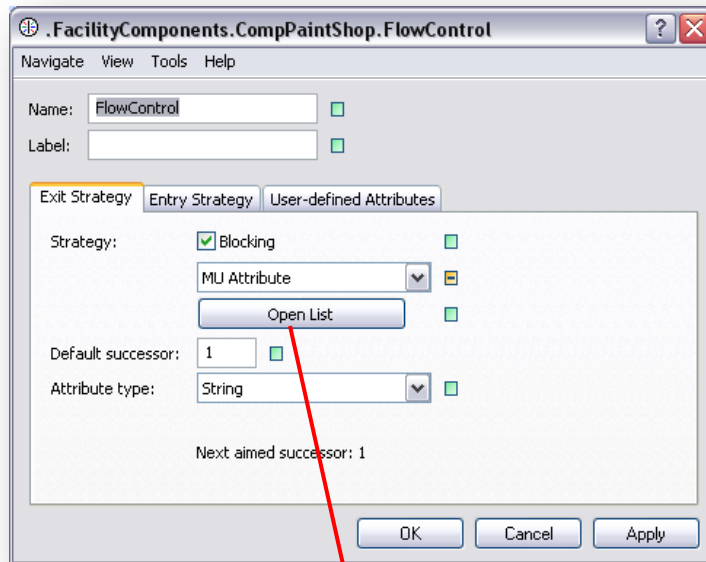
A festőüzem az asztallapokat különböző színűre festi. A hibás festésű asztallapokat kiveszi a gyártásból és megsemmisíti.

A Source (a *TestFrame Frame*-ben) hozza létre az asztallapokat jó és rossz festésekkel.

A festőüzem ellenőrzi a minőséget és a megfelelő állomásra mozgatja tovább az MU-kat.

1. Készítsen egy új *Frame* -et a *FacilityComponents* mappában és legyen a neve: **CompPaintshop**. Helyezze el az ábrán látható objektumokat a *Frame*-ben.

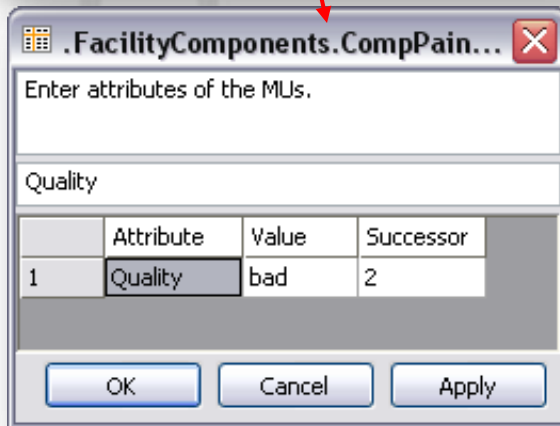
Figyeljen az öröklődésre: a *CompPaintshop Frame* -et csak az osztályban módosítsa, ne a példányban!

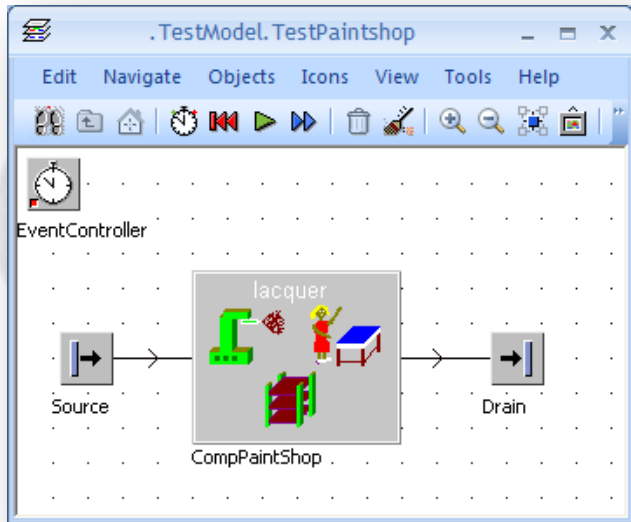


2. Válassza ki a **Strategy > Attribute** opciót és a **Data type string** opciót a *FlowControl* objektumban. Majd kattintson az *Apply* gombra, és az **Open List** gombra.

3. Adja meg a **Quality** attribútumot a táblázatban, és az értékét és a kimeneti ág sorszámát is, ahol az MU mozogni fog. **Az érték kis-nagybetű érzékeny!**

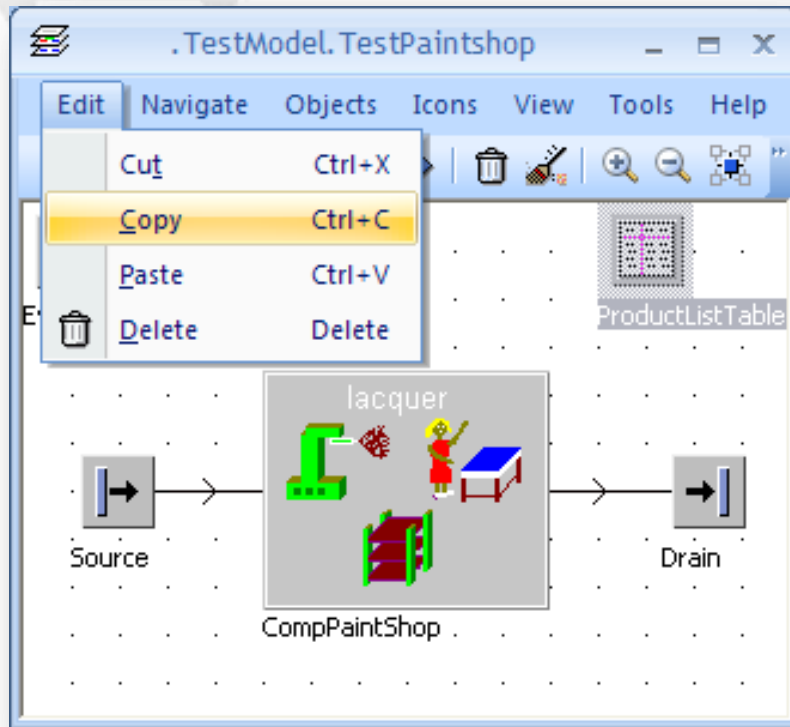
4. Nyissa meg az *Icon Editor*-t a *CompPaintshop Frame*-re, rajzoljon egy új ikont és illesszen be néhány animációs pontot.





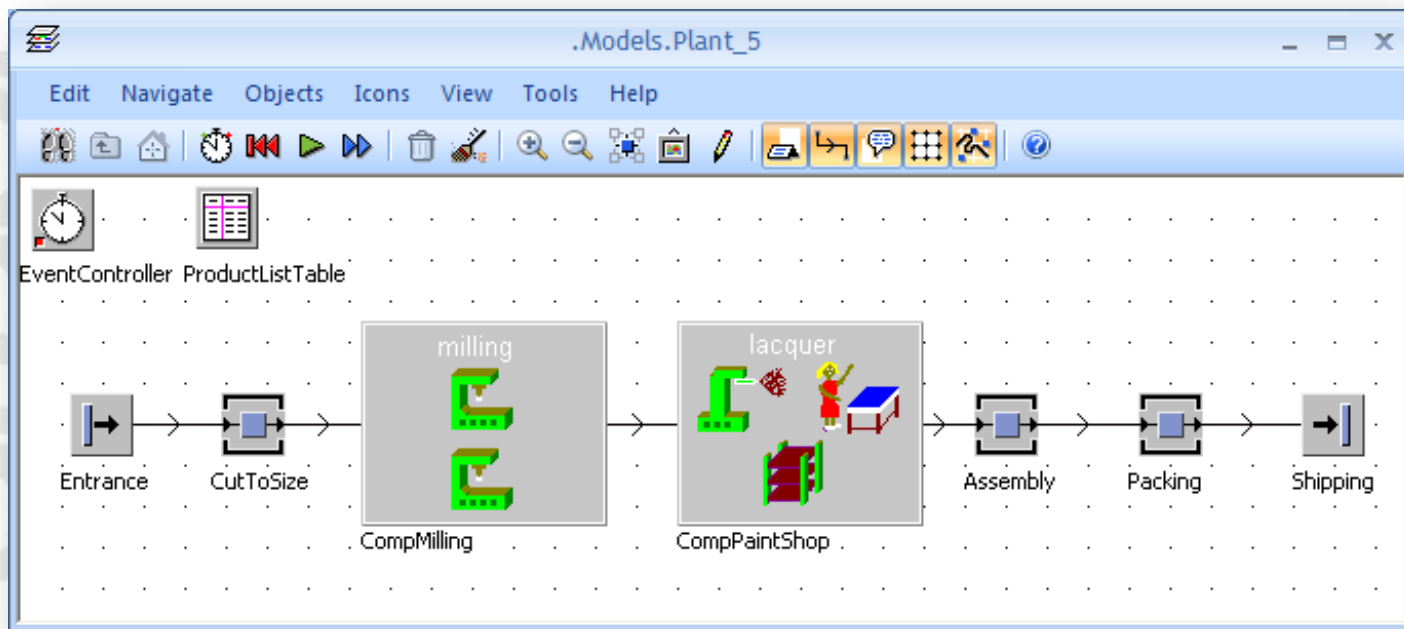
1. Készítsen egy teszt környezetet a **TestModel** mappában, legyen a neve: *TestPaintshop*. Illessze be a következőket: *Source*, *Drain*, *EventController*, és a *CompPaintshop Frame* -et a *Class Library* -ből és egy *TableFile* objektumot. Kösse össze az anyagáram objektumokat.
2. Nyomja le az **F2** billentyűt és nevezze át a *TableFile* objektumot *productListTable* névre.
3. Válassza ki az **MU selection > Sequence Cyclical** opciót a *Source* objektumban és válassza ki ezt a *TableFile* -t. Válassza ki a **Format table** opciót. Adja meg a korábbi értékeket (161. dia)
4. Figyeljen arra, hogy a Plant Simulation különbséget tesz a kisbetűs és NAGYBETŰS szövegértékek között, amikor a quality attribútum értékét megadja az legyen: **bad** és **good**!
5. Ellenőrizze le a *Frame* funkcióit.

A *productListTable* objektumot újra fel szeretnénk használni, más leckékben mivel az tartalmazza a kibocsátási sorrendjét a *Source* objektumnak, ezért bemásoljuk ezt a táblázatot a *Class Library*-be.



1. Válassza ki a *TableFile*-t és másolja a Vágólapra a **Ctrl+C** billentyűzetkombinációval.
2. Másolja be a táblázatot ahova kívánja a *Class Library* mappájának helyi menüjében lévő paranccsal.

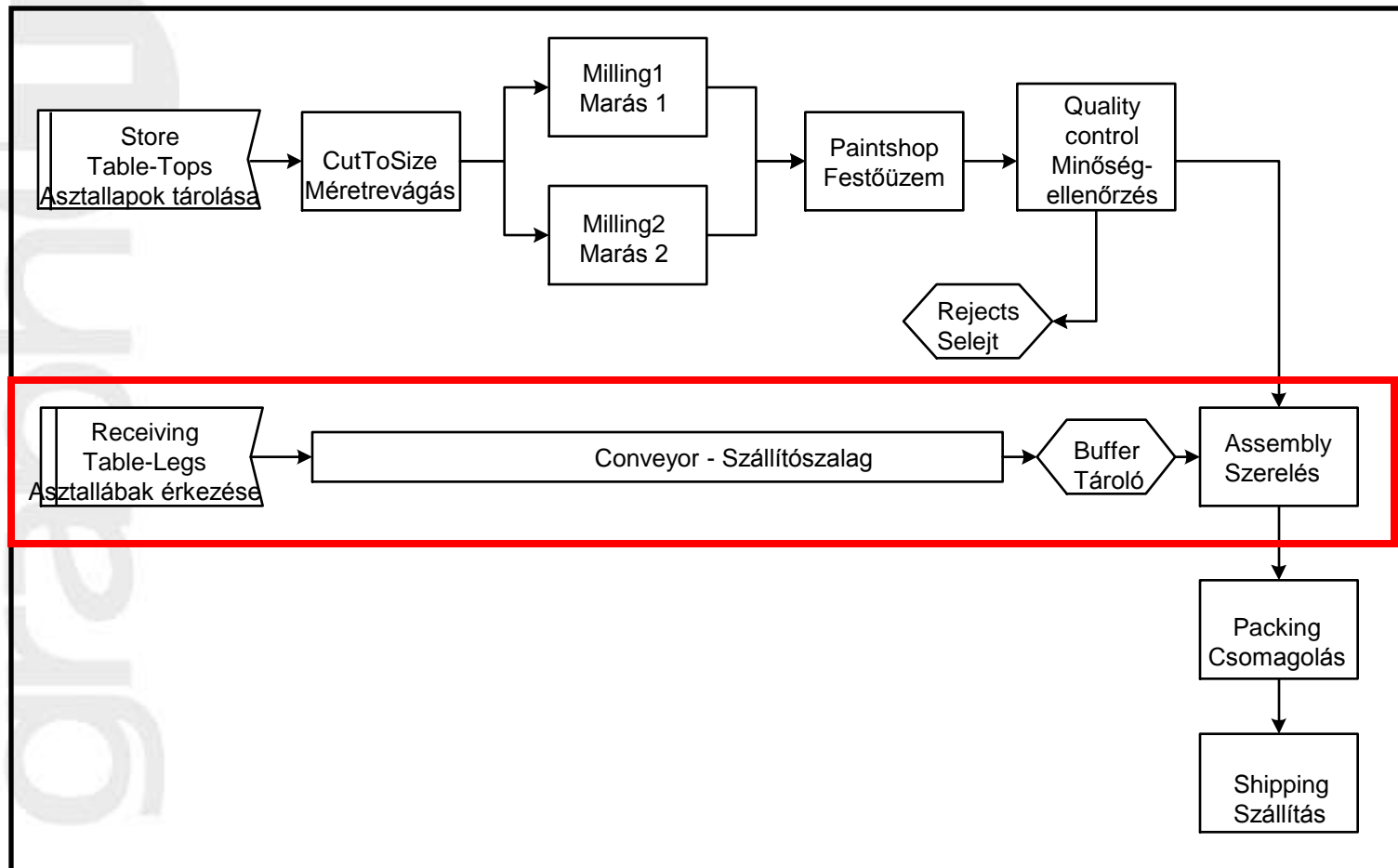
1. Készítsen másolatot a **Plant_4** objektumból és legyen a neve: **Plant_5**.
2. Cserélje ki a *Paintshop SingleProc* objektumot a *Paintshop Frame* -mel.
3. Illessze be a *productListTable* objektumot, ami a kibocsátási szekvenciát tartalmazza, és adja meg ezt a táblázatot, mint a *Source sequence cyclical* típusú kibocsátása.






12. fejezet

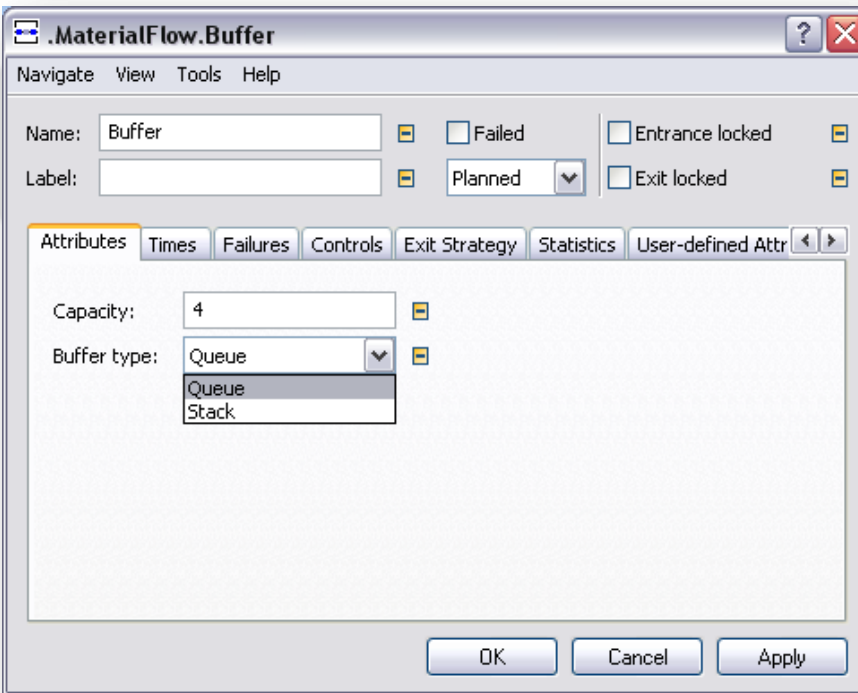
Anyagáram objektumok és olyan objektumok, amelyek több, mint egy elem kapacitásúak



Jellemzők:

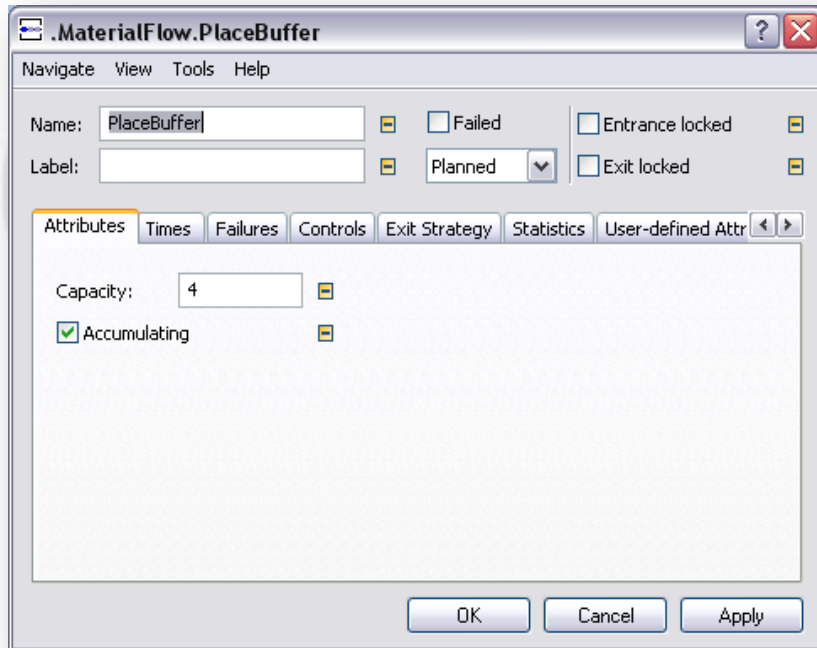
- Ikon: 
- Kapacitás: tetszés szerinti, nem mátrixos
- Kapacitás orientált anyagáram objektum

A *Buffer* a következő állomásra mozgatja az MU-kat **FIFO** stratégiával (QueueFile) vagy **LIFO** stratégiával (StackFile).



Nem lehet a *Buffer* különálló celláihoz hozzáférni.

A *Buffer* nagy számú MU fogadására szolgál. A megadott feldolgozási idő azt adja meg, amíg egy MU kilép a *Buffer* objektumon.

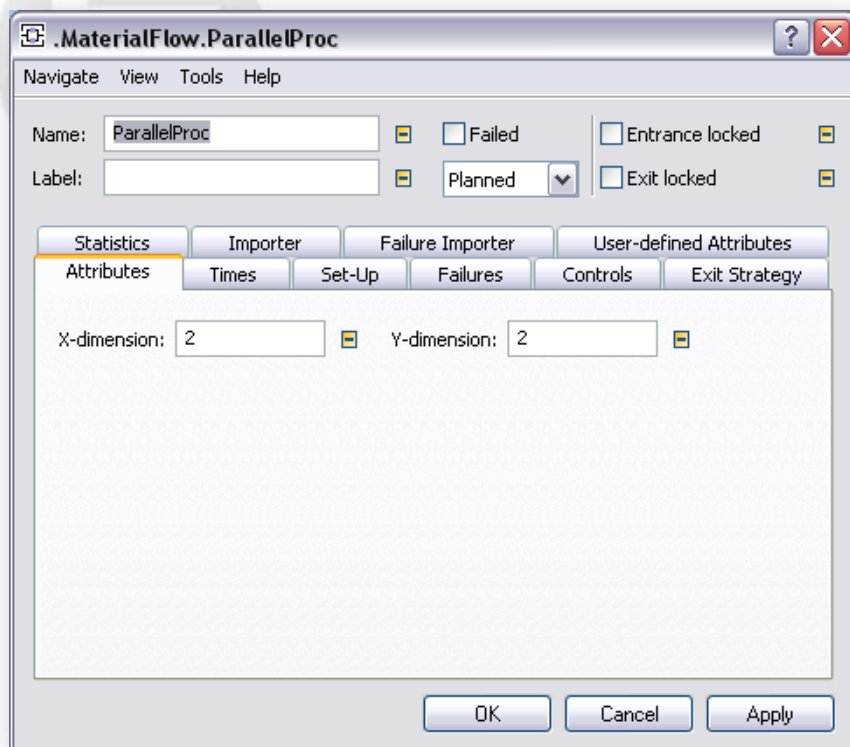


Jellemzők :


- Ikon: 
- Kapacitás: tetszés szerinti
- Helyorientált anyagáram objektum

A *PlaceBuffer* több sorban lévő cellával rendelkezik. Tetszés szerinti számú MU-t tud feldolgozni és tárolni.

A feldolgozási időt a **teljes** *PlaceBuffer* objektumra lehet megadni, nem külön a cellákra. Az objektum felosztja a megadott feldolgozási időt a cellák között, meghatározva azt, hogy milyen módon halad az MU végig az összes cellán. Az MU-k nem előzhetik meg egymást. A feldolgozási idő lehet nulla perc is.



Jellemzők :

- Ikon: 
- Kapacitás: tetszés szerinti, mátrix alapú
- Helyorientált anyagáram objektum

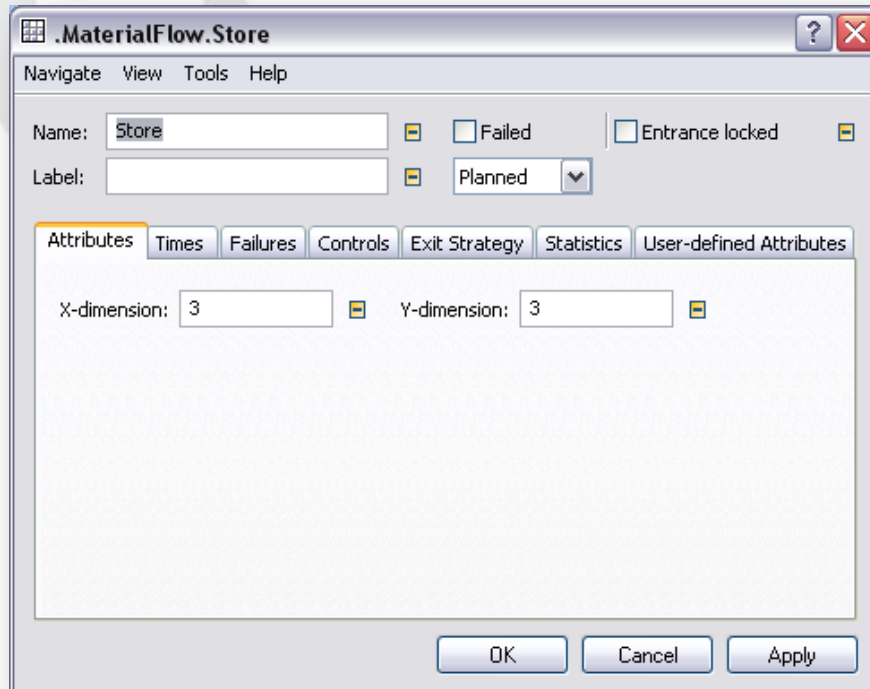
A *ParallelProc* több cellával rendelkezik, hogy az MU-kat párhuzamosan dolgozhassa fel.

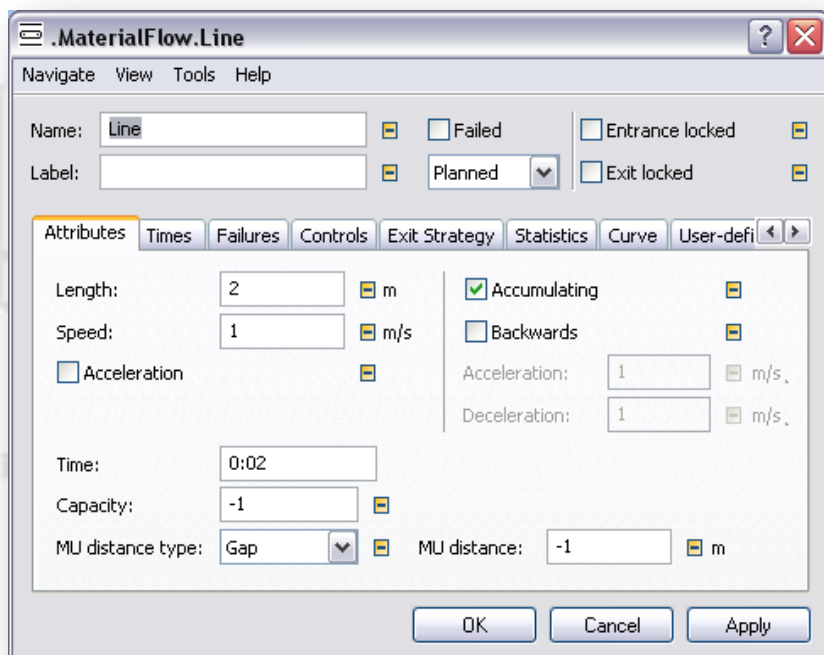
A feldolgozási időt bármikor meg lehet változtatni. Ez nem érinti a *ParallelProc* másik celláján lévő MU-kat. Ennek következtében viszont az MU-k megelőzhetik egymást.

Jellemzők :

- Ikon: 
- Kapacitás: tetszés szerinti, mátrix alapú
- Helyorientált anyagáram objektum

Az MU-k a *Store* objektumon maradnak, amíg egy metódus (*Method*) el nem távolítja azokat. A Plant Simulation az érkező MU-kat tetszőleges szabad cellára helyezi el az objektumon. Ha megtelt (failure), akkor a *Store* nem fogad el több MU-t.





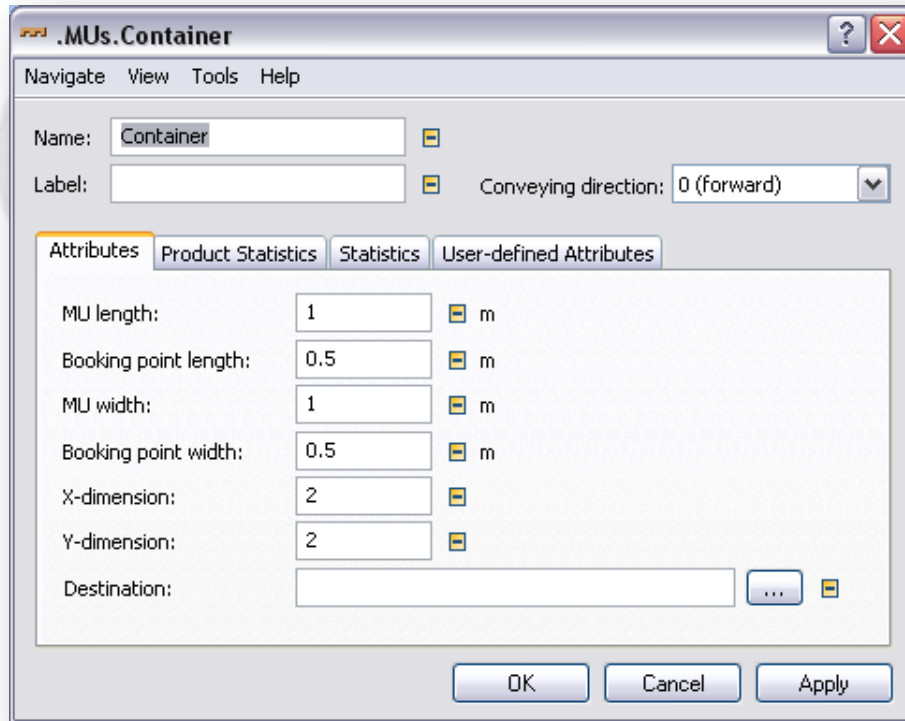
Jellemzők :

- Ikon: 
- Kapacitás: tetszés szerinti, hossz orientált
- Hosszorientált anyagáram objektum


A *Line* egy szállítószalagot reprezentál. A hossz (length) és a sebesség (speed) határozza meg azt az időt, amíg az MU végighalad rajta.

Az MU-k feltorlódását a *Line*-on az **Accumulating** opció bekapcsolásával lehet engedélyezni.

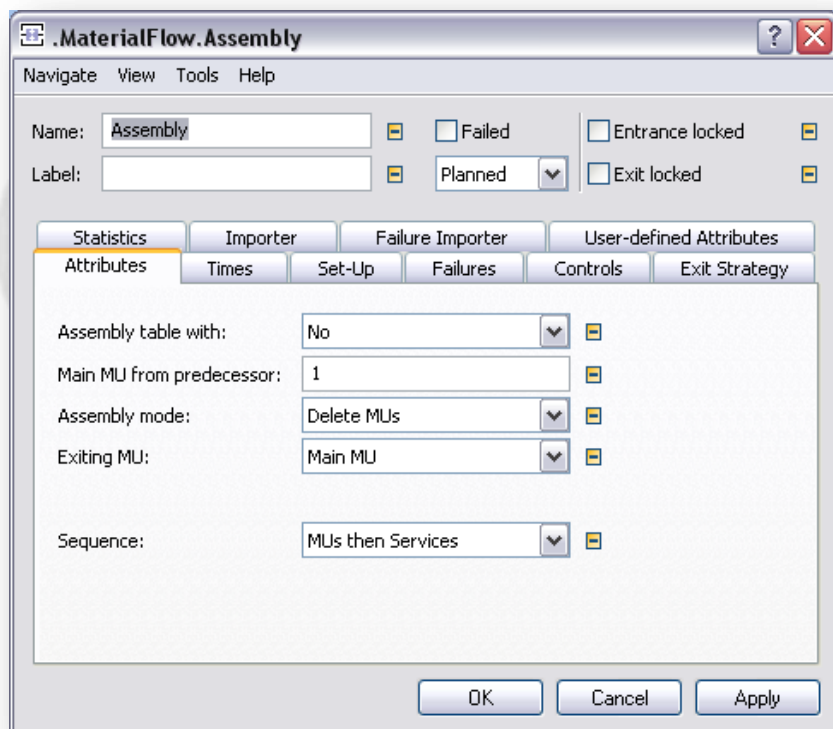
Ez lehetővé teszi az MU-k számára, hogy egymás mögött feltorlódjanak, ha a kilépés nem lehetséges. Alapértelmezésben a *Line* kapacitásának nincs határa, ezt a -1 érték jelöli. Megadható más számérték is, és ekkor a *Line* nem használja fel a teljes hosszt.




Jellemzők :

- Ikon: 
- Kapacitás: tetszés szerinti, mátrix alapú
- Helyorientált anyagáram objektum

A *Container* egy MU, amely elemeket (*Entity*) szállít és más *Container* objektumokat. Ez egy palettát reprezentál. A *Container* használható elemként (*Entity*) is. A *Container*en lévő *Container* szállíthat más elemeket vagy konténereket.



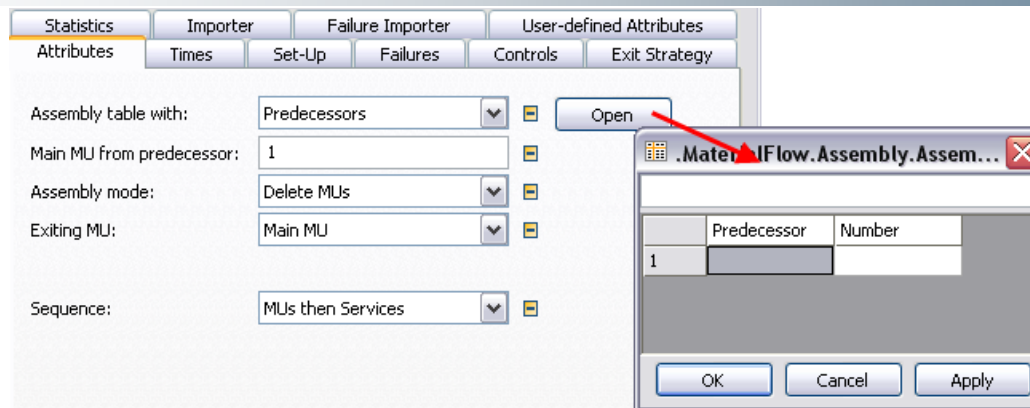
Jellemzők :

- Ikon: 
- Kapacitás: 1 fő elem, és szerelvény alkatrészek
- Helyorientált anyagáram objektum

Az *Assembly* a szerelvény alkatrészeket adja hozzá a fő elemhez. A szerelvény alkatrészeket vagy hozzáadja a fő MU-hoz, vagy törli azokat az **Assembly Table** - ben megadott értéktől függően.

Meg kell adni az **Assembly table** - ben, hogy mely alkatrészek kapcsolódnak hozzá a Main MU (fő MU) objektumhoz.

Az **Assembly mode** listából kiválasztható, hogy a szerelvény alkatrészek (mounting part) törölve legyenek, vagy a fő MU elemre legyenek mozgatva.



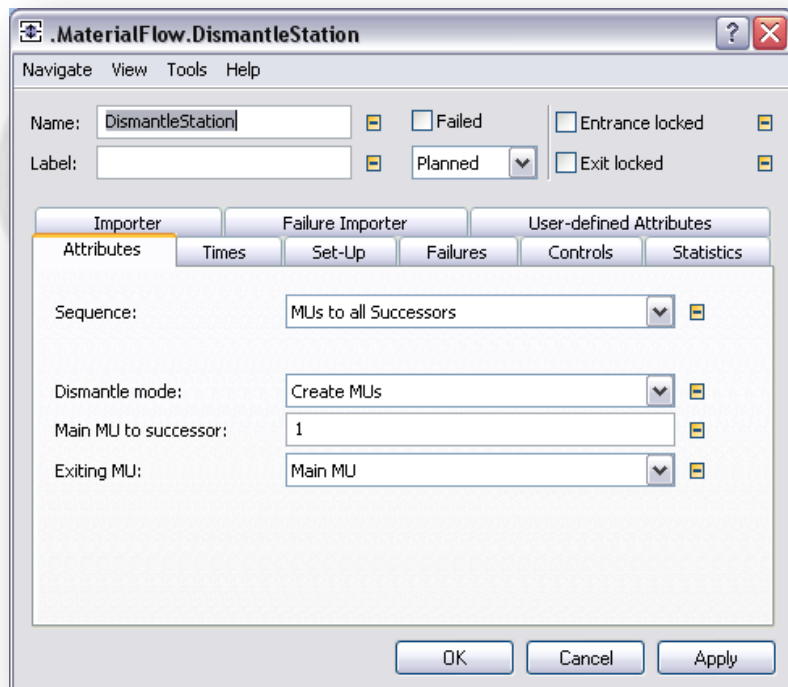
- **Assembly table with** (Szerelési tábla): adja meg a **Predecessors** (bemenő ágak) vagy **MU Types** (MU típusok) opciót és a *TableFile* celláiba írja be a bemenő ágakat vagy az MU típus neveket.
- **Main MU from predecessor** (fő MU bemenő ága): a bemenő ágnak a sorszáma (**Tools > Options > Modeling > Show Predecessor**).
- **Assembly mode** (szerelési mód): válassza ki a **Delete MUs** opciót a szerelés után a szerelvény alkatrészek törléséhez, az **Attach MUs** opciót a szerelvény alkatrészek rárakásához a fő MU - ra.
- **Exiting MU** (kilépő MU): a **Main MU** (fő MU) vagy **New MU** (új MU) továbbmozgatása a kimenő ágon.

Jellemzők :

- Ikon: 
- Kapacitás: 1 fő elem szerelvény alkatrészekkel együtt
- Hely-orientált anyagáram objektum

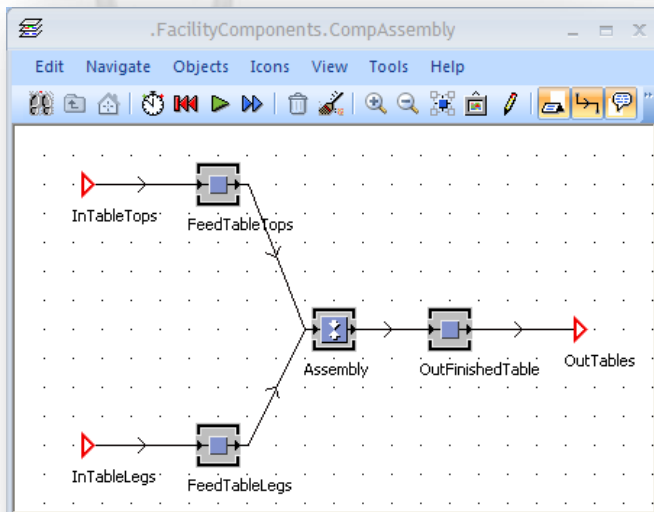
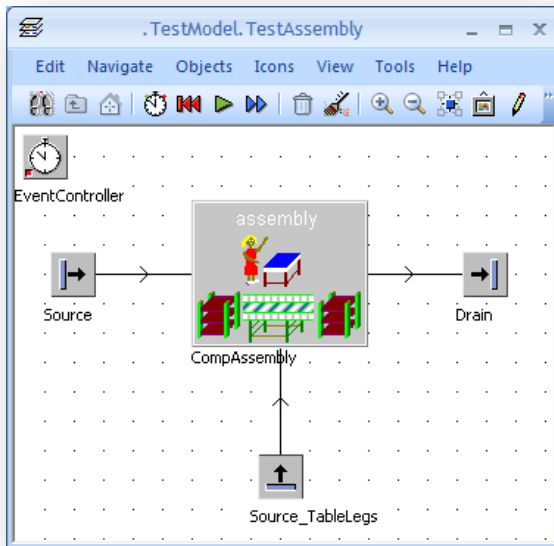
A *DismantleStation* objektum eltávolítja a szerelvény alkatrészeket a fő MU-ról, vagy újakat készít. Szétszerelési folyamatok modellezésére szolgál.

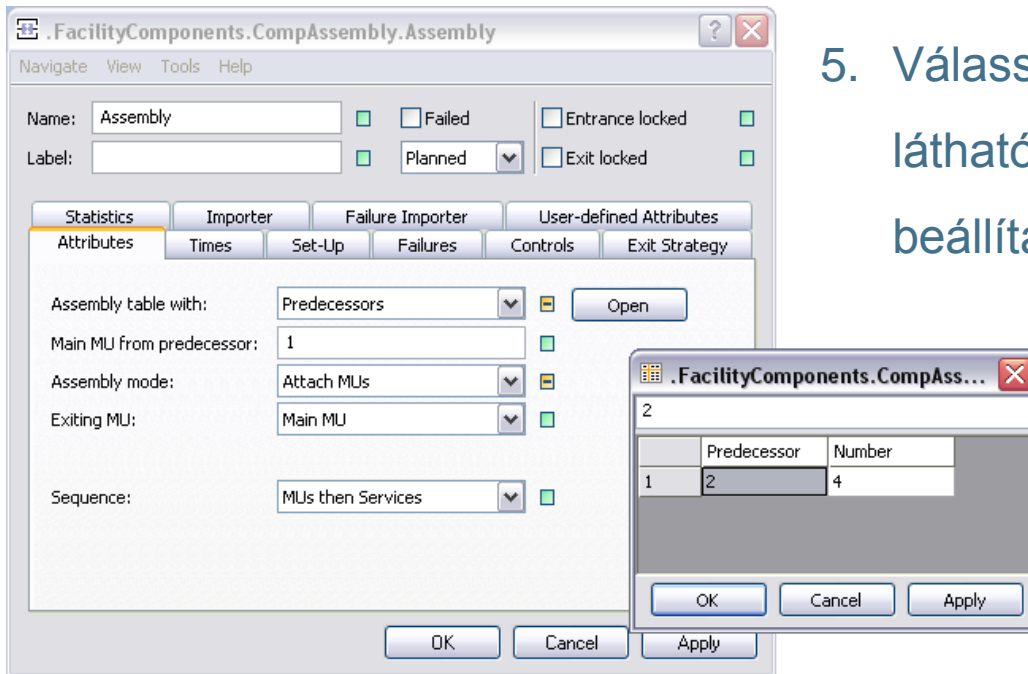
Az **Attributes** fülön lehet megadni, hogy a *DismantleStation* objektum hogyan hajtja ezt a feladatot végre.



Az asztallábak az asztallapokhoz a **CompAssembly** *Frame*-ben adódnak hozzá.

1. Készítsen egy új *Frame* - et a *FacilityComponents* mappában *CompAssembly* névvel.
2. Illessze be a bal oldali ábrán látható objektumokat. A feldolgozási ideje a *Assembly* objektumnak legyen 10 perc.
3. Az *InTableTop Interface* beviszi az asztallapokat az állomásba, az *InTableLeg Interface* pedig az asztallábakkal teszi ugyanezt.
4. Válassza ki a **Tools > Options > Show Predecessor** opciót a menüből.



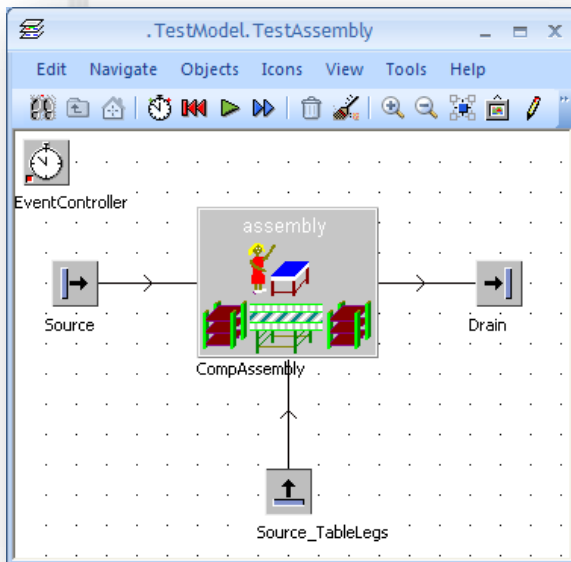
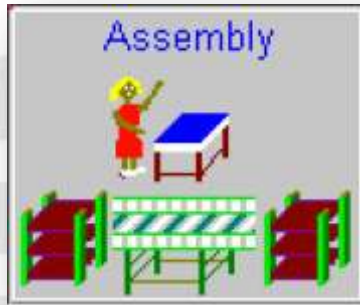


5. Válassza ki és adja meg a bal oldalon látható ábrának megfelelően a beállításokat az *Assembly* objektumra:

Először adja meg, hogy melyik *Predecessor* - ból kapja az *Assembly* állomás a fő MU-kat, ebben az esetben az asztallapokat.

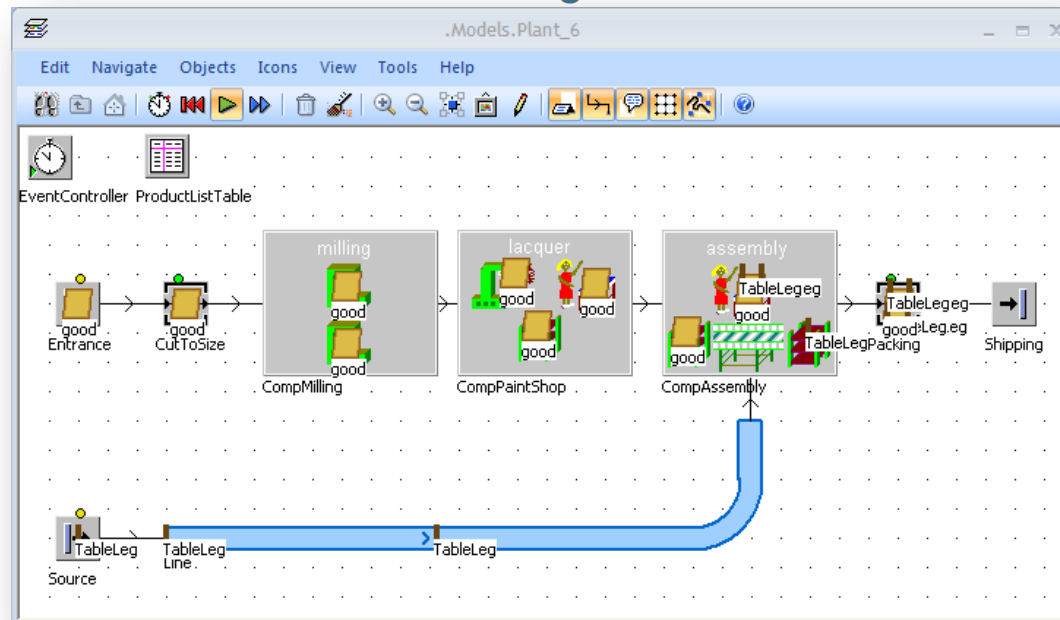
Adja meg a predecessor sorszámát, amelyik az asztallábakat adja az asztallaphoz.

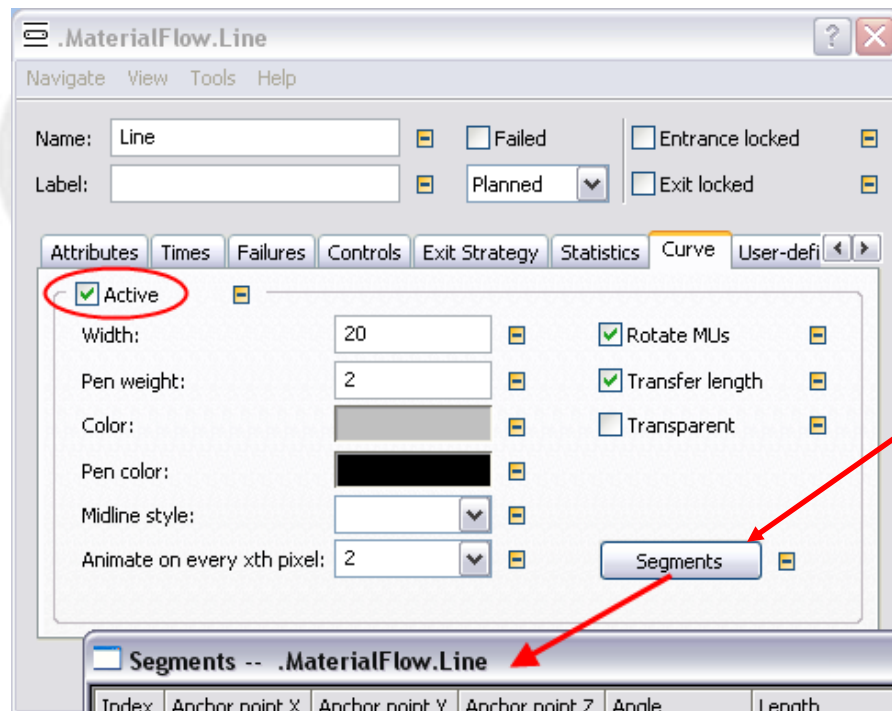
Válassza ki az **Assembly mode > Attach MUs** opciót, így a fő MU lép ki az *Assembly* állomásról.



1. Rendeljen hozzá a *CompAssembly Frame*-hez új ikont.
2. Készítsen egy új *Frame*-et a *TestModel* mappában, legyen a neve: *TestAssembly*.
3. Illessze be a következőket: két *Source*, *Drain*, *EventController* és az *Assembly Frame*, majd kösse össze az objektumokat. Győződjön meg arról, hogy a *Source* objektumok a megfelelő típusú MU-kat hozzák létre.
4. Ellenőrizze az *Assembly* funkcióit.

1. Készítsen másolatot a **Plant_5** objektumból, és legyen a másolat neve: **Plant_6**.
2. A *TestAssembly Frame* funkcionalitásának tesztelése után helyezze azt be az *Assembly SingleProc* helyére.
3. Illessze be a *Source* objektumot (állítsa be a készítendő MU-k típusát), amelyik az asztallábakat készíti, és egy *Line* komponenst a *Frame*-hez. Legyen a *Line* hossza 2 m és sebessége 0,5 m/s.





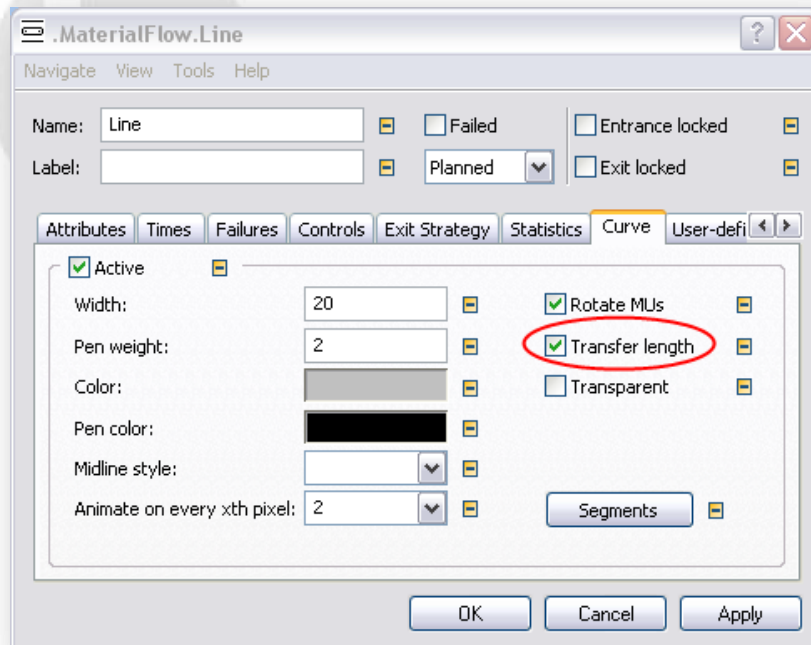
A szállítószalag görbeként, nem csupán objektumként történő beillesztéséhez a görbe módot kell aktivizálni.

A **Segments** (szegmensek) táblázat mutatja a görbe szegmenseinek paramétereit.

The Segments dialog box displays a table with the following data:

Index	Anchor point X	Anchor point Y	Anchor point Z	Angle	Length	Radius	Center X	Center Y
0	0	0	20	0.000000	40.000000			
1	40	0	20	0.000000	0.000000			

At the bottom of the dialog are buttons for OK, Cancel, and Apply.



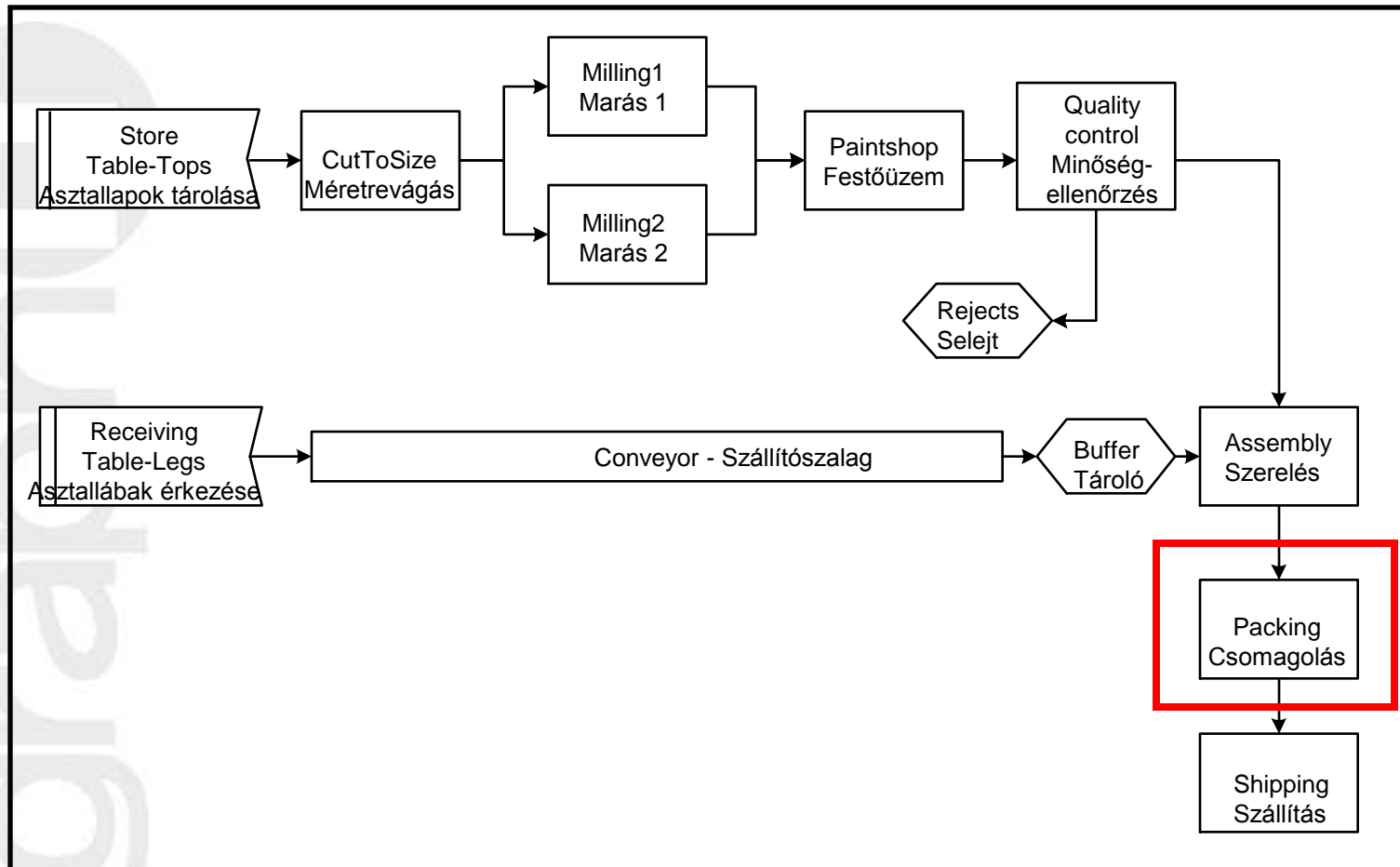
A szállítószalag a saját hosszát a görbék hosszából számítja.

Ha ez nem megfelelő számunkra, akkor a **Curve** fülön ki lehet kapcsolni a számítását.



13. fejezet

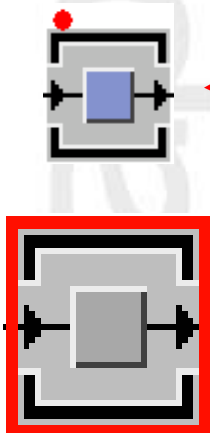
Hibák, meghibásodások



A valós élet szituációinak modellezéséhez, ahol a gépek időnként meghibásodnak, ami érinti a használhatóságukat, és a rájuk szervezhetőséget, a rendszerben lehetőség nyílik meghibásodások definiálására is.

Ez a komponens állapotát átállítja az **operational** (működő) állapotból a **failed** (hibás) állapotba. A Plant Simulation hozzáadja a meghibásodás idejét a feldogozási időhöz és az otttartózkodási időhöz.

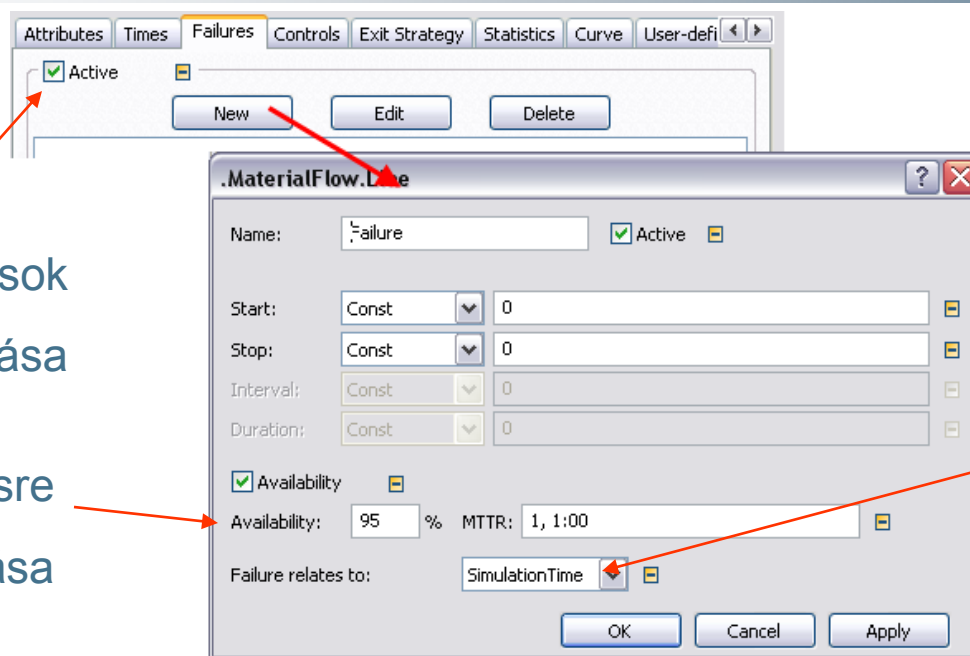
Meghibásodás minden anyagáram objektumra megadható.



Az objektumnak a bal felső részénél egy piros LED jelenik meg. Ha korábbi Plant Simulation verzióban készült modellnél szeretne az állapotokhoz ikont rendelni, akkor egy új ikont kell készíteni failed névvel. Valamint válassza ki az

Icon > Use State Icons

pontot az ikonszerkesztőben, hogy az objektum használja az állapotikonokat a LED - ek helyett.



Meghibásodások
aktivizálása

Rendelkezésre
állás megadása

Min alapuljon a
meghibásodás

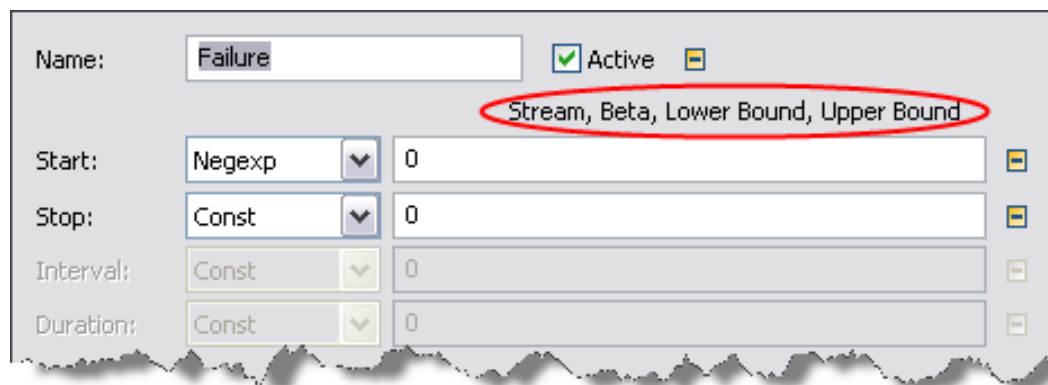
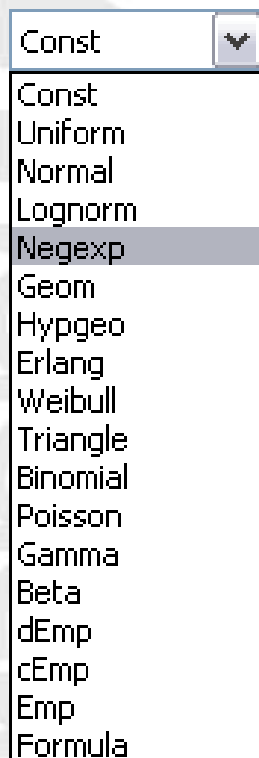
A Mean Time Between Failures (meghibásodások közötti idő) (MTBF) és a Mean Time To Repairs (javítás ideje) (MTTR) használható a meghibásodás idejének kiszámítására.

Adja meg a rendelkezésre állást százalékban. A Plant Simulation Erlang eloszlást használ az intervallumra és Weibull eloszlást az időtartamra.

Egy komponenshez tetszőleges számú meghibásodás rendelhető.

A Plant Simulation számos eloszlást nyújt a meghibásodási intervallum és a meghibásodás időtartamának megadására. Ki kell választani az eloszlást, és meg kell adni a rá jellemző paramétereket.

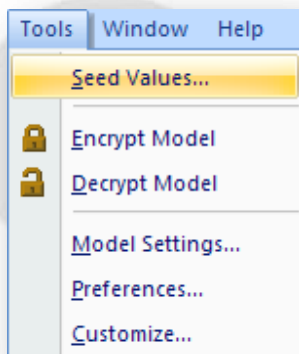
A Plant Simulation a **Start** idő szövegmező fölött mutatja, hogy milyen paramétereket kell megadni.



A screenshot of the 'Failure' configuration dialog box. The 'Name' field is set to 'Failure'. The 'Active' checkbox is checked. A red circle highlights the text 'Stream, Beta, Lower Bound, Upper Bound' above the 'Start' field. The 'Start' field is set to 'Negexp' with a value of '0'. The 'Stop' field is set to 'Const' with a value of '0'. The 'Interval' field is set to 'Const' with a value of '0'. The 'Duration' field is set to 'Const' with a value of '0'.

Válassza ki a

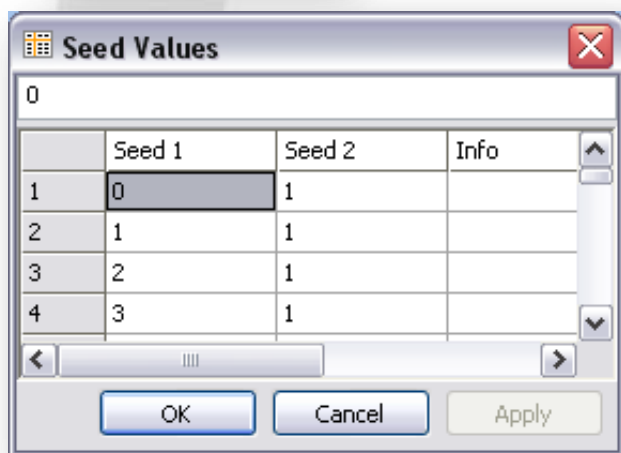
Tools > Seed Values menüpontot a főmenüből, ami megnyitja a seed values (alapszámok) táblázatot.



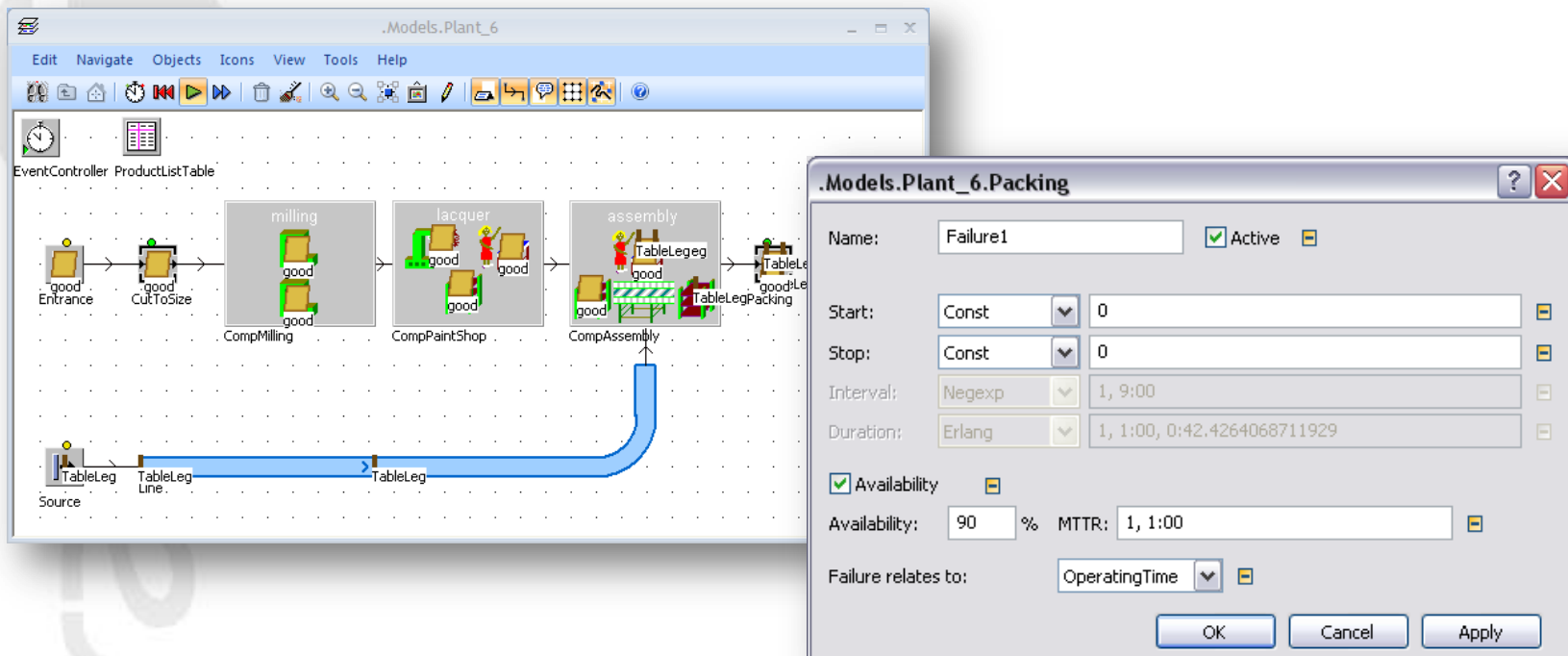
A **Seed Values** táblázat véletlenszerű értékeket tartalmaz egymás után.

Ezek mindegyike két alapszám értéket igényel, amelyek egész számok. Ezek az értékek adják meg a véletlen számok sorát, amelyet a Plant Simulation generál.

Különböző véletlenszám sorok azonos alapszámokkal azonos véletlenszám sorokat generálnak.



Meghibásodások a Plant_6 Packing modellben



Adja meg az ábrának megfelelő értékeket a szövegmezőkbe a **Failures** fülön a *Packing SingleProc* ablakában. A meghibásodások az **Operating Time**-ra (működési időre) hivatkoznak.



14. fejezet

Az EventDebugger

Az *EventDebugger* annak megtekintésére szolgál, ahogy a Plant Simulation feldolgozza a különálló eseményeket a szimuláció futása során.

A modellben lévő hibák megtalálására használható. A még fel nem dolgozott eseményeket jeleníti meg listában.

Illesszen be egy breakpoint - ot (töréspontot) az esemény előtt, és hajtsa végre az eseményt lépésről-lépésre.

Az *EventDebugger* aktivizálása:

Kattintson az *EventController* ablakában a **List** (lista) nyomógombra.

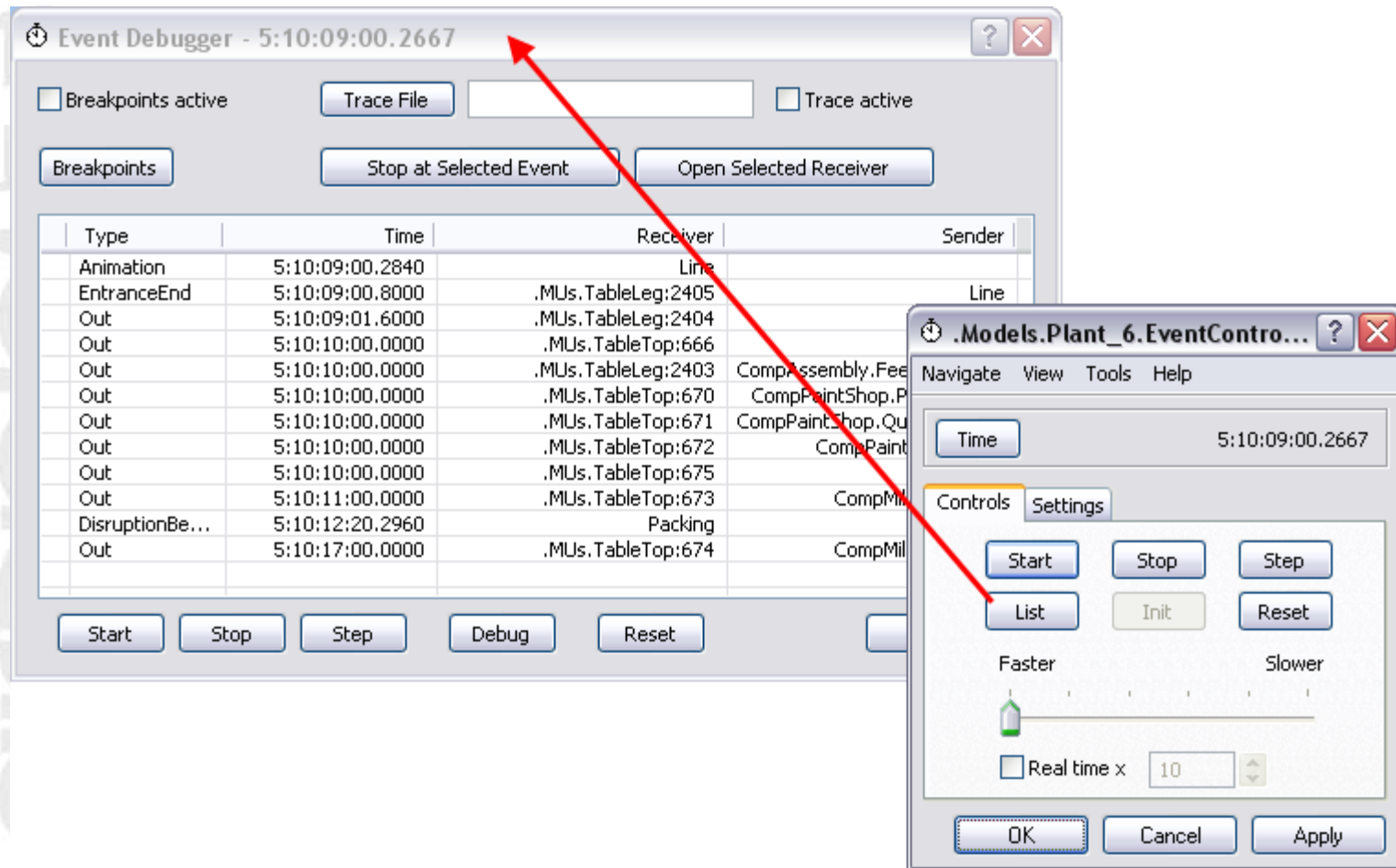
Lehetőségek:

Step (végiglépkedés) az eseményen egyszerre egy lépésen.

Adja meg **Breakpoint** - okat az események előtt.

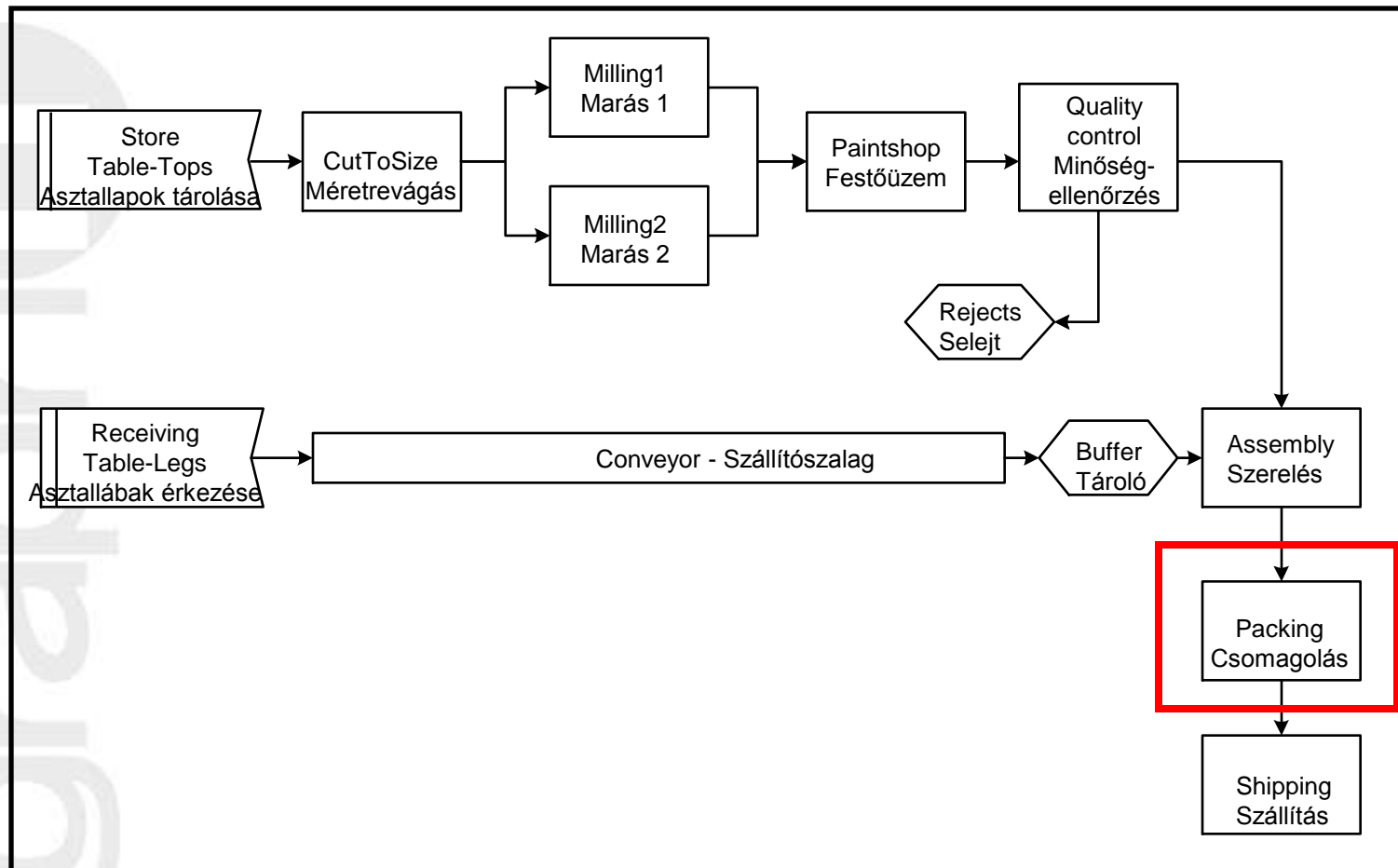
Bezárás:

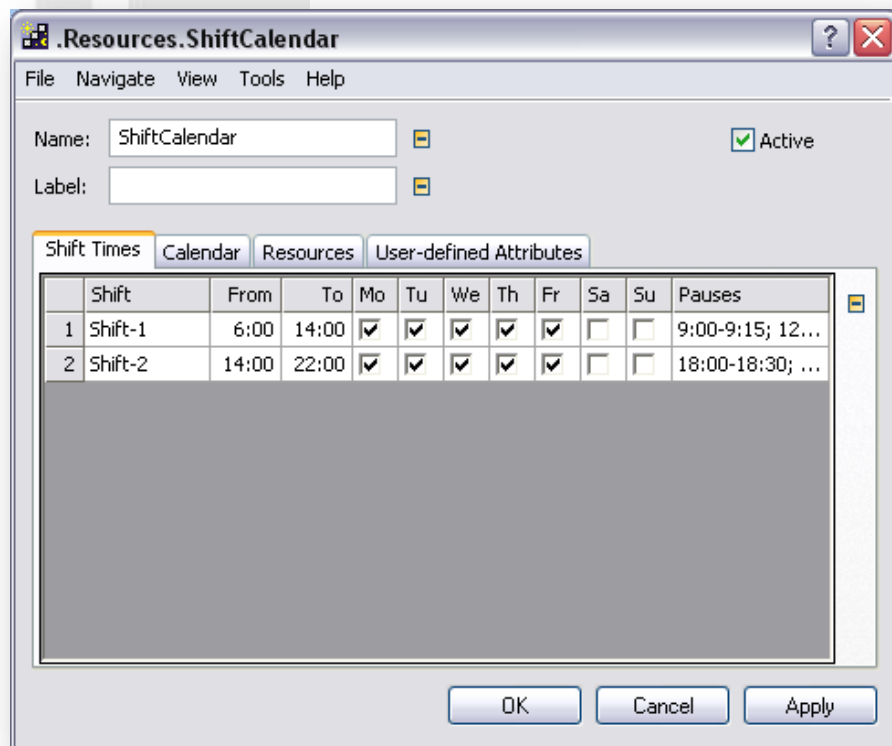
Kattintson a **Close** gombra.




15. fejezet

Időkezelés



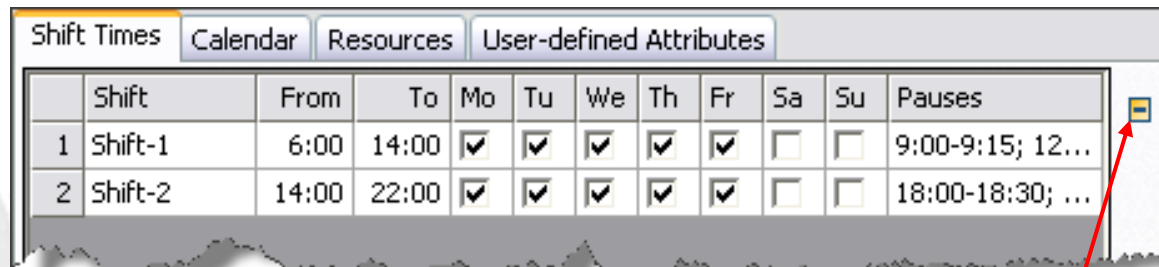


Jellemzők :

- Ikon: 
- Kapacitás: 0
- Erőforrás objektum

A *ShiftCalendar* az üzemben lévő különböző műszakok modellezésére szolgál.

Egy műszakban ki lehet kapcsolni gépeket a modellben az anyagáram objektum **Pause** attribútumának átállításával igazra (true).

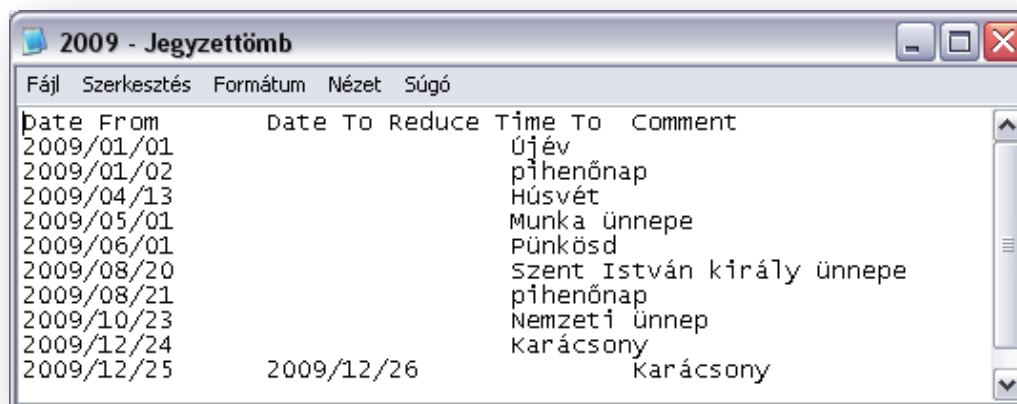


	Shift	From	To	Mo	Tu	We	Th	Fr	Sa	Su	Pauses
1	Shift-1	6:00	14:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	9:00-9:15; 12:...
2	Shift-2	14:00	22:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	18:00-18:30; ...

Mielőtt a beállításokat szerkeszteni kapcsolja ki az öröklődést és kattintson az **Apply** gombra.

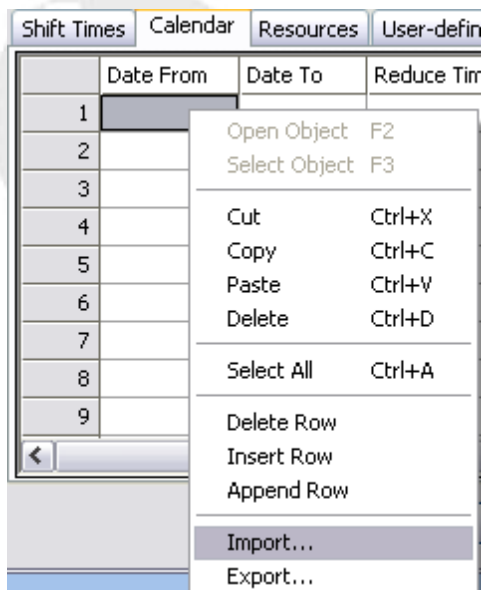
Adja meg a műszak (**Shift**) nevét, a kezdési időt (**From**), a befejezési időt (**To**), és a szünetek hosszát (**Pauses**) a **Shift Times** fülön. Annak megadására, hogy melyik műszak melyik napon dolgozik a következő cellákra kell kattintani: **Mo, Tu, We, Th, Fr, Sa, Su**.

A műszakok között lehet átfedés. Amíg egyetlen műszak sem dolgozik, a kapcsolódó objektumok sem működnek. A Plant Simulation a statisztikában ezt **UnplannedTime** néven jegyzi be.



Meg lehet adni azokat a napokat, amikor a gyár nem dolgozik (**Date From** és **Date To**) vagy csak részidőben dolgozik (**Reduced Time**). Megjegyzés (**Comment**) fűzhető a **Calendar** fülön, ami leírja a napot. Például 2010. január 2. megadható 02.01.2010 vagy 2010/01/02 formátumban is.

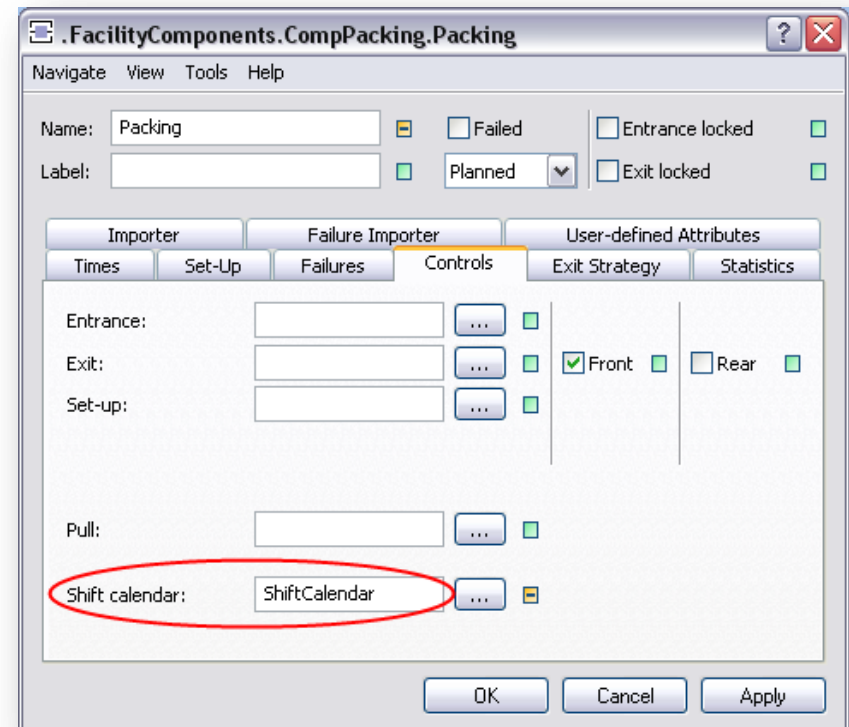
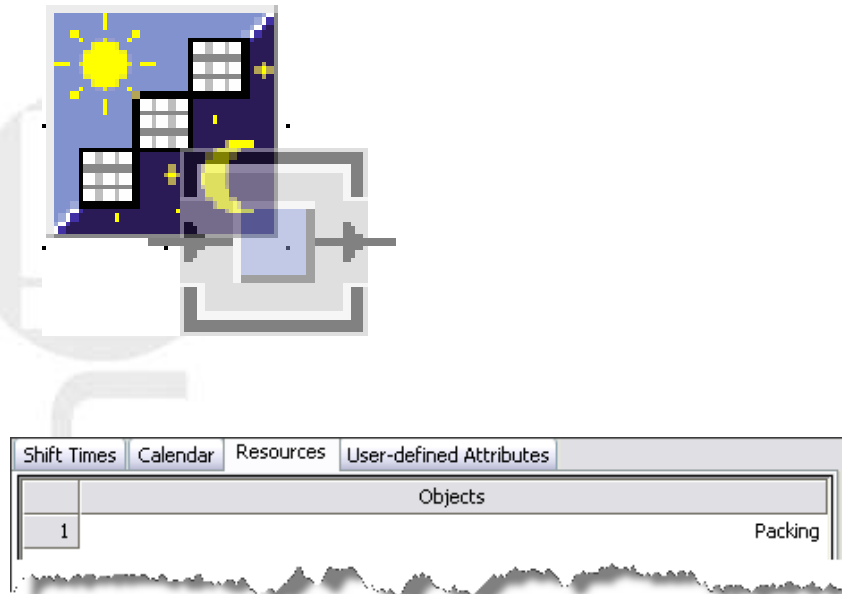
Ha a gyár egész nap nem dolgozik, akkor a **Reduced Time** oszlopba nem kell beírni semmit.



Az ünnepeket és egyéb napokat tartalmazó fájlok tab-
elválasztott szövegfájlként importálhatók:

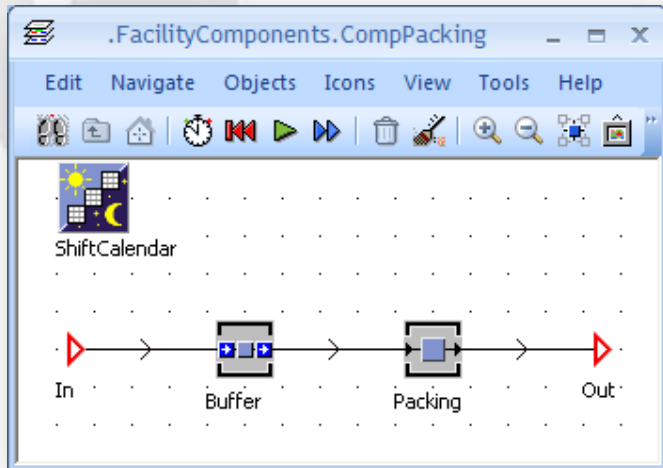
Kattintson a Calendar fül hátterén és válassza ki az
Import opciót.





Bármely anyagáram objektumot húzd és ejtsd módszerrel rá kell ejteni a *ShiftCalendar* objektum ikonjára, vagy meg kell adni a használni kívánt *ShiftCalendar* objektum nevét a **Controls** fülén az anyagáram objektum ablakában.

Leállások vezérlése ShiftCalendar segítségével



1. Készítsen egy *CompPacking Frame*-et a *FacilityComponents* mappában.
2. Illessze be a képnek megfelelő objektumokat, a *Buffer* Kapacitása (**Capacity**) legyen 100.
3. Húzza rá a *Packing SingleProc* ikonját a *ShiftCalendar* objektumra.
4. Adja meg a műszakidőket, és a csökkentett munkaidejű napokat a *ShiftCalendar* objektumban, kapcsolja ki az öröklődést. Ezek az értékek a következő fólián lesznek. Adja meg az *EventController* kezdő időpontját.

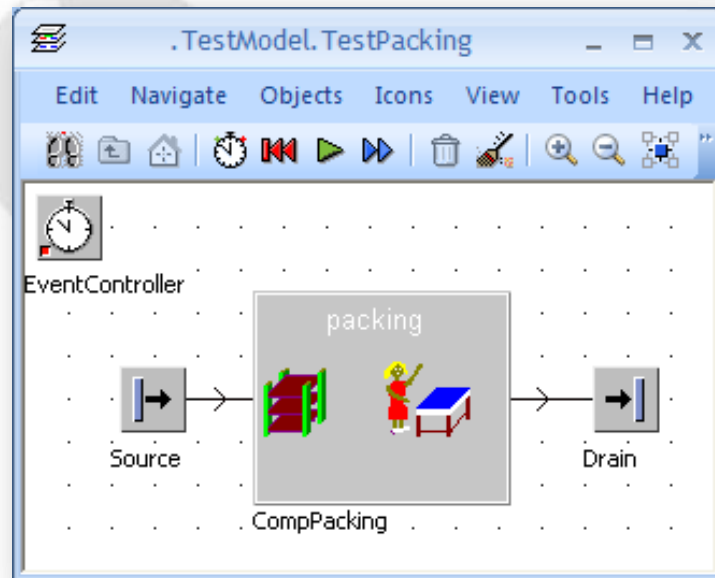
Műszakidők és nem munkanapok megadása.

Shift Times											
Calendar											
Resources											
User-defined Attributes											
	Shift	From	To	Mo	Tu	We	Th	Fr	Sa	Su	Pauses
1	Morning Shift	6:00	14:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	9:00-9:15
2	Afternoon Shift	14:00	22:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	12:00-12:45

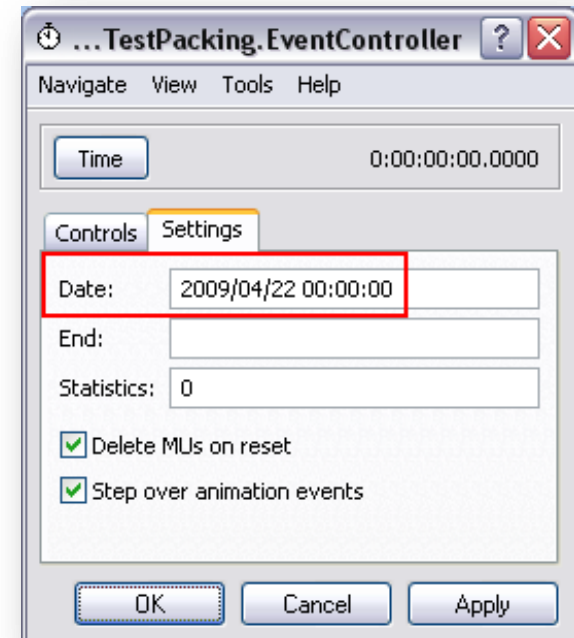
Szünetek
(munkanapokon)

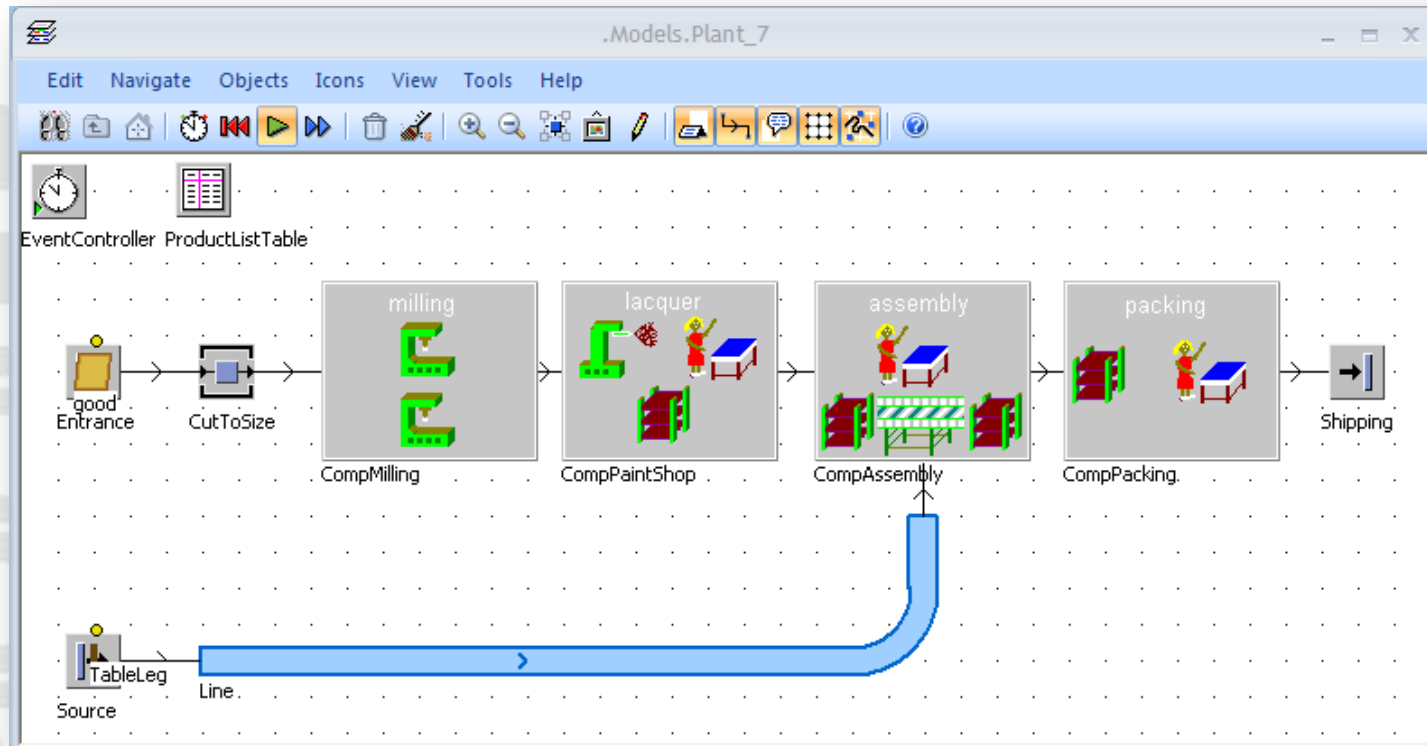
9:00-9:15
12:00-12:45
18:00-18:30
20:30-21:00

Shift Times				
Calendar				
Resources				
User-defined Attributes				
	Date From	Date To	Reduce Time To	Comment
1	2009/04/23	2009/04/24	14:00 - 15:00	Company Meeting
2	2009/05/01			Nemzeti Ünne



1. Készítsen egy tesztkörnyezetet a *TestModel* mappában. Tesztelje a *Packing Frame*-et. Ellenőrizze a *Packing* állomás megállásait.



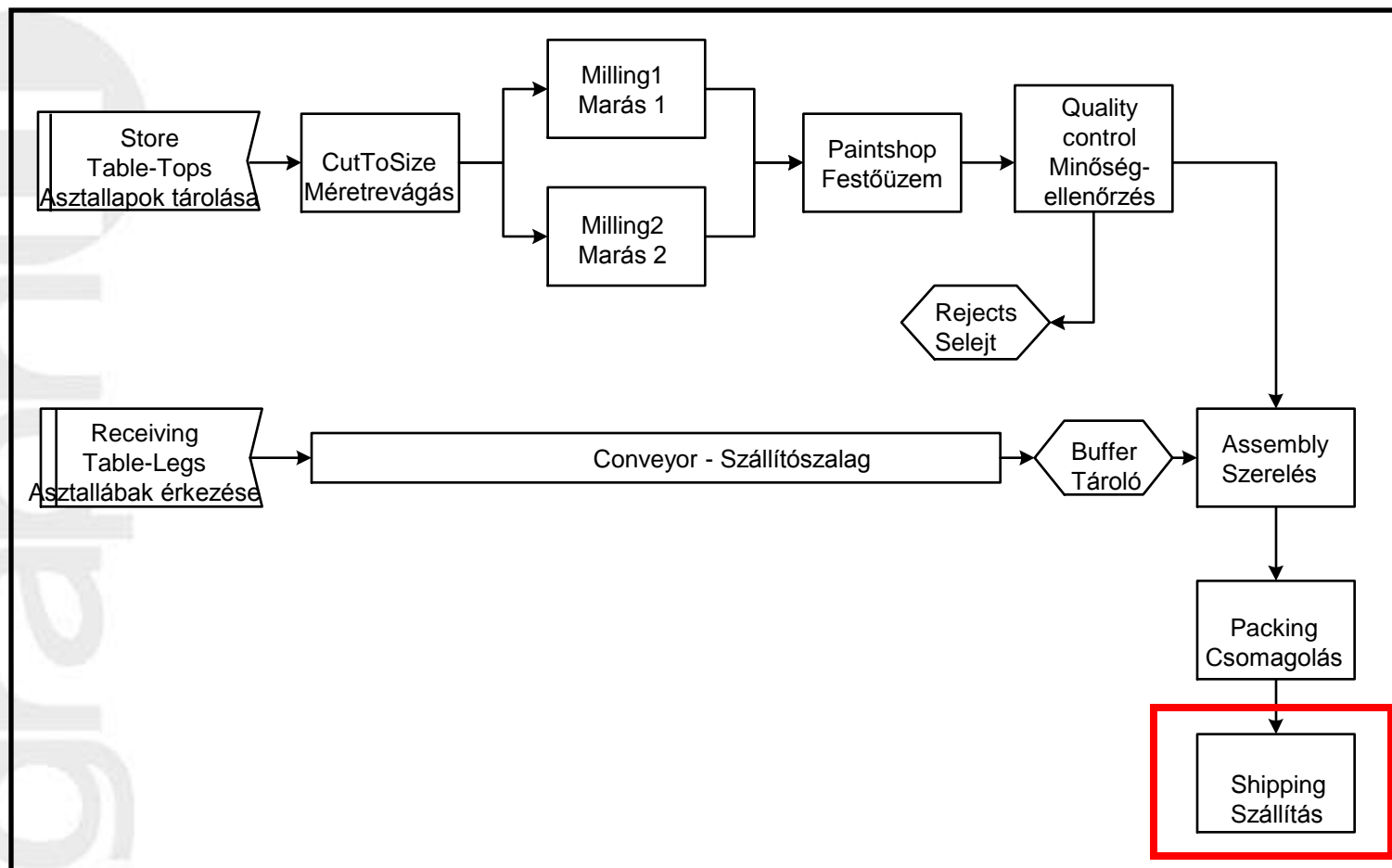


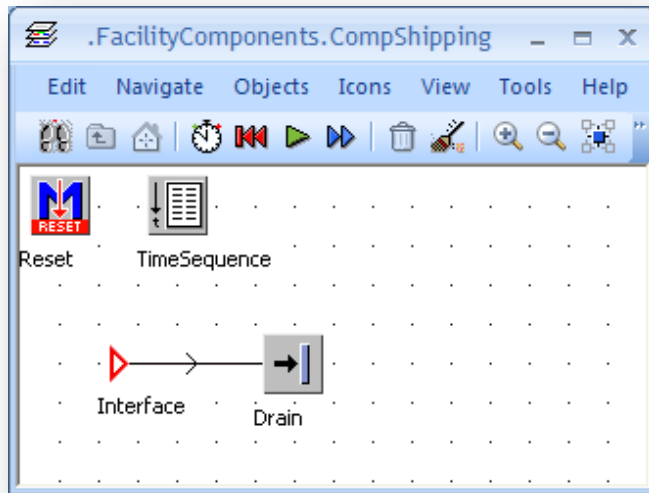
Készítsen másolatot a **Plant_6** objektumból, legyen a másolat neve **Plant_7**, és illessze be a *CompPacking Frame*-et a *Packing SingleProc* helyére.



16. fejezet

Statisztikák megjelenítése a
TimeSequence segítségével





Szeretnénk statisztikát látni arról, amikor az összeszerelt asztalok megérkeznek a szállításra (*Shipping*).


A *TimeSequence* objektumot használjuk az időpillanatok és az asztalok darabszámának rögzítésére a *Shipping* osztályon.

Ehhez a *Drain statNumIn* attribútumát fogjuk használni.

The screenshot shows a properties window titled '.FacilityComponents.CompShipping.Drain'. It contains a table with the following data:

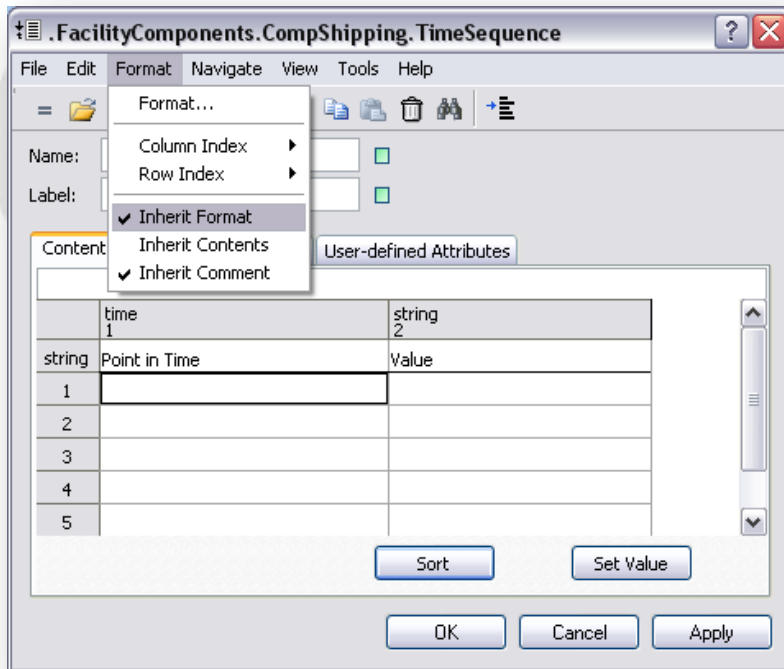
Name	Signature	Value
statFailMu	time	0.0000
statFailPortion	real	0
statFailTime	time	0.0000
statistics	([any])	
statisticsStartRes	time	0.0000
statMaxNumMU	integer	0
statMinNumMU	integer	0
statNumIn	integer	0
statNumOut	integer	0
statPausingCount	integer	0
statPausingDelta	time	0.0000
statPausingMu	time	0.0000
statPausingPortion	real	0
statPausingTime	time	0.0000
statPredeposition	integer	0
statProdFailPortion	real	0

Jellemzők :

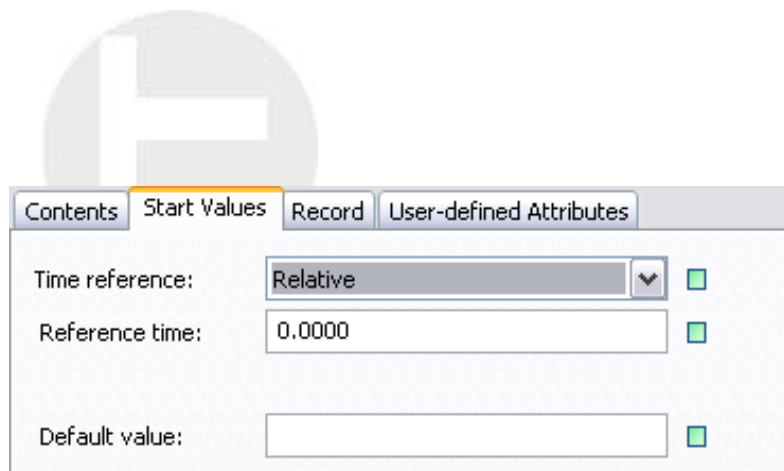
- Ikon: 
- Kapacitás: 0
- Információs áram objektum

A *TimeSequence* a szimuláció futása során értékeket rögzít, mint például az áramló elemek számát, stb.

Ki kell választani a megfelelő adattípust a rögzített attribútumhoz a **Value** oszlopban!



Győződjön meg arról, hogy a **Resource statistics** opció aktív arra az objektumra, melynek az adatait gyűjteni kívánja.

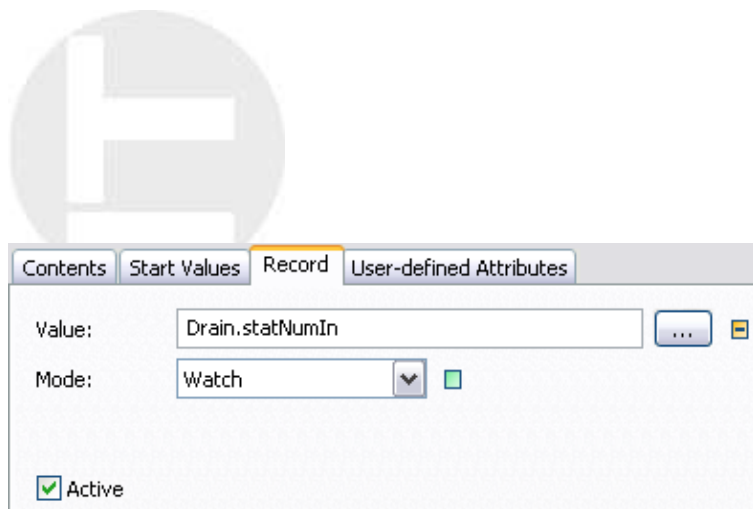


Válassza ki a **Time Reference** – t a **Start Values** fülön:

Absolute: Időpont.

Relative: Időtartam.

A legtöbb esetben a relatív időreferencia használható nulla értékkel, ami azt jelenti, hogy a *TimeSequence* azonnal elkezdi az aktív értékek rögzítését.



Meg kell adni a rögzítendő érték útvonalát a **Recording** fülön.

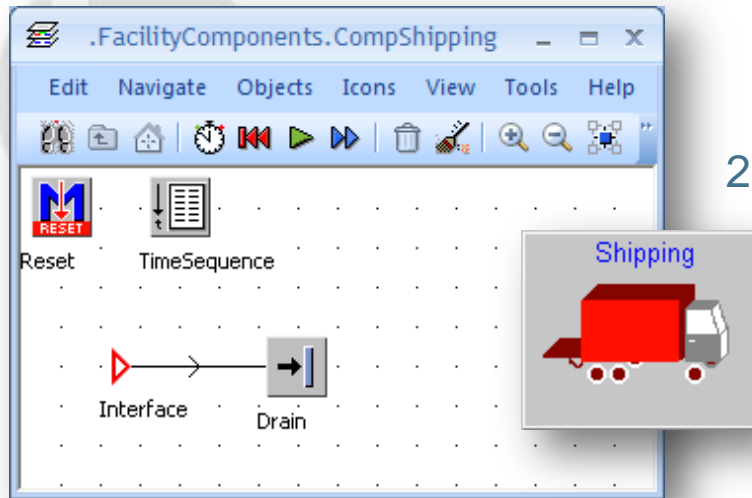
Ki kell választani a figyelés módját (**Mode**) :

Watch: Amikor az érték változik.

Sample: Periódikusan, akár változott, akár nem.

Az **Active** opciót be kell kapcsolni, hogy rögzítse is az értékeket!

Statisztika gyűjtése a TimeSequence használatával



1. Készítsen egy **CompShipping** *Frame* - et a *FacilityComponents* mappában.
2. Illessze be az ábrán látható objektumokat. A *TimeSequence* a *Drain* objektumba érkező MU-k számát kell, hogy rögzítse. A statisztika a *TimeSequence* **Contents** fülén jelenik meg.

3. Illesszen be a reset metódust. Ez a metódus törli a *TimeSequence* által rögzített értékeket minden szimulációfutás után.

4. Mit kell kiválasztani, hogy a *Drain* statisztikáját gyűjtsük?

```
is
do
    timesequence.delete;
end;
```

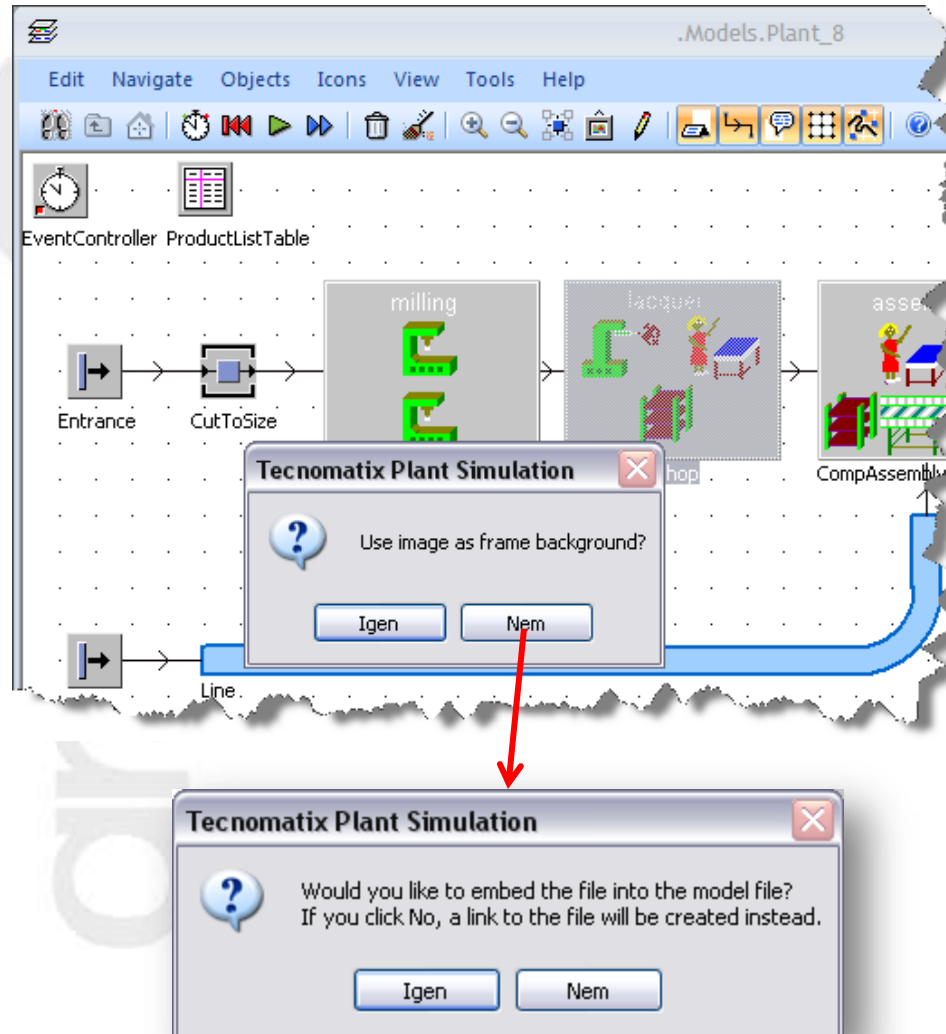
5. Készítsen másolatot a **Plant_7**-ből **Plant_8** néven. Illessze be a *Shipping* *Frame*-et.



17. fejezet

CAD modell betöltése háttérként

- Áttekinthetőbbé teszi az elrendezést
- A szállítószalagok és az utak használhatják annak arányait
- Pontos pozícionálást tesz lehetővé

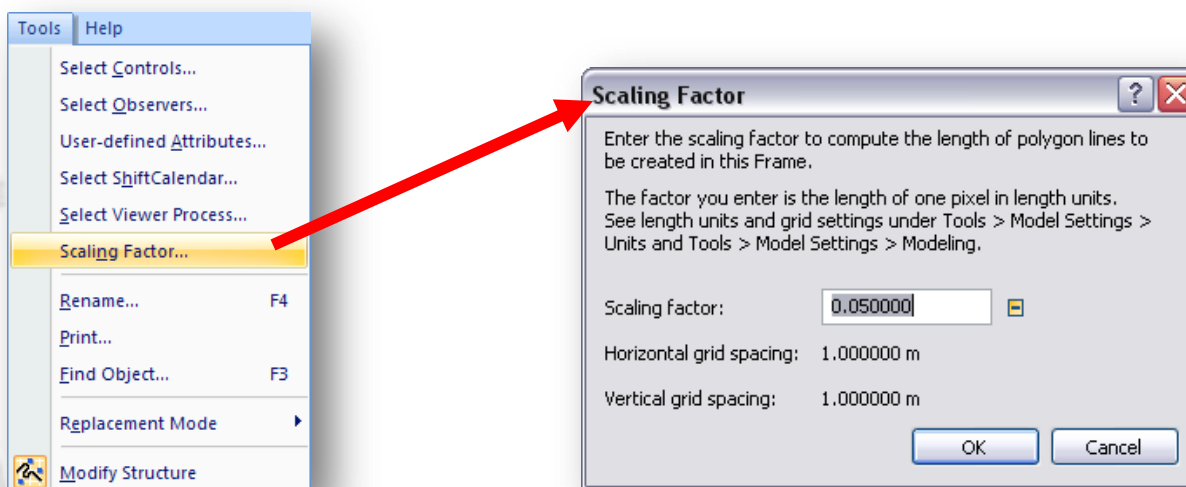


Húzza és ejtse a CAD fájlt a megfelelő *Frame*-re.

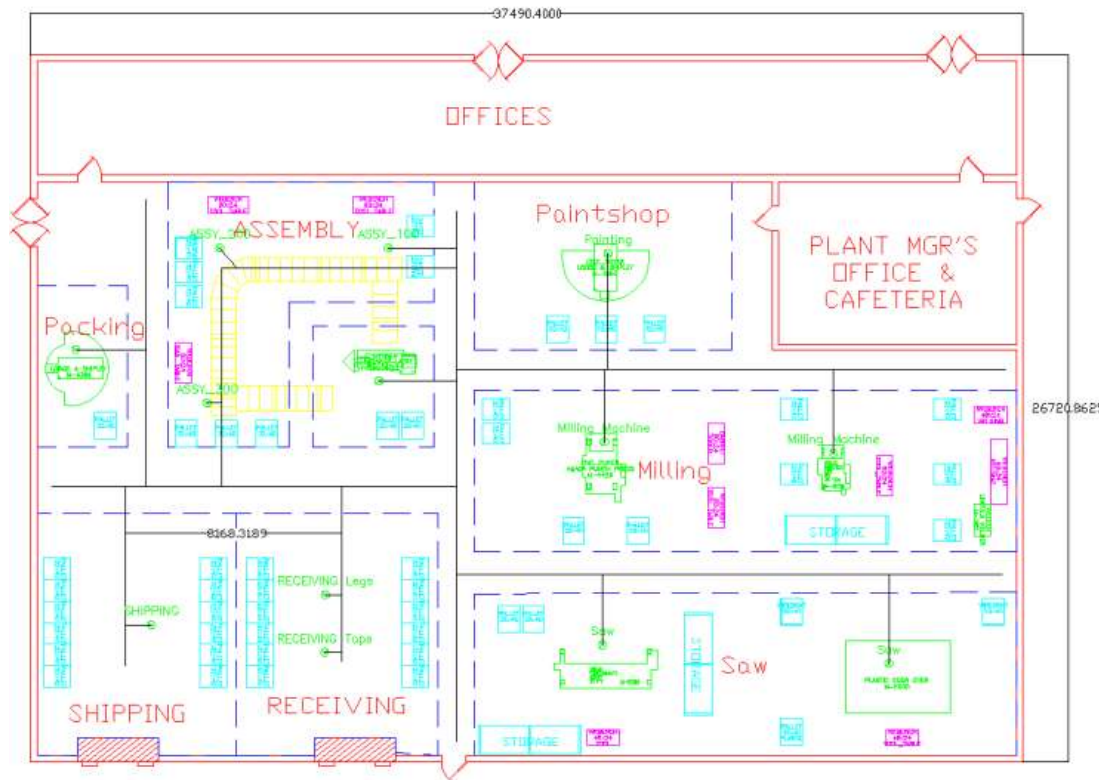
Megjelenik egy üzenetablak, amely megkérdezi, hogy valóban háttérként kívánja-e használni a fájlt.

Nem válasz esetén egy újabb üzenet jelenik meg, rákérdezve arra, hogy beágyazni kívánja-e a fájlt a *Frame*-be.

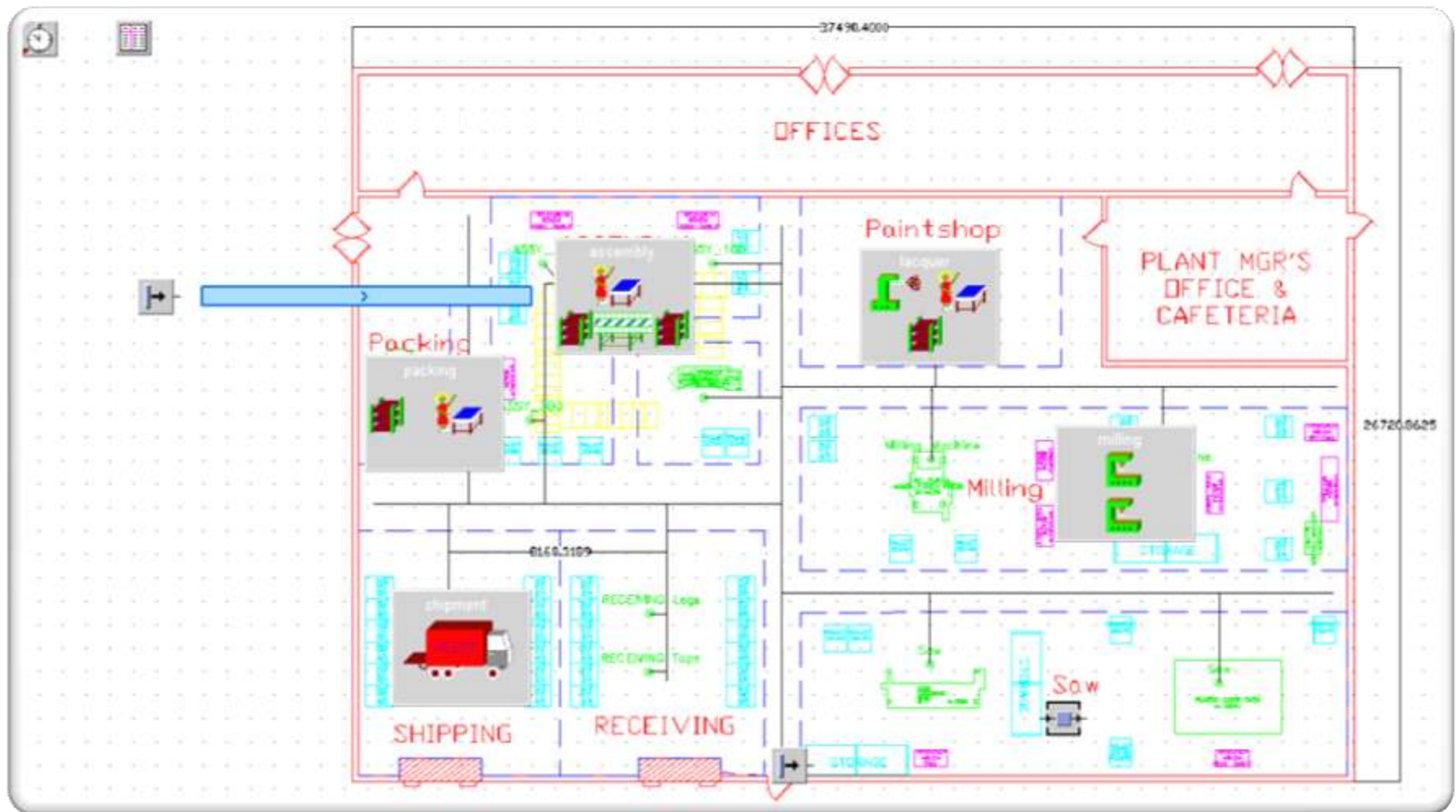
A skálázási tényező a hosszmérték egy képpontnak megfelelő hossza. Alapértelmezetten két rácspont között (20 pixel) a távolság 1 méter.



A „TableAssembly.dwg“ modell egy pixel milliméterenként. Így a skálázási tényező az importáláskor 50.



- Készítsen másolatot a **Plant_8** modellből, legyen a neve **Plant_8_with_CAD**
- Adja meg a skálázási tényezőt (**Scaling factor**)
- Importálja a layout - ot
- Állítsa alapértékre a skálázási tényezőt
- Rendezze át az objektumokat az elrendezésnek megfelelően



Tanfolyam

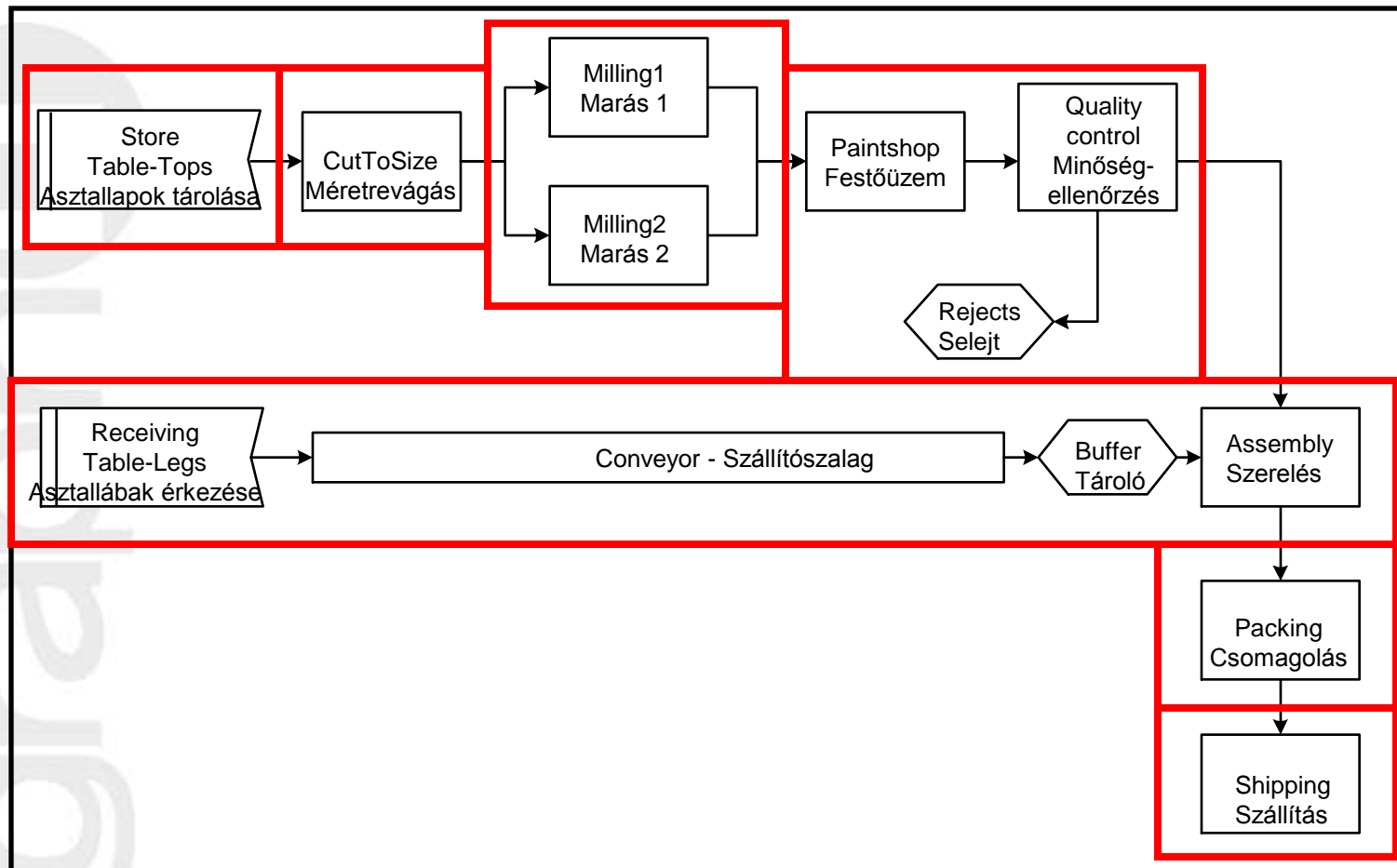
Plant Simulation 9

2. modul: Stratégiák

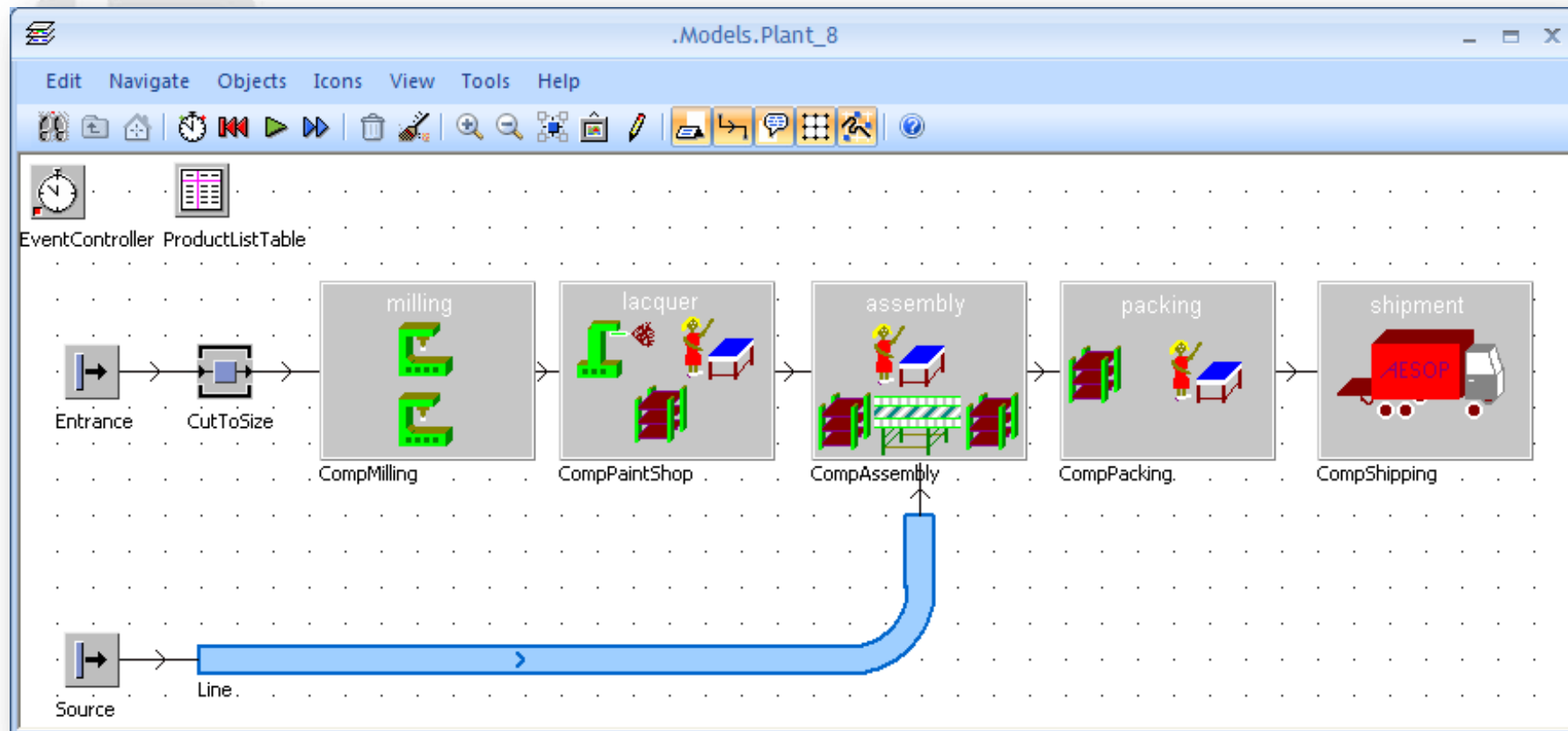


1. fejezet

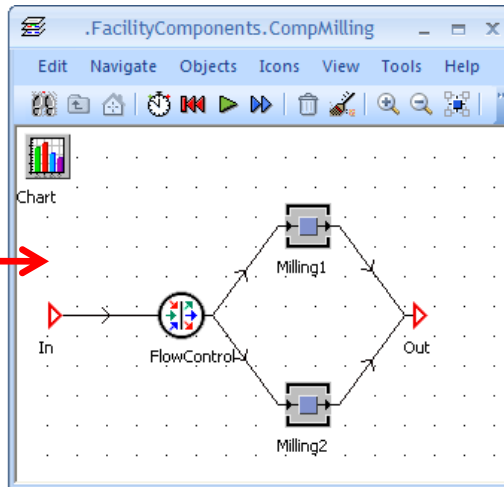
A feladat



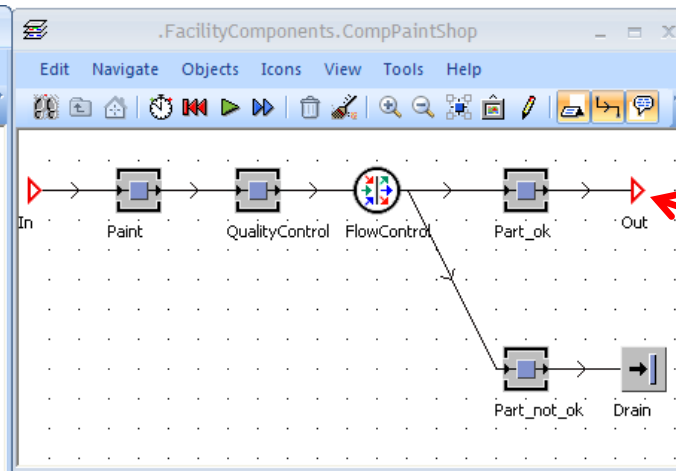
Az asztalgyártó üzem



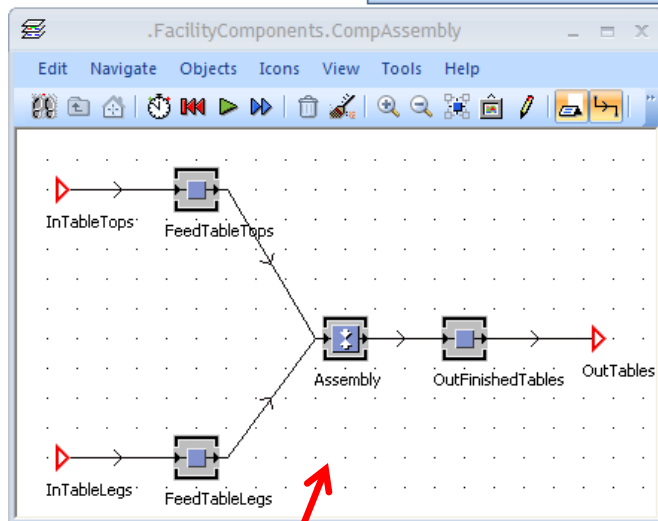
Milling
Marás



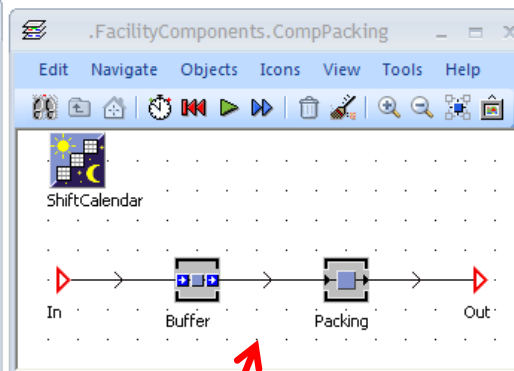
Paintshop
Festés



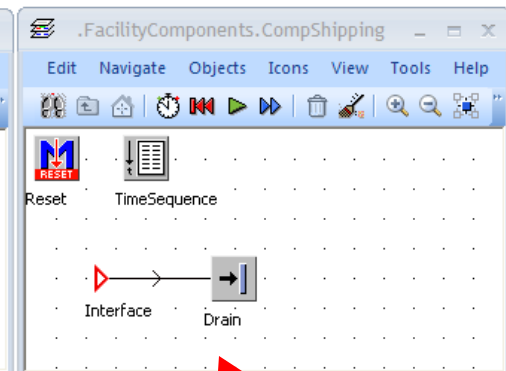
Assembly
Szerelés



Packing
Csomagolás



Shipping
Kiszállítás





2. fejezet

A Metódus (Method)
objektum, használatának módja és
beállításai

... Az objektum-orientált nyelvek objektumokon alapulnak, amelyek adatszerkezeteket tartalmaznak, és műveleteket (metódusokat). Az objektumban tárolt információ csak az objektumosztály metódusain keresztül manipulálhatók. Az objektum egy metódusának elindításához egy üzenetet kell küldeni az objektumnak, ami tartalmazza a metódus nevét és paramétereit. A működés futás közben az üzeneteken alapul.

*... **

** Dubbel Taschenbuch für den Maschinenbau, 18. Edition*

A metódus (*Method*) egy kis program, hasonló a Basic, Pascal vagy C++ nyelvek függvényeihez.

A **SimTalk**, a Plant Simulation programozási nyelve az **Eiffel** programozási nyelven alapszik. Ez erősen hasonlít más programozási nyelvekre.

Készíthető olyan metódus, amely képes a modellbe illesztett objektumok attribútumainak olvasására, és módosítására. Egy objektumot kiválasztva és az **Objects > Show Attributes and Methods** parancsot használva az összes beépített attribútum és metódus megjeleníthető.

Emellett tetszőleges számú felhasználói attribútum is definiálható.

A *Method* objektum teljesen integrált a Plant Simulation objektum-orientált koncepciójába. A forráskód közvetlenül a *Method* objektumba illeszthető be. Ez után a rendszerbe beépített fordító (*Interpreter*) feldolgozza a programkódot a szimuláció futása során.


Egy metódus alkalmas a következőkre:

- A szimuláció során történő eseményekre történő reagálásra.
- Különböző jellemzők lekérdezése és beállítása.
- Kifejezések futtatása.
- Objektumok módosítása és viselkedésük kiterjesztése.
- Új felhasználók számára egyéni párbeszédablakok készítése a modellhez.

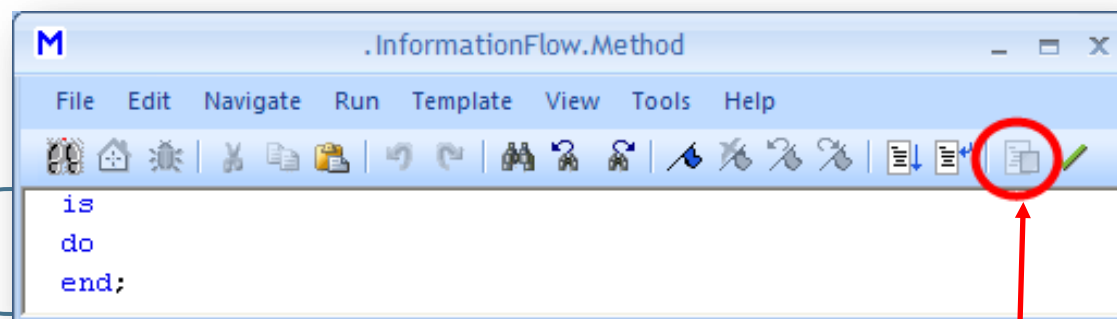
Ez lehetővé teszi:

- Különböző megközelítések implementálását, a modell teljeskörűen igényekre szabását.
- A szimulációs modell hatékonyságának növelését.
- A szimulációs modell rugalmasságának növelését.

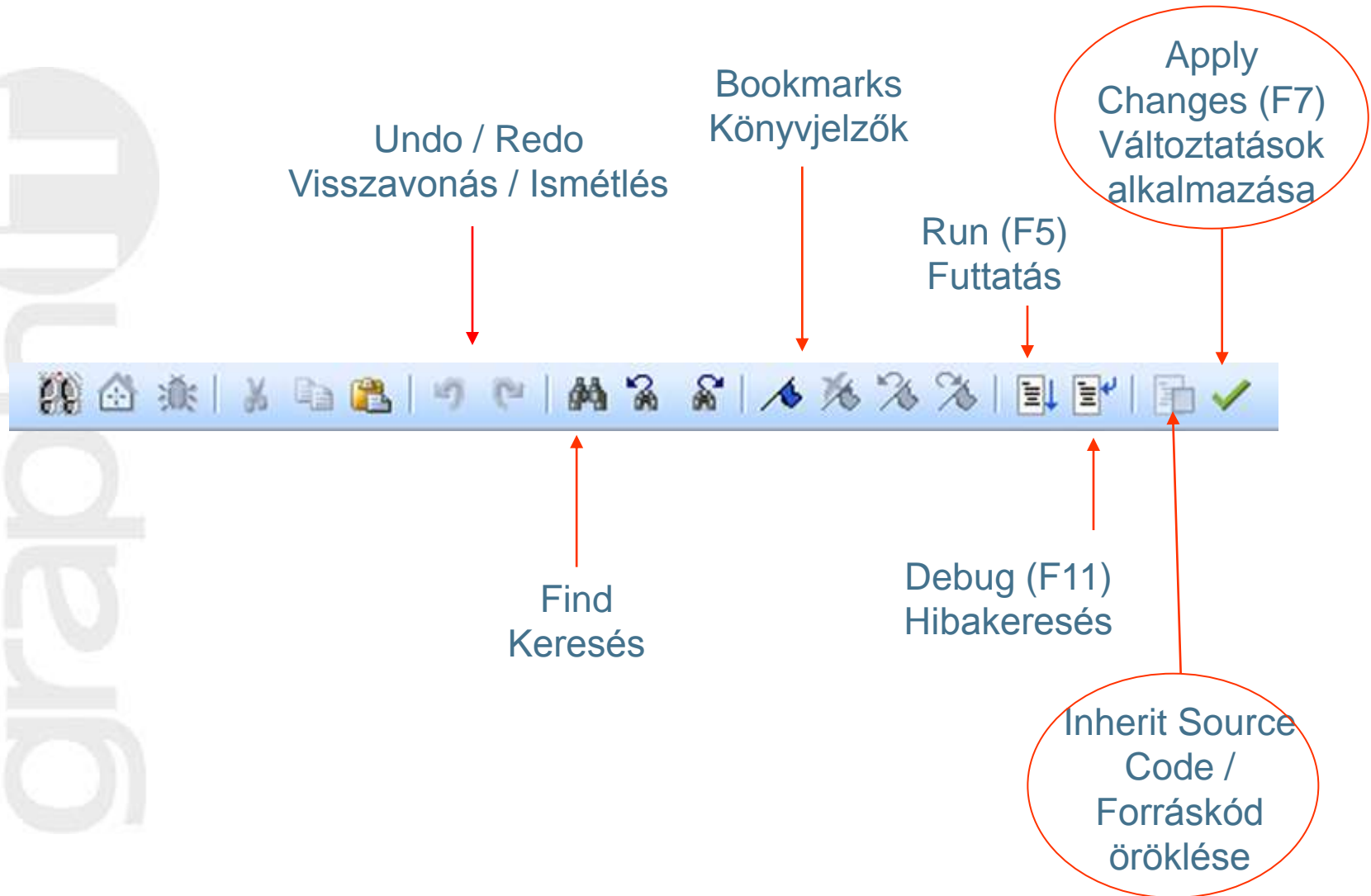
Jellemzők :

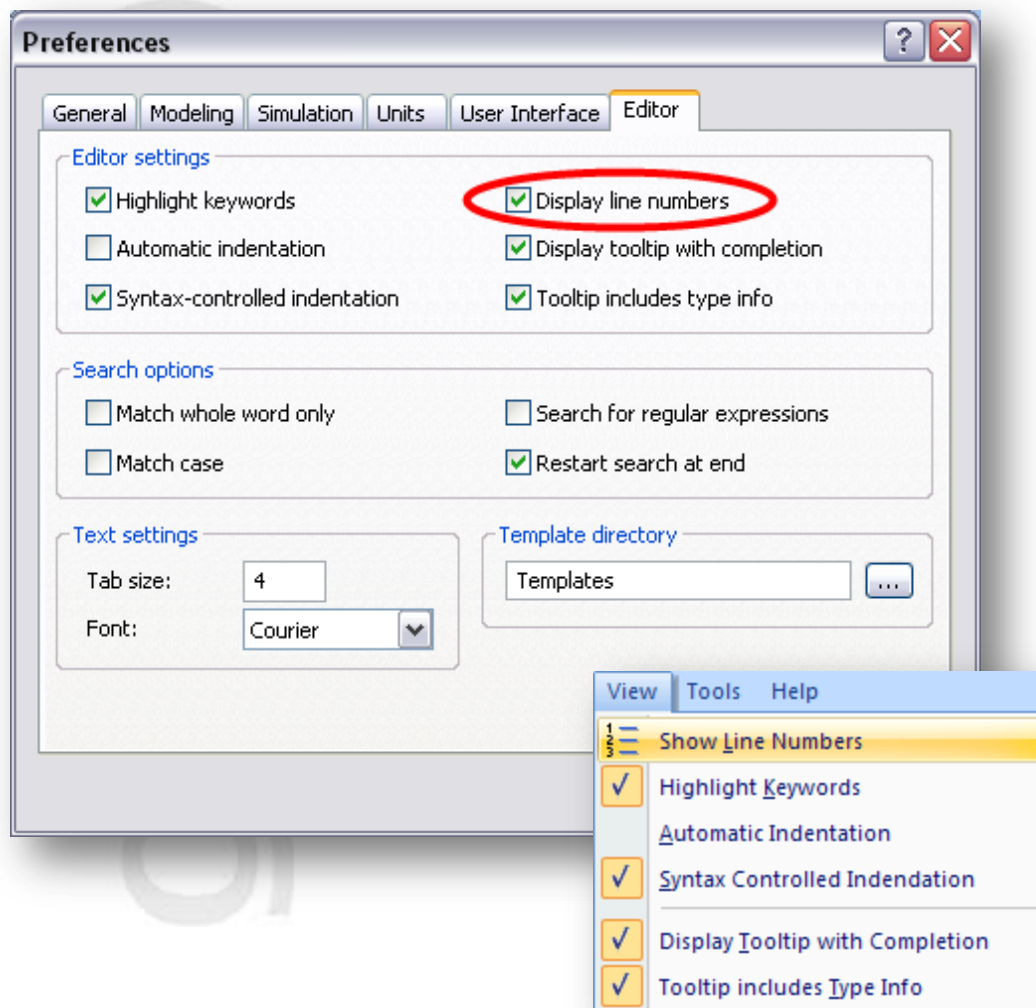
- Ikon: 
- Kapacitás: 0
- Információ áram objektum

A metódus
teste



A modellbe beillesztve egy metódust, a Plant Simulation kikapcsolja az **Inherit Source Code** (forráskód öröklése) opciót (az osztálytól).





Válassza ki a következő menüpontot:

Tools > Preferences (Eszközök > Beállítások)

Meg lehet adni a metódus szerkesztő beállításait.

Javasolt bekapcsolni a

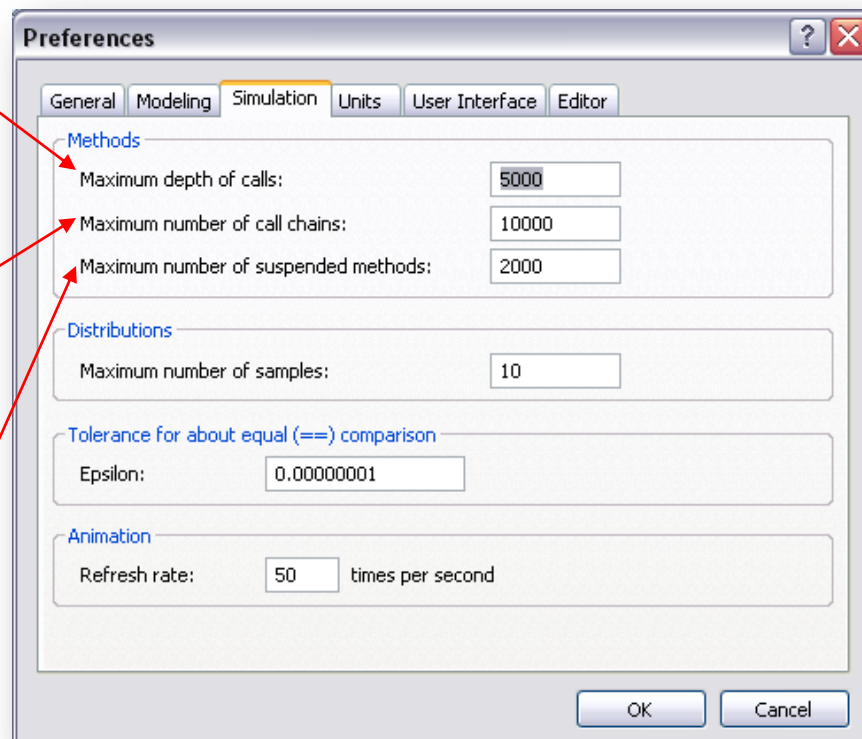
Display line numbers
(Sorszámozás mutatása)

opciót.

Azt adja meg, hogy hány metódus hívható meg másik metódusból.

Az egyszerre meghívható metódusok maximális száma.

Az egy időben nem futó metódusok maximális száma, amelyek esemény történésére várnak.





3. fejezet

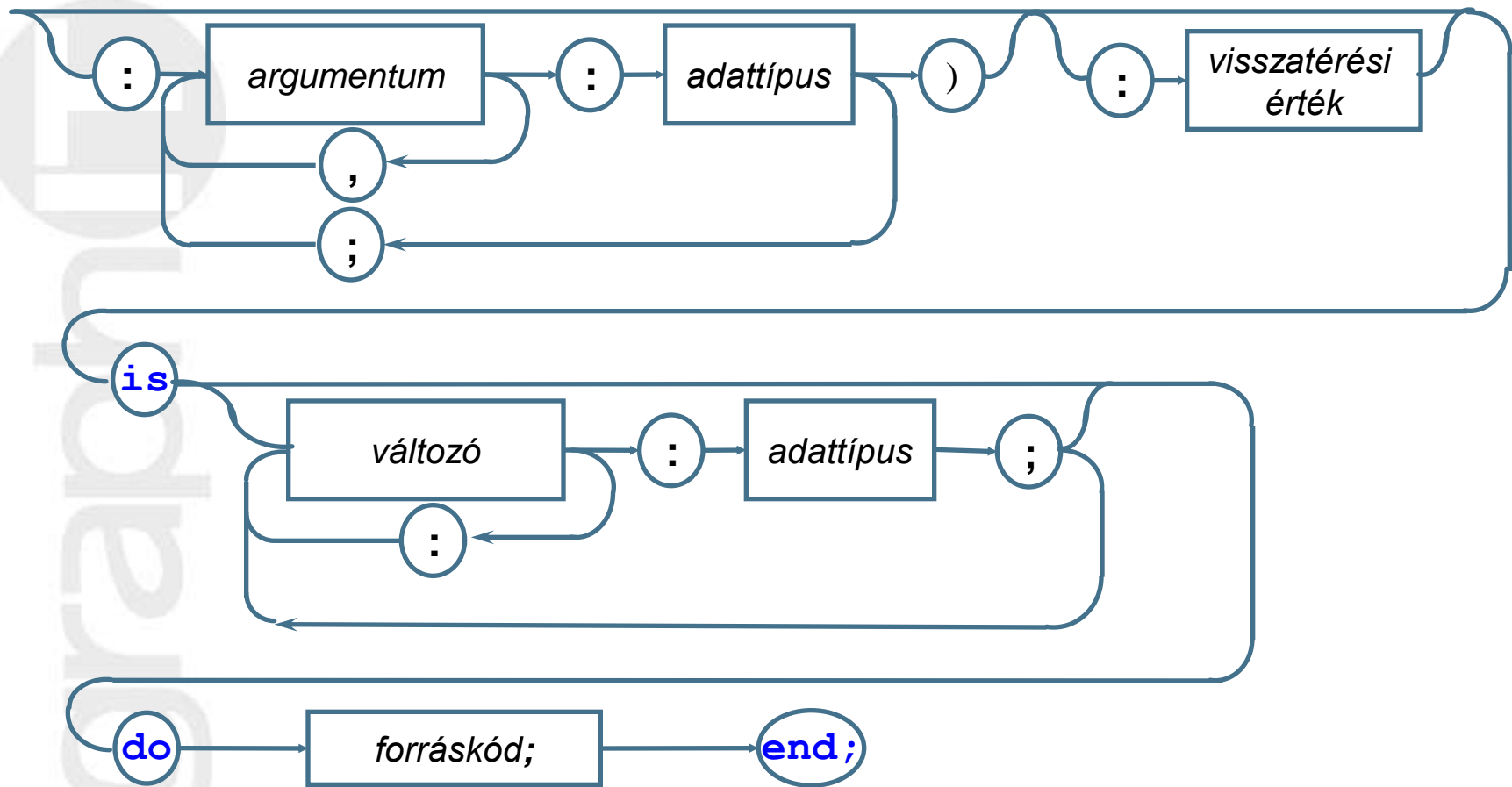
Metódusok szintaxisa

A SimTalk programozási nyelv

Metódus hibakeresés

Egy *Method* (metódus) objektum a következő részekből áll, nem mind kötelező.

- [Argument]** ← a funkcionalitás argumentumok (paraméterek) hozzáadásával bővíthető. A meghívónak annyi paramétert kell átadnia, amennyi a metódusban definiált. Az átadott adattípusnak meg kell egyeznie a várt adattípussal.
- [Result]** ← meg kell adni a visszatérési érték típusát. Ezt a meghívóhoz a **result** kulcsszóval kell kapcsolni, amikor visszatér az értékkel.
- is** ← elválasztja az argumentumok és a helyi változók deklarálását
- [Local variables]** ← a helyi változókhoz csak abban a metódusban lehet hozzáférni, amelyben azok deklarálva lettek. Akkor kell használni ezeket, ha csak az adott metóduson belül van szükség az értékeikre.
- do** ← a megfelelő forráskód a do után következik.
- [Statements]** ← a forráskód, a kifejezések, ciklusok, metódus hívások, stb.
- end;** ← a forráskód végét jelzi. Az **end** után már csak megjegyzések következhetnek.



- Hozzárendelés művelet

A hozzárendelés művelet $:=$ új értéket rendel egy változóhoz. $\langle \text{változó} \rangle := \langle \text{új érték} \rangle$;

- Számítási műveletek

Összeadás, kivonás, szorzás, osztás (+, −, *, /) és egyéb műveletek, mint például a trigonometriai függvények, logaritmus és exponenciális függvények.

- Összehasonlító műveletek

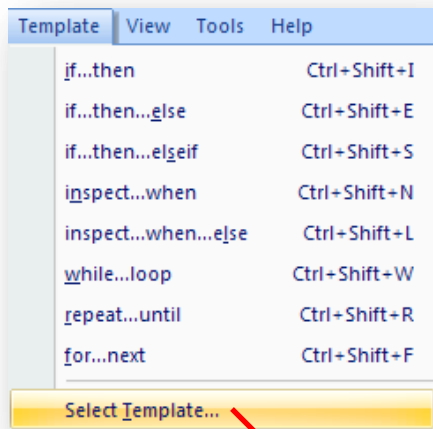
Összehasonlító műveletek $=$, \neq , $>$, \geq , $<$, \leq , $==$ két értéket hasonlítanak össze.

Az eredmény TRUE (igaz) vagy FALSE (hamis) *boolean* adattípussal.

Az **and** (és), **or** (vagy), **not** (nem) *boolean* műveletek is használhatók kifejezésekre.

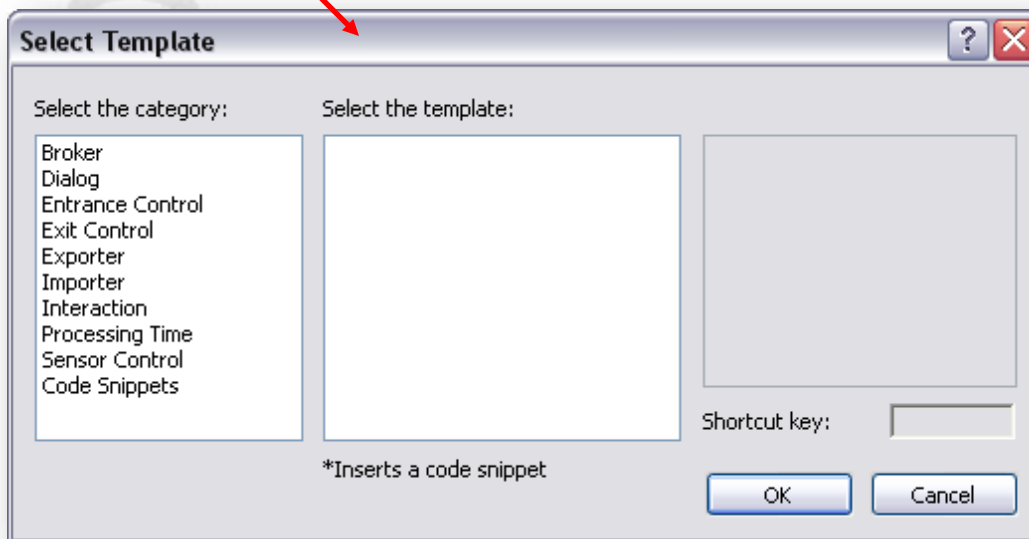
- Bemeneti és kimeneti műveletek

Adat betöltés és mentésre szolgálnak.



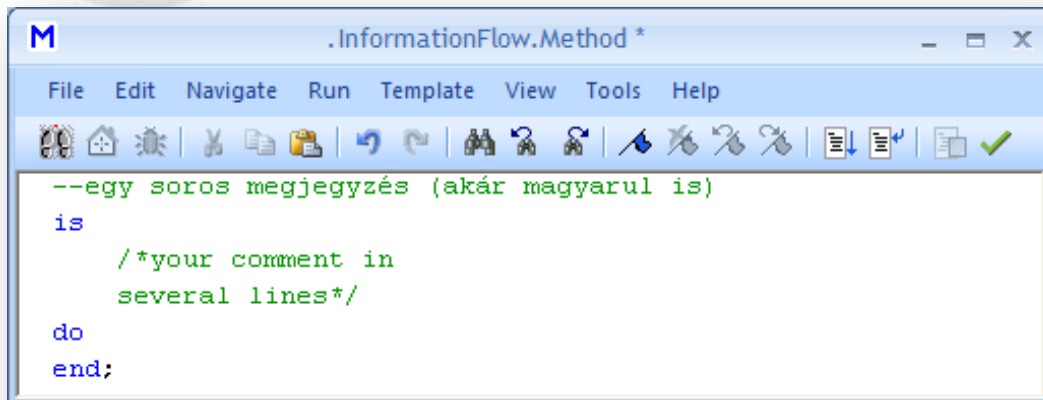
A metódus sablonok használata nagyon egyszerű és biztonságos módja a programozásnak, mivel egy komplett kódrészletet illeszt be.

A metódus sablonok definíciója lehetővé teszi előre definiált metódusok használatát.



A megjegyzések megnövelik a metódus olvashatóságát, és segítenek megérteni annak funkcióját.

Az is beírható, hogy melyik objektum hívja meg a metódust.



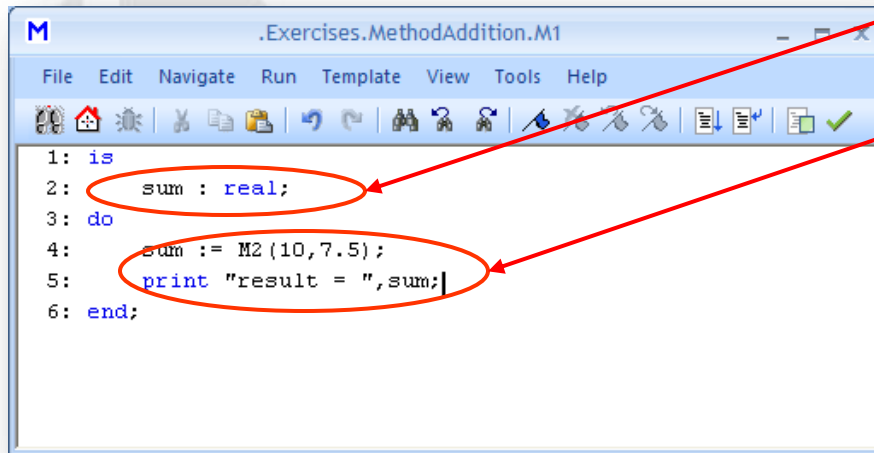
```
--egy soros megjegyzés (akár magyarul is)
is
    /*your comment in
       several lines*/
do
end;
```

Két kötőjellel `--` kezdődik az egy soros megjegyzés, ami a sor végén véget is ér.

Több soros megjegyzés `/*` kezdet és `*/` vég közé írható be.

A *Method* (metódus) a megjegyzés szövegét zöld színnel jeleníti meg.

A példa egy egyszerű kifejezést, helyi változót, és bemeneti és visszatérési paramétert mutat be:



```
1: is
2: sum : real;
3: do
4: sum := M2(10,7.5);
5: print "result = ",sum;
6: end;
```

Helyi változó (Local variable)

Forráskód (Statement)

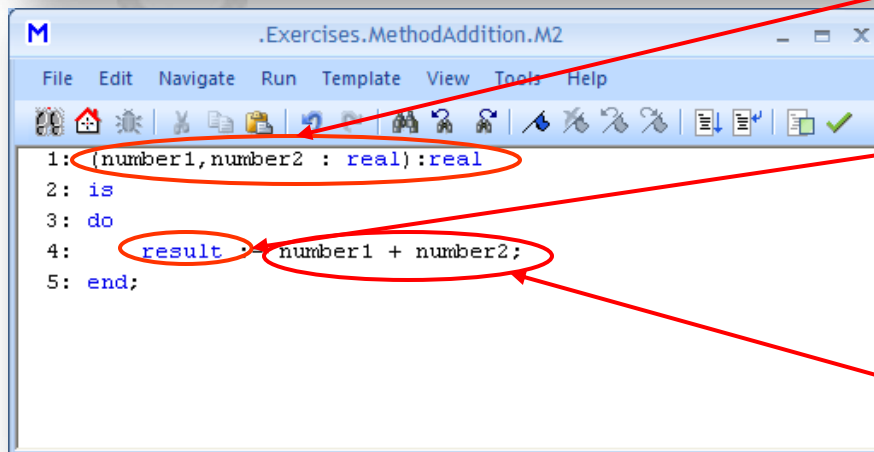
A metódus meghívja az *M2* metódust, és átadja az argumentumokat.



Bemeneti argumentum (Input argument)

Visszatérési argumentum (Return argument)

A *result* kulcsszóhoz történő hozzárendelés szükséges ahhoz, hogy az érték visszaadódjon a meghívó metódusnak.



```
1: (number1,number2 : real):real
2: is
3: do
4: result := number1 + number2;
5: end;
```


A Plant Simulation **automatikusan** megnyitja a *Hibakeresőt (Debugger)*, amikor a forráskód olyan hibát tartalmaz, ami miatt a kód nem futhat le. A hibák megtalálására is lehet használni, és megvizsgálható vele ahogyan a megírt programrészek futnak.

A Debugger megnyitása:

- Adja meg a töréspontokat (F9).
- Válassza ki a menüből a **Debugger > Open Debugger** parancsot vagy nyomja le az **F8** billentyűt vagy tartsa lenyomva a **Shift+Alt+Ctrl** billentyűket, amikor a metódus elkezd futni.

Használható továbbá:

- Helyi változók, argumentumok és a hívó objektumról szóló információk megtekintésére, és az MU műveletek megjelenítésére a *Watch Window* ablakban.
- Metódus futásának végigkövetése lépésről lépésre.
- Hibák felfedezése azokból a hibaüzenetekből, amelyeket a hibakereső megjelenít.

Breakpoints (töréspontok) –Metódus futtatása lépésről lépésre

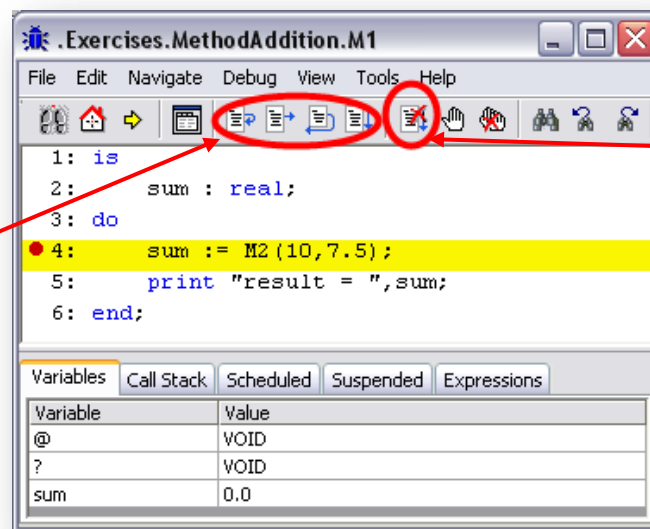
A töréspontok a metóduson belül segítenek a működésének áttekintésében, vagy lépésről lépésre futtatásában.

Nyissa meg a metódus ablakát, mozgassa a kurzort abba a sorba, ahova töréspontot kíván elhelyezni, és nyomja le az **F9** billentyűt.

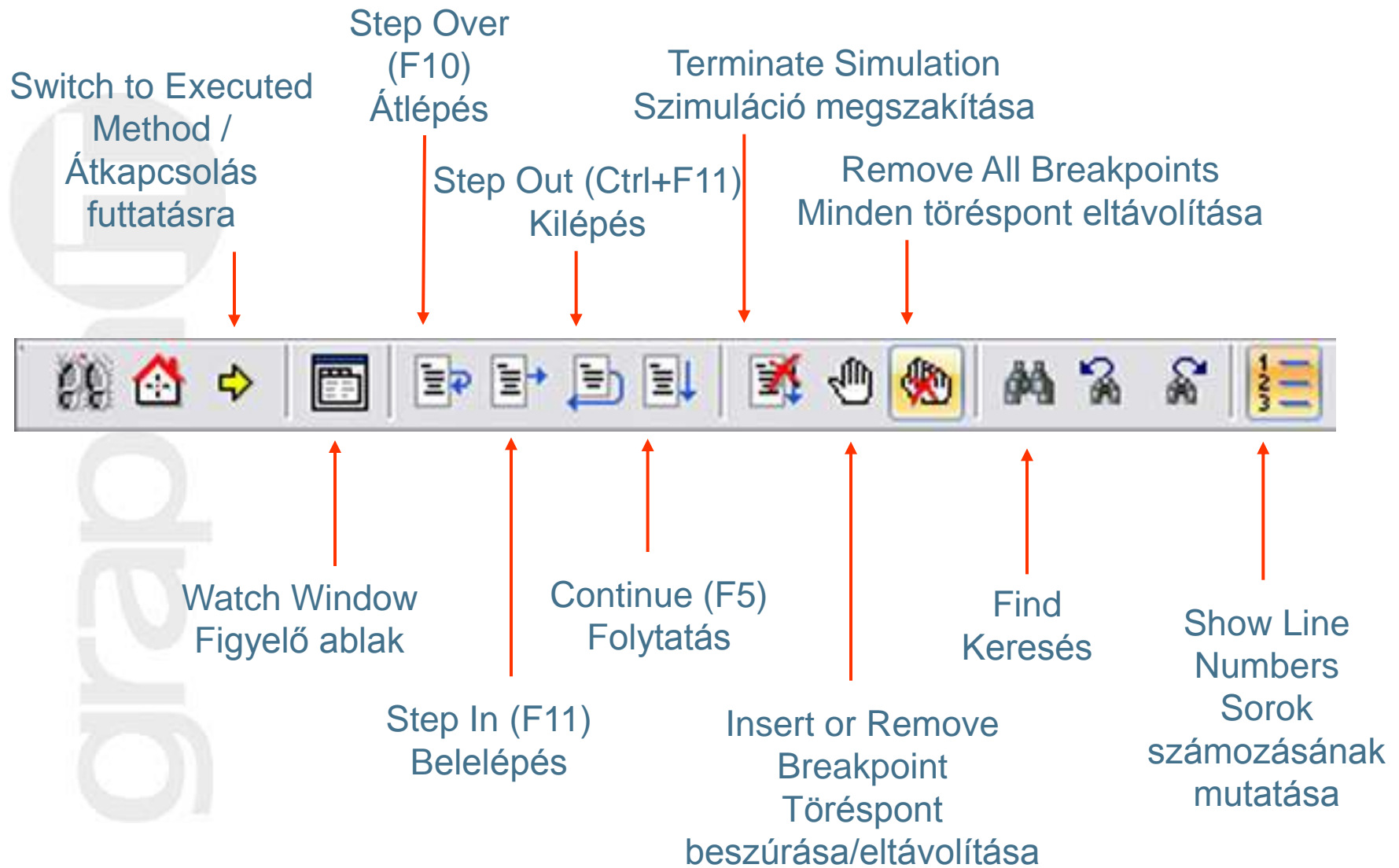
Amikor a forráskód futáskor ehhez a sorhoz ér, a Plant Simulation megnyitja a *Debugger* ablakot és kiválasztja a töréspontot.

Kattintson az eszköztáron lévő gombokra a program futásának folytatásához, vagy megszakításához.

Forráskód
futtatása
lépésről lépésre



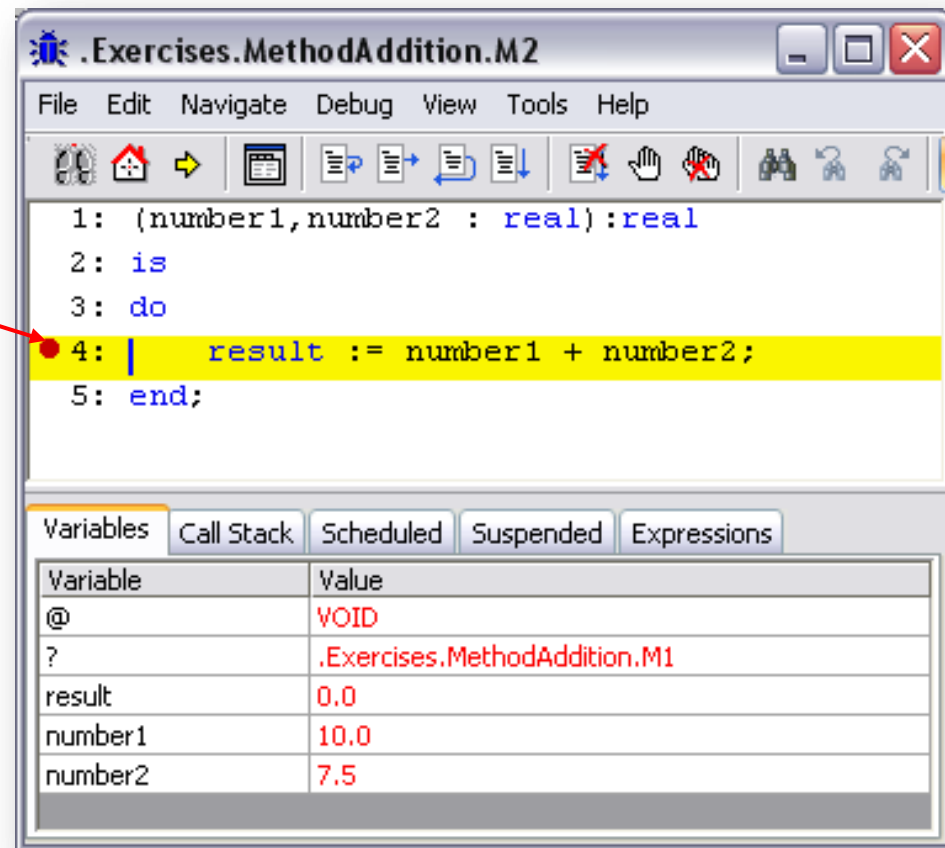
Terminate
Simulation
(Szimuláció
megszakítása, ha
hiba történik)



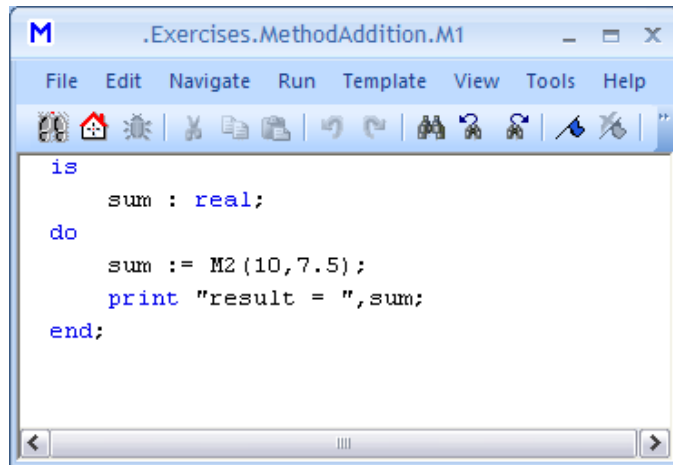
Breakpoint

Töréspont

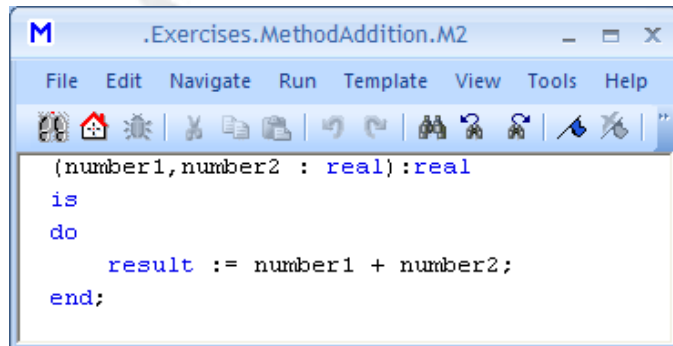
A *Watch Window* mutatja, hogy az MU amely meghívja a *Method* (@) - ot, és a nevét az objektumnak, amely meghívja a *Method* (?) - ot.



Az argumentumok és a helyi változók értékei megjelennek a *Watch Window*-ban.

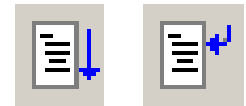


```
is
  sum : real;
do
  sum := M2(10,7.5);
  print "result = ",sum;
end;
```



```
(number1,number2 : real):real
is
do
  result := number1 + number2;
end;
```

1. Adja hozzá a *MethodAddition Frame*-et az *Exercises* mappához. Illesszen be két *Method* objektumot.
2. Írja be az ábrán látható kódot. Kapcsolja ki az öröklődést.
3. Győződjön meg, hogy a *Console* (Konzol) megjelenik (**View > Viewers > Console**).
4. Indítsa el az *M1* metódust a menü parancsokkal vagy ezekkel a gombokkal:



5. Futtassa a metódust lépésről lépésre, nyissa meg a *Watch Window* ablakát és figyelje meg az értékeket.
6. Adjon meg egy töréspontot és futtassa a metódust ismét.

- Készítsen és teszteljen olyan metódust, amely egy szám négyzetét számítja ki



4. fejezet

Nevek, azonosítók és útvonalak

Útvonalak

Minden objektumnak egyedi címe van. Néha hasznos tudni ezt a címet, hogy egyedileg azonosíthassuk az objektumot.

Különbség van azonos hierarchia szinten lévő, de azonos illetve különböző névtérben lévő objektumok között.

A **relatív útvonal** azonos névtérben lévő objektumot hív meg. Az **abszolút útvonal** pedig másik hierarchia szinten lévő objektumot hív meg.

Emellett használhatók névtelen azonosítók, általános azonosítóval rendelkező objektumok meghívására.

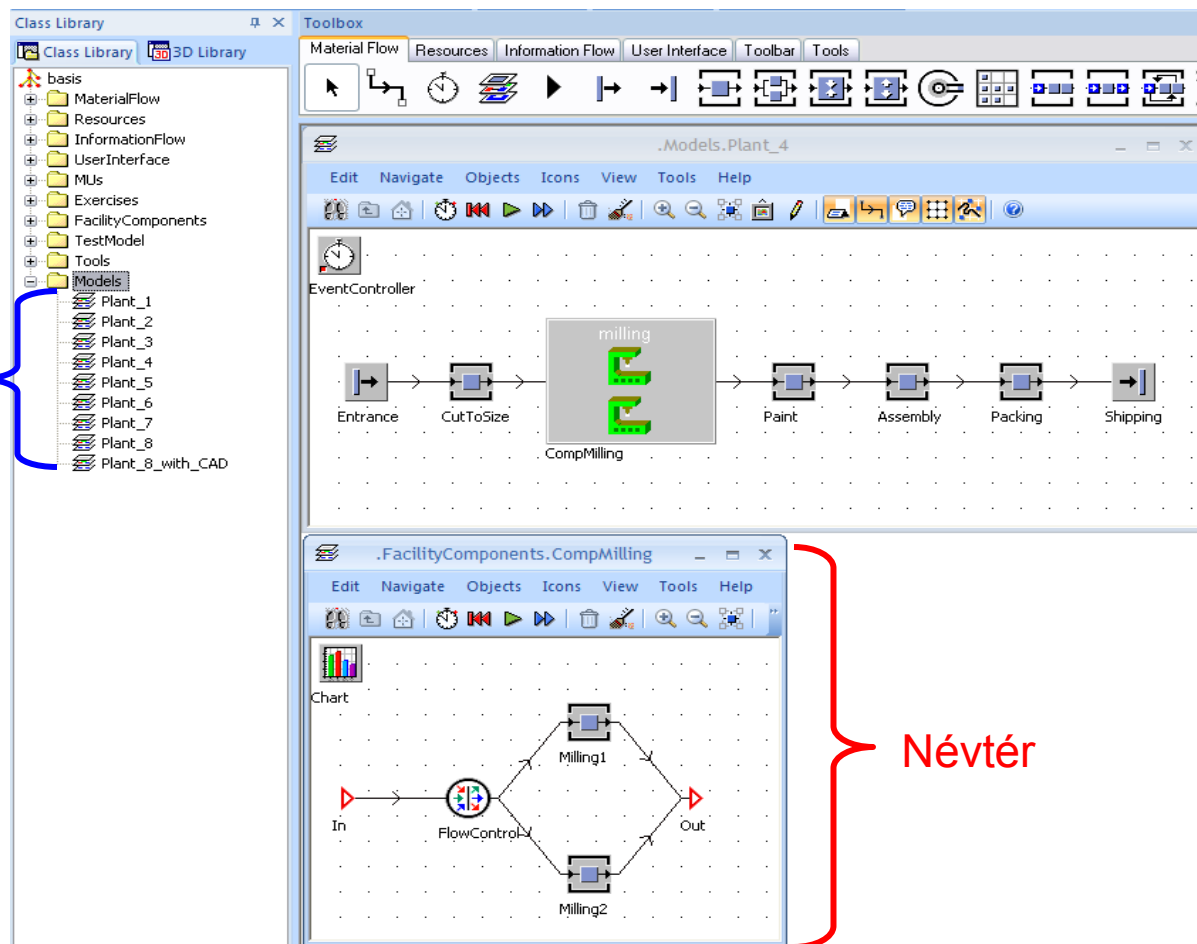
Minden objektum a hierarchia azonos szintjén azonos névtérben található. A névtérben belül minden objektumnak egyedi névnek kell lennie. Különböző névtérben lévő objektumoknak lehet azonos a nevük.

Névtér

Névtér

Névtér

Névtér



Névtér

Névtér

Névtér

Névtér

`.Models.Plant_4.CompMilling.Milling2`



- A pontos helyét adja meg a objektumnak tekintet nélkül a *Frame*-ben lévő hierarchiai szintjére.
- Más hierarchia szinten lévő objektumok hívása esetén szükséges.
- **Mindig a Class Library**-vel kezdődik, és végighalad a mappákon és *Frame*-eken az objektumig.
- Mindig ponttal kezdődik, majd utána név, majd pont, majd név, majd pont egészen addig, amíg az objektum nevét eléri:

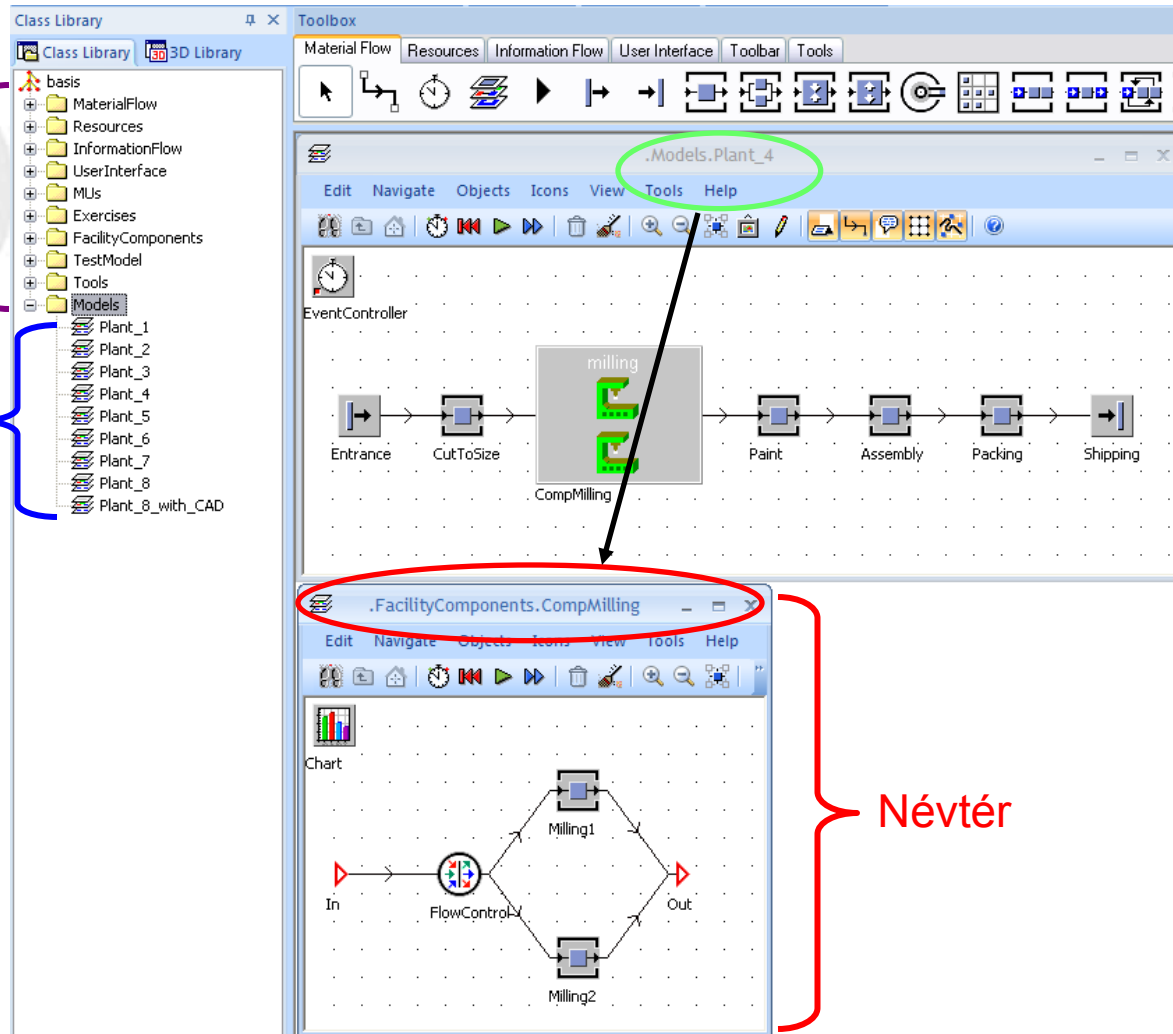
.<név>.<név>.<név>
- A hívott objektum nevét tartalmazza az utolsó pozícióban.
- A feldolgozásához idő kell.

Névtér

Névtér

Névtér

CompMilling.Milling2

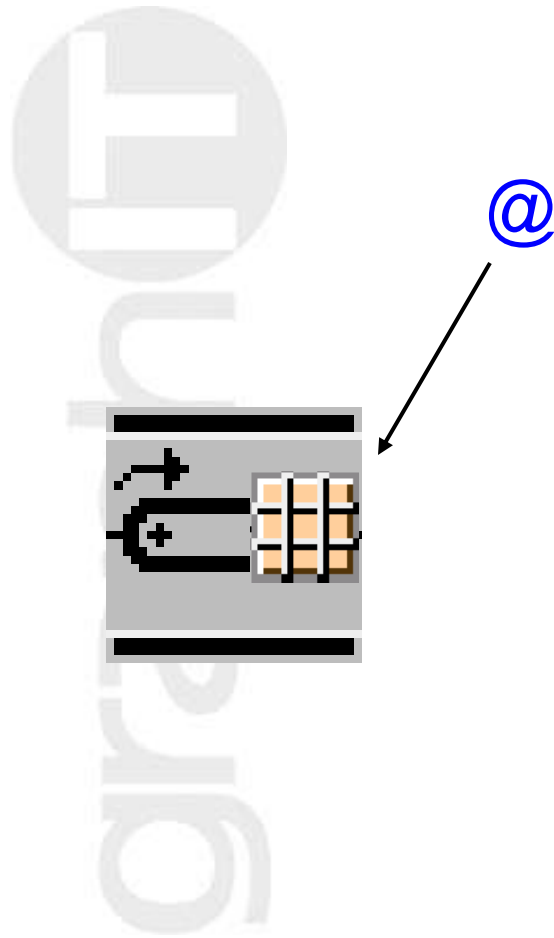


- Objektumot hív meg **azonos névtérből**, anélkül, hogy másik hierarchia szintre lépne.
- Soha nem kezdődik ponttal.
- Rövidebb a feldolgozási ideje, mint az abszolút útvonalnak.
- Lehetővé teszi metódusok és objektumok rugalmas és általánosan használható elkészítését.
- Frissül, ha változik a modell struktúrája.

Ha egy útvonalat egy objektumhoz a lehető leguniverzálisabbá akar tenni, és a lehető legrugalmasabbá, akkor lehet szükség a névtelen azonosítókra.

Erre a célra számos **névtelen azonosító** használható.

Amellett, hogy univerzálisak, pontosan azonosítanak egy adott objektumot.



A névtelen azonosító @ egyedileg azonosítja azt az MU-t, például **table_top:6712**, amely a metódust kiváltotta, amikor belépett az objektumba vagy kilépett az objektumból.

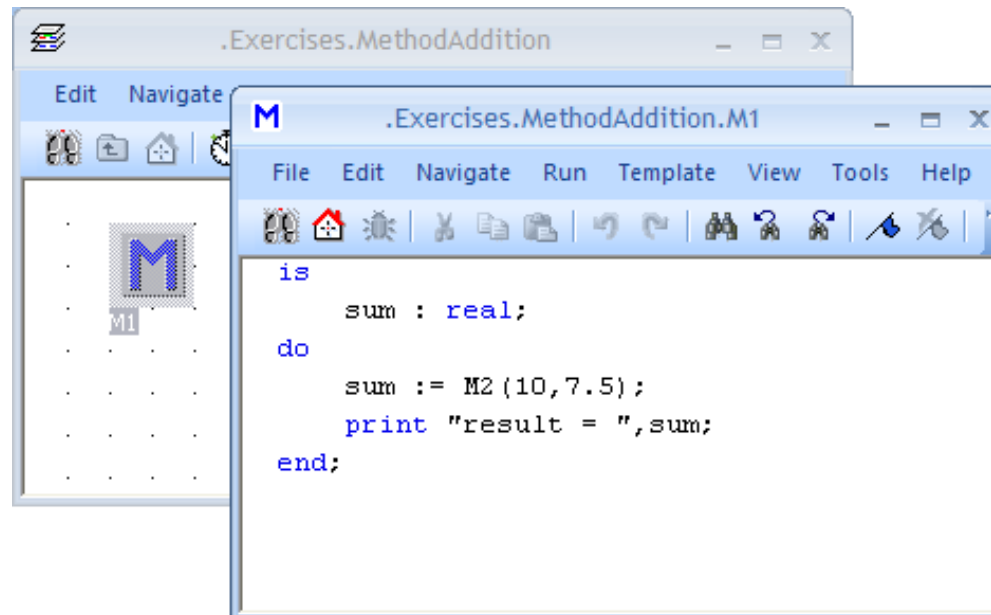
A Plant Simulation akkor is egyedileg jelöli az MU-kat, ha több objektum metódusai hívódnak meg. Az MU másik objektumon is lehet.

self

A self névtelen azonosító az éppen futó metódus útvonalával tér vissza:

self; visszaadja a metódus nevét és útvonalát
(*.Exercises.Frame.M1*)

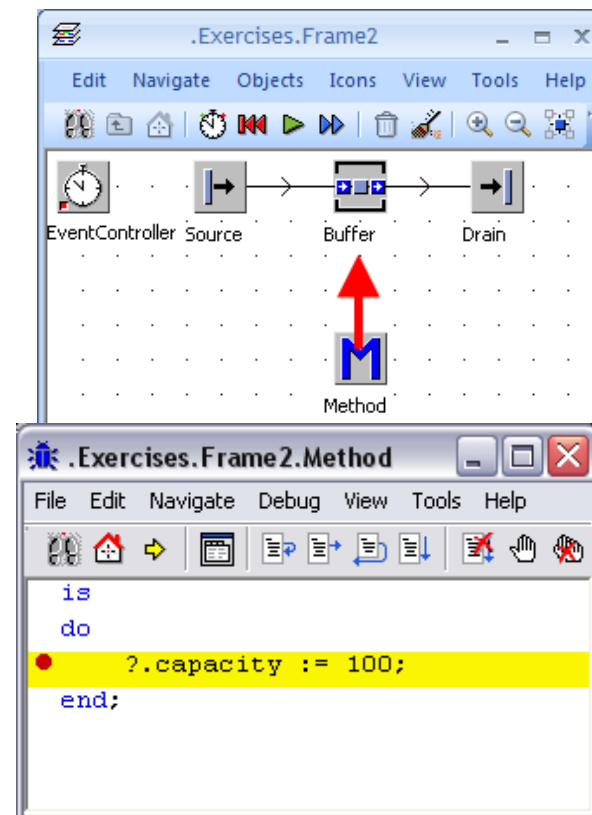
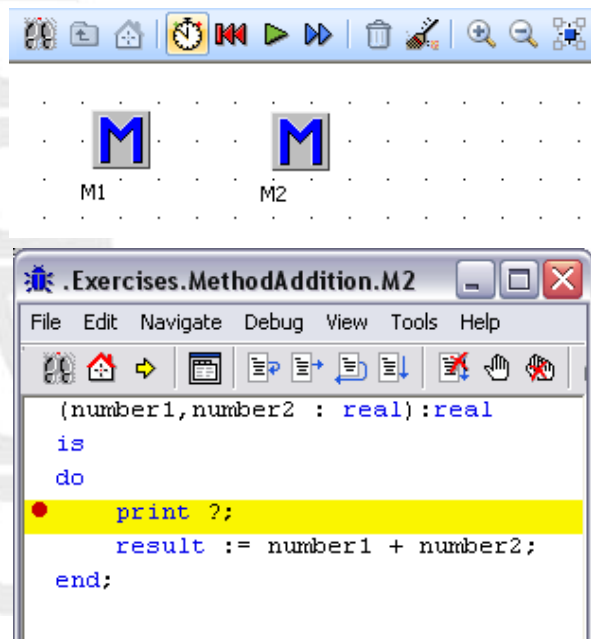
self.Name; visszaadja a metódus nevét (*M1*)



Objektum vagy metódus meghívója - ?

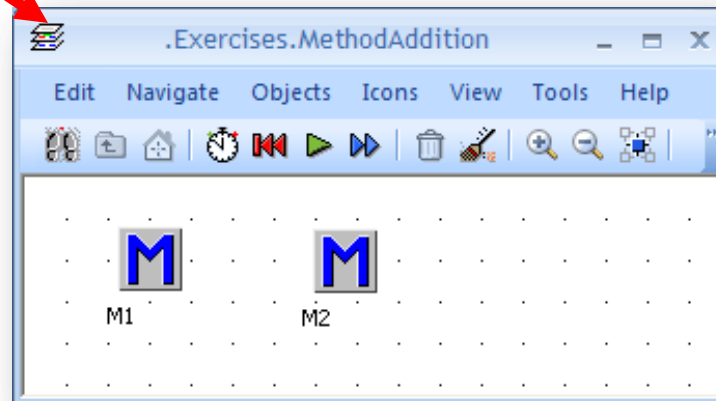
?

A ? névtelen azonosító azt az anyagáram objektumot vagy metódust jelenti, amely a metódust meghívta. A ? alkalmazásával lehetőség nyílik olyan metódusok készítésére, amelyek módosítás nélkül használhatók több objektummal is.



current

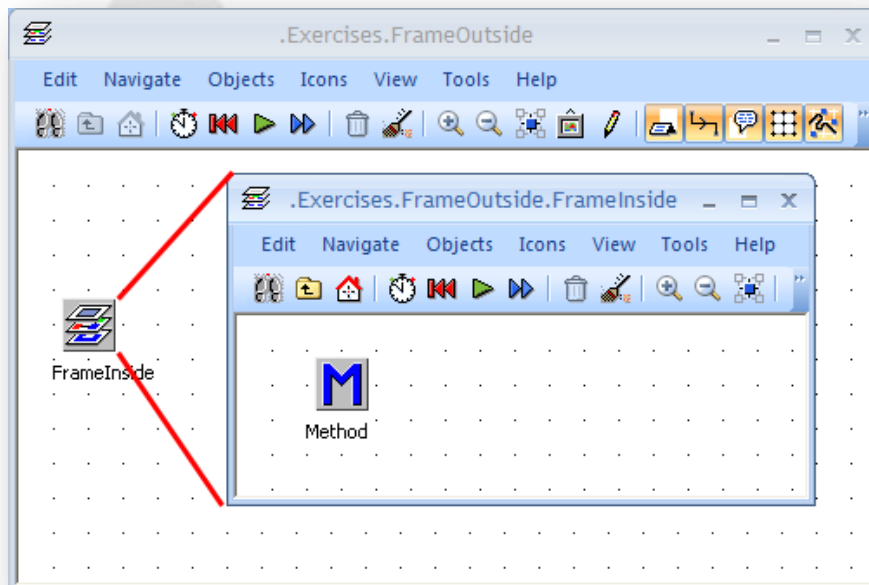
A **current** (jelenlegi) névtelen azonosító azzal a *Frame* - mel tér vissza, amelyen a metódus objektum található. Ezzel a módszerrel könnyen kinyerhető a metódus *Frame* - je, és átadható argumentumként más *Frame* - ek számára.



location vagy ~

A **location** vagy ~ metódus azzal az objektummal tér vissza, amelyik közvetlenül a <path> által kijelölt objektum fölött van. Az MU – ra ez az az objektum, amelyen található.

<object>.location azzal a *Frame* útvonallal tér vissza, amelyen az objektum található.

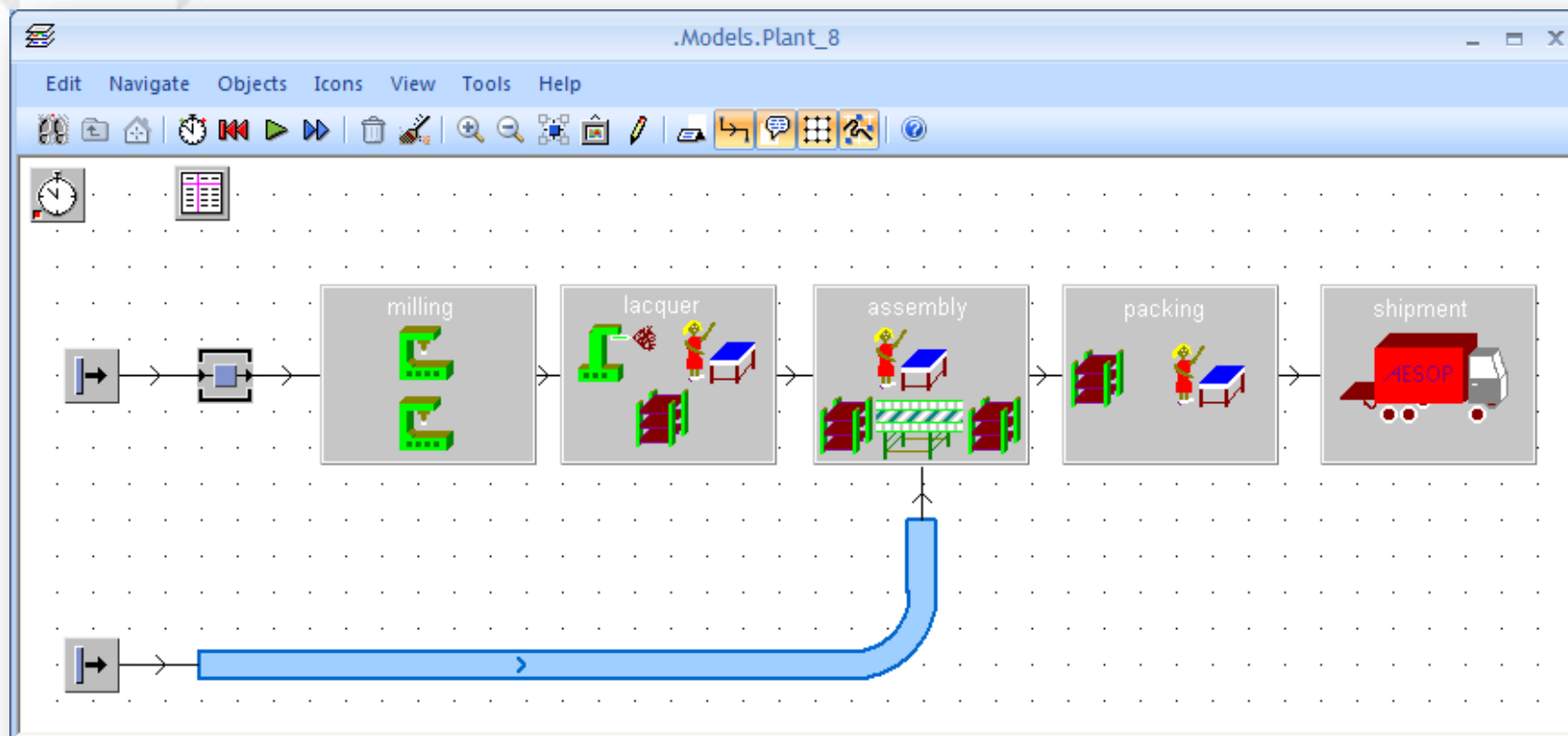


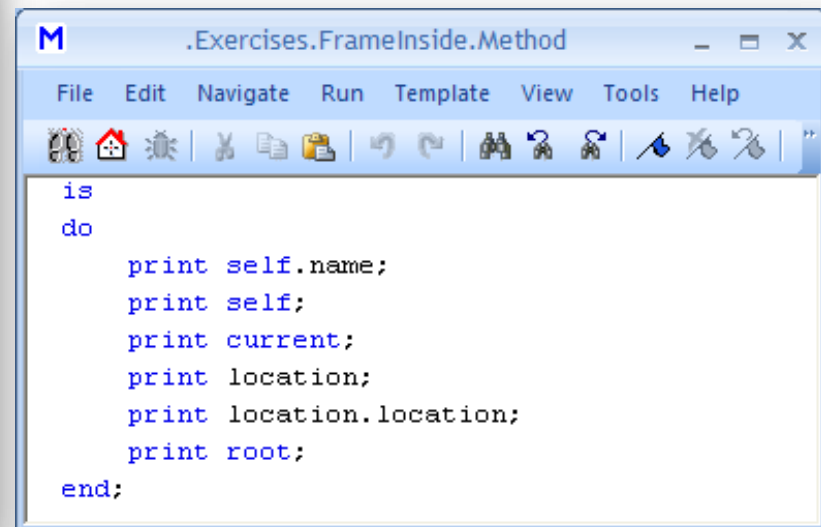
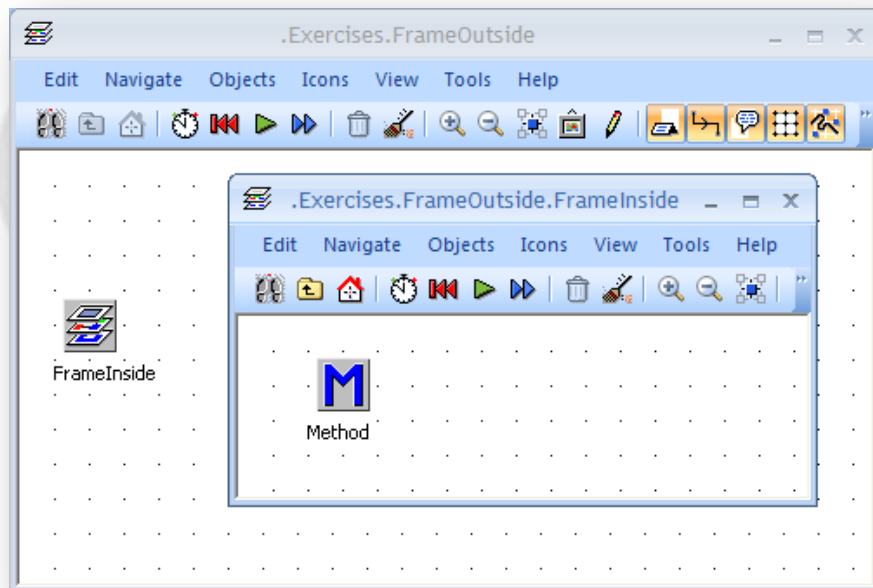
@. location az MU esetében a location az az objektum, amelyen az MU található.

A hierarchia legfelsőbb szintjének meghívása - root

root

A **root** (gyökér) névtelen azonosító a *Frame* hierarchiában a legfelső *Frame*. Innen lefelé lehet haladni a modell felé. Ez az azonosító különösen hasznos, a legfelső, gyökér *Frame* neve nem ismert.

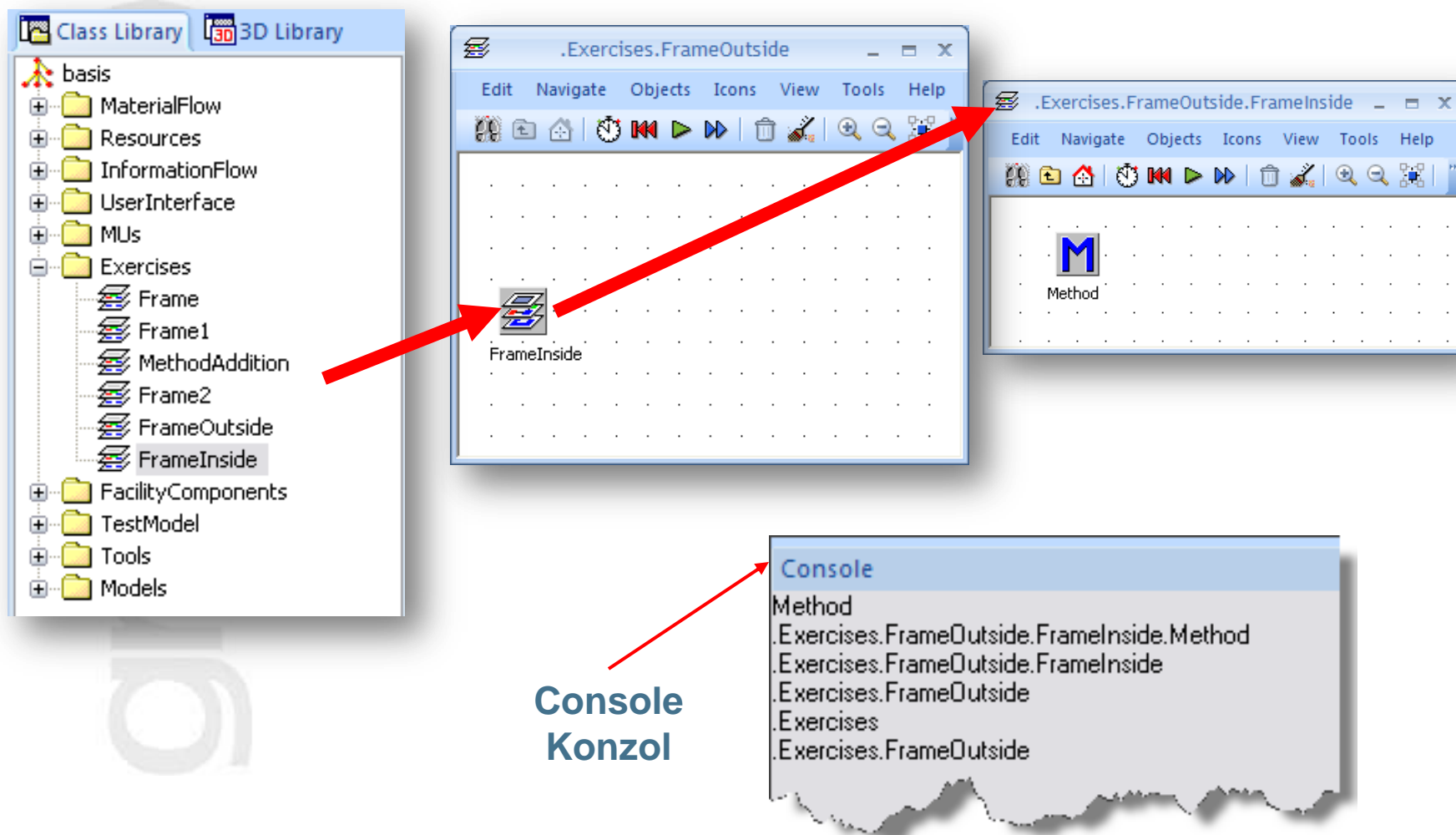




1. Adja hozzá a **FrameOutside** *Frame* - et az *Exercises* mappához.
2. Adja hozzá a **FrameInside** *Frame* - et ugyanahhoz a mappához.
3. Illesszen be egy *Method* objektumot a **FrameInside** *Frame* - be és írja be az ábrán látható programkódot.

4. Nyissa meg a **FrameOutside** *Frame* - et, és húzza bele a **FrameInside** *Frame* - et.
5. Figyelje meg a névtelen azonosítók értékeit a konzolon (*Console*).
6. Nyissa meg a metódust a **FrameInside** *Frame* - ben, futtassa lépésről-lépésre és figyelje meg amit a konzol megjelenít. Az értékek használhatók lokális változóknak, és megjeleníthetők a **Watch Window** ablakában a hibakeresőnek.

A metódusok egymásba ágyazása megmutatja a névtelen azonosítók használatát különböző hierarchia szinteken.



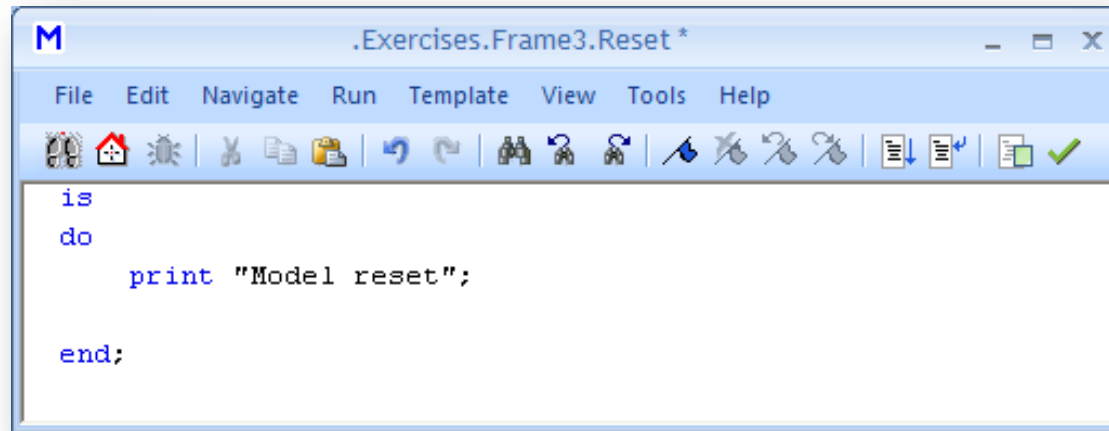


5. fejezet

Metódus hívások időzítése

Egy esemény, ami egy adott pillanatban történik a modellben meghívhat metódust:

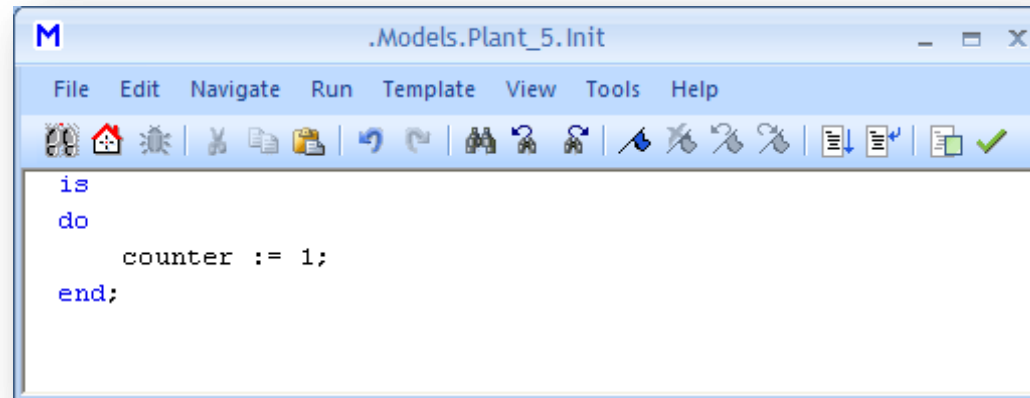
1. Az alábbi gombokra kattintva az *EventController* objektumban a következő metódusokat hívja meg:
Reset: meghívja a `reset` metódust a megnyitott modellben.
Init: meghívja az `init` metódust a model resetelése után.
2. A Plant Simulation a modell fájl megnyitásakor lefuttatja az *autoexec* metódust, aminek az osztálykönyvtárában kell lennie.
3. A Plant Simulation a szimuláció futásának végén az *endSim* metódust hívja meg. A szimuláció akkor ér véget, amikor az *EventController* feldolgozta az eseményeket a **List of Scheduled Events** listából, vagy amikor a Plant Simulation eléri azt az időpontot, ami az **End** mezőben meg lett adva a **Settings** fülön.
4. A *Trigger* és a *Generator* objektumok szintén meg tudnak hívni metódust.



```
is
do
    print "Model reset";
end;
```

Az *EventController* **Reset** gombjára kattintva a Plant Simulation végigkeresi a teljes modellt, beleértve az összes *Frame*-et, **reset** nevű metódusokért és lefuttatja azokat.

A **Reset** metódus használható változók, számlálók értékeinek alapértékre állítására.



Az *EventController Init* gombjára kattintva a Plant Simulation lefuttatja az összes **init** nevű metódust a modellben. Az **init** metódusok a modell inicializálására szolgálnak, például változók beállítása, MU-k elhelyezése az anyagáram objektumokon stb.

Illesszen be egy metódust a modellbe, nevezze át (**Rename**), és a metódus neveként adja meg az *init* nevet.



6. fejezet

Anyagáram objektumok által meghívott
metódusok

Minden anyagáram objektumra meg lehet adni egy **Entrance** (belépési) és/vagy egy **Exit** (kilépési) kontrollt. Ez hasonló a fotocellához, ami az anyagáram objektum elejénél és végénél van elhelyezve.

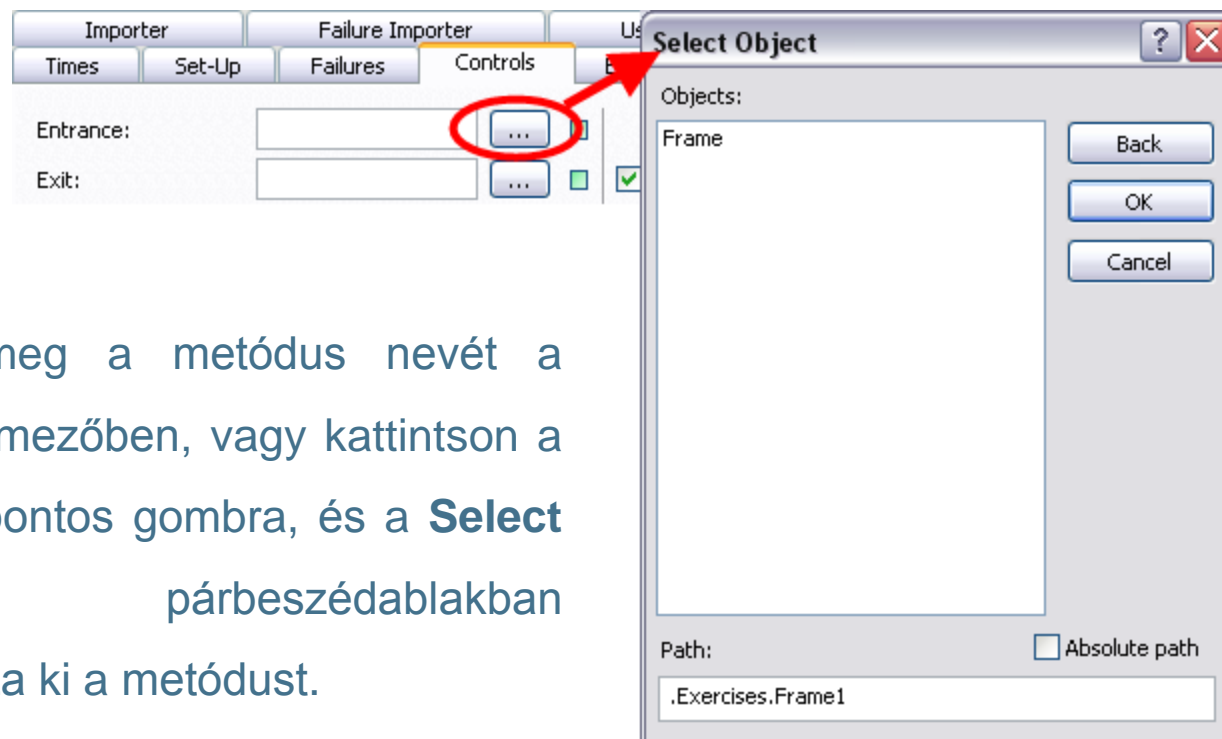
Minden ilyen fotocellához hozzárendelhető egy metódus, amit a Plant Simulation lefuttat, amikor az MU belép az objektumba, vagy kilép az objektumból.

Entrance control

Exit control

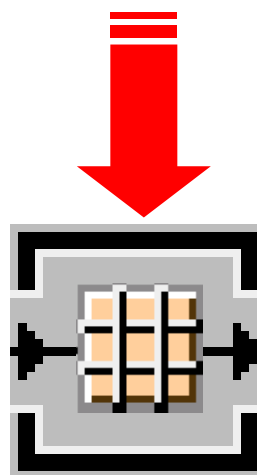
Különbséget lehet tenni a **helyorientált** és a **hosszorientált** objektumok között.





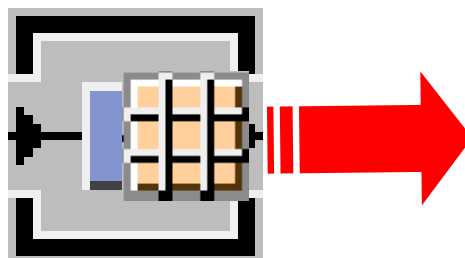
Adja meg a metódus nevét a szövegmezőben, vagy kattintson a hárompontos gombra, és a **Select Object** párbeszédablakban válassza ki a metódust.

Aktiválásra kerül, amikor az MU belép az objektumba, az MU végig nyilván van tartva az objektumban, függetlenül annak hosszától.



Miután meg lett adva nem lehet változtatni azt az időt, amit az MU az objektumon tölt, kivéve, ha a feldolgozási idő képlettel van megadva.

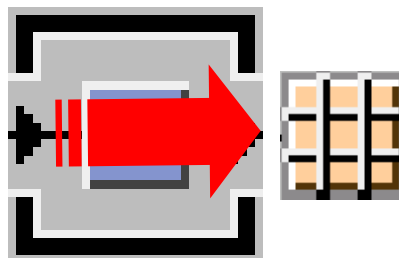
Akkor **hívódik meg**, amikor az MU kilépni szándékozik az objektumból. Az MU még az objektumon tartózkodik.



Az elől aktivált kilépési kontroll **felülírja az alapértelmezett kilépési stratégiáját a Plant Simulation** - nek. Ezért az MU továbbmozgatását meg kell oldani.

A kilépési kontroll többször aktivizálható: ha az MU nem tud kilépni az objektumból, például mert a kimeneti ág tele van, a Plant Simulation addig hívja meg a kilépési kontrollt, amíg a successor szabaddá válik.

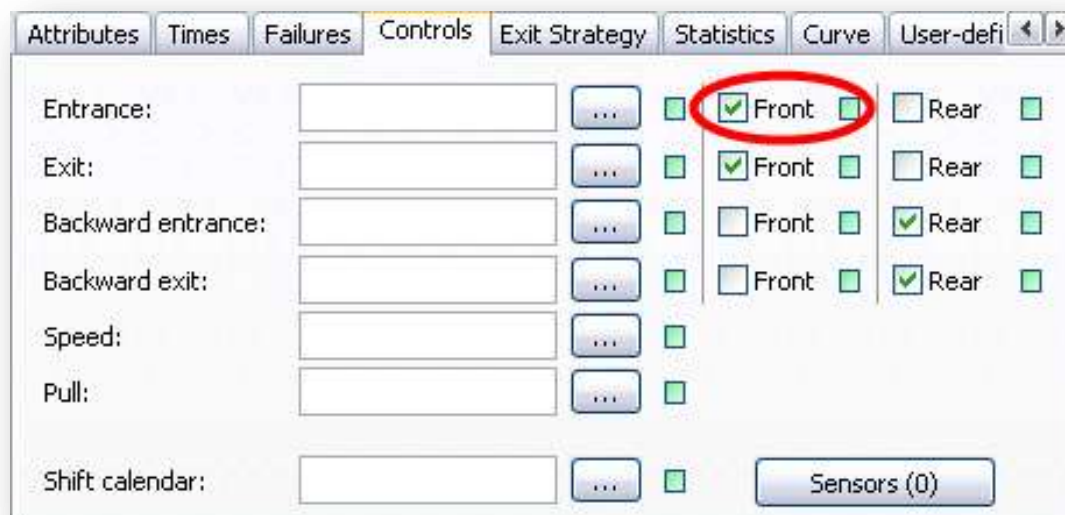
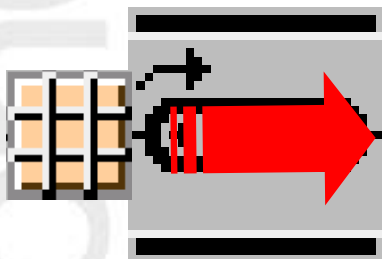
Akkor **hívódik meg**, amikor az MU elhagyja az objektumot, az MU a successor - on található.



A hátul aktivált kilépési kontroll egyszer hívódik csak meg.

Nem írja felül az alapértelmezett kilépési stratégiát.

Akkor **hívódik meg**, amikor az MU be akar lépni az objektumra. Az MU eleje már az objektumon van, amikor a metódus meghívódik.

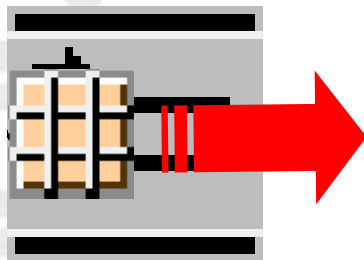


Attributes	Times	Failures	Controls	Exit Strategy	Statistics	Curve	User-defi
Entrance:			<input checked="" type="checkbox"/> Front	<input type="checkbox"/> Rear			
Exit:			<input checked="" type="checkbox"/> Front	<input type="checkbox"/> Rear			
Backward entrance:			<input type="checkbox"/> Front	<input checked="" type="checkbox"/> Rear			
Backward exit:			<input type="checkbox"/> Front	<input checked="" type="checkbox"/> Rear			
Speed:			<input type="checkbox"/>				
Pull:			<input type="checkbox"/>				
Shift calendar:			<input type="checkbox"/>				Sensors (0)



Miután meg lett adva nem lehet változtatni azt az időt, amit az MU az objektumon tölt, kivéve, ha a feldolgozási idő képlettel van megadva.

Akkor hívódik meg, amikor az MU teljes hossza az objektumon található.



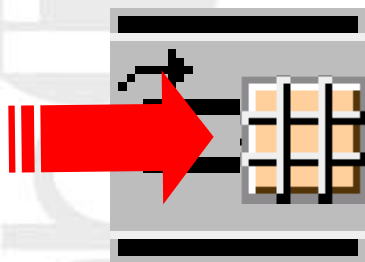
Attributes	Times	Failures	Controls	Exit Strategy	Statistics	Curve	User-defi
Entrance:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear
Exit:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Front <input type="checkbox"/> Rear	<input checked="" type="checkbox"/> Front <input type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear
Backward entrance:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear
Backward exit:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear	<input type="checkbox"/> Front <input checked="" type="checkbox"/> Rear
Speed:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>				
Pull:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>				
Shift calendar:	<input type="text"/>	<input type="button" value="..."/>	<input checked="" type="checkbox"/>				

Sensors (0)



Az MU objektumra érésének pillanata, és annak a pillanatnak a különbsége, amikor a hátul aktivált belépési kontroll aktiválódik a különbsége, az az idő, ami ahhoz szükséges, hogy az MU teljesen az objektumra mozogjon.

Akkor **aktivizálódik**, amikor az MU el szándékozik hagyni az objektumot. Az MU még teljesen az objektumon található.



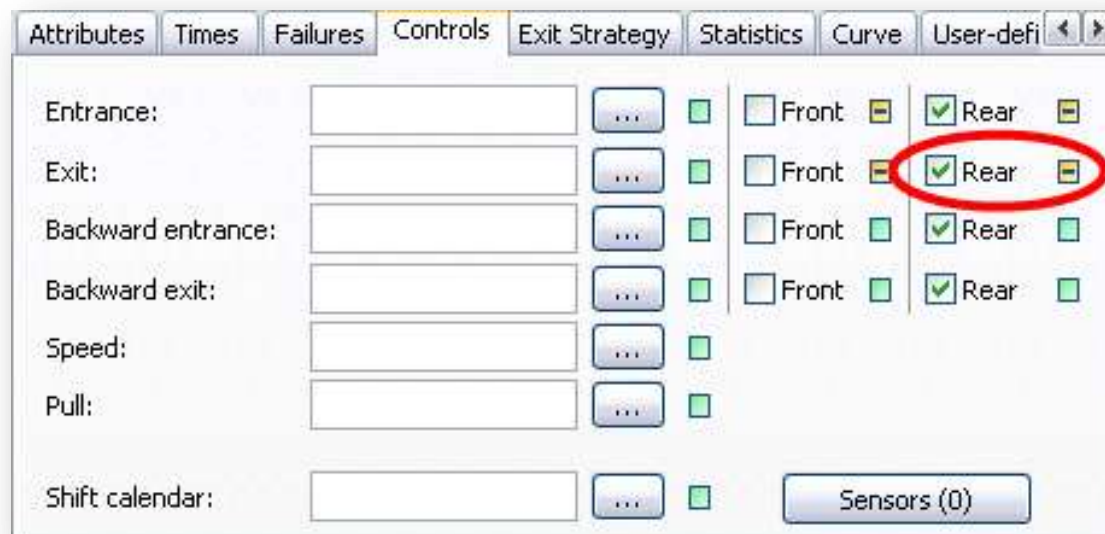
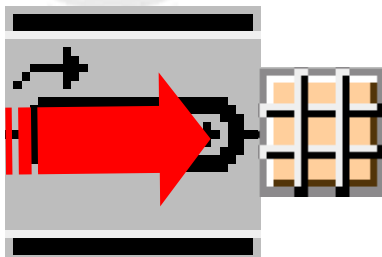
	Entrance:	Exit:	Backward entrance:	Backward exit:	Speed:	Pull:	Shift calendar:
Front	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rear	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Az elől aktivált kilépési kontroll **felülírja az alapértelmezett kilépési stratégiáját a Plant Simulation - nek**. Ezért az MU továbbmozgatását meg kell oldani.

A kilépési kontroll többször aktivizálható: ha az MU nem tud kilépni az objektumból, például mert a kimeneti ág tele van, a Plant Simulation addig hívja meg a kilépési kontrollt, amíg a successor szabaddá válik.

Akkor **hívódik meg**, amikor az MU elhagyja az objektumot. Az MU a kilépési objektumon található.

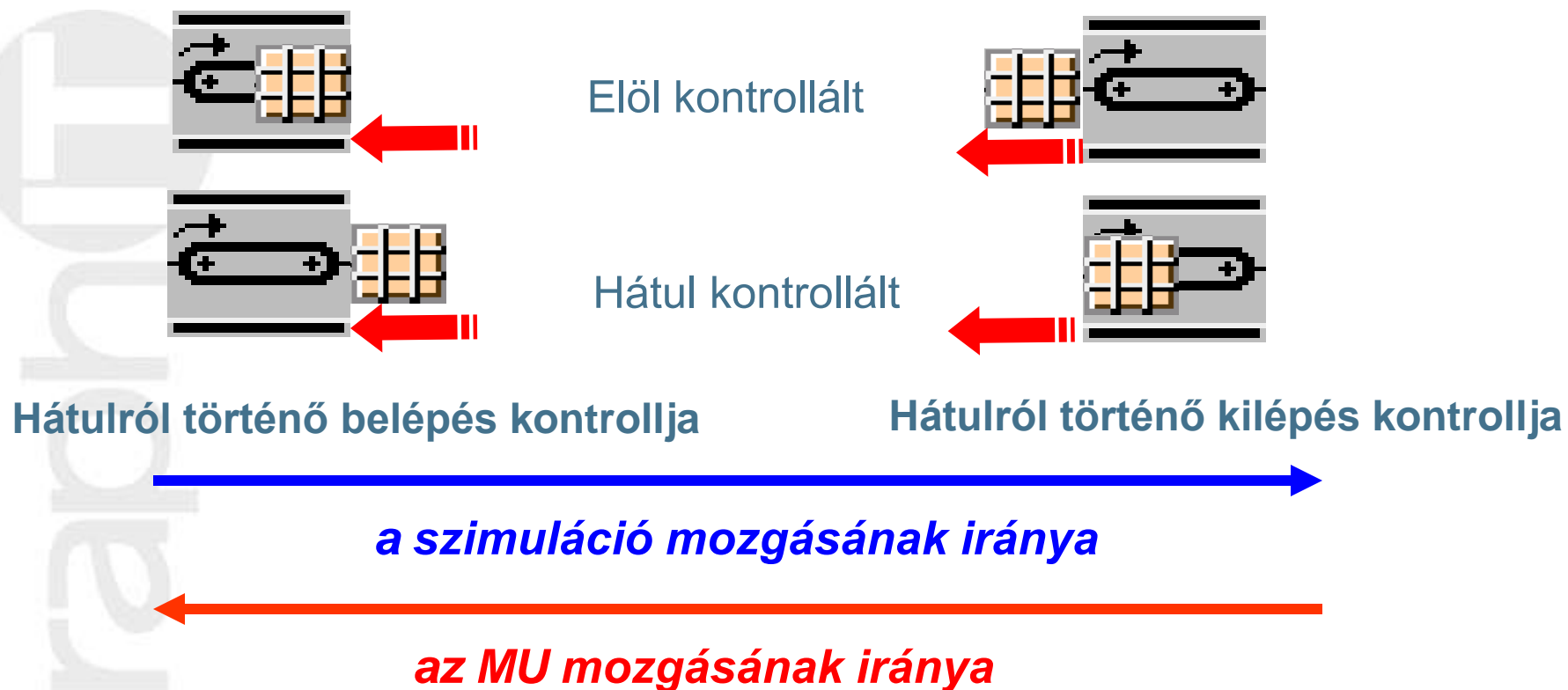


Attributes	Times	Failures	Controls	Exit Strategy	Statistics	Curve	User-defi
Entrance:			<input checked="" type="checkbox"/> Front <input checked="" type="checkbox"/> Rear				
Exit:			<input checked="" type="checkbox"/> Front <input checked="" type="checkbox"/> Rear				
Backward entrance:			<input checked="" type="checkbox"/> Front <input checked="" type="checkbox"/> Rear				
Backward exit:			<input checked="" type="checkbox"/> Front <input checked="" type="checkbox"/> Rear				
Speed:			<input checked="" type="checkbox"/>				
Pull:			<input checked="" type="checkbox"/>				
Shift calendar:			<input checked="" type="checkbox"/>				

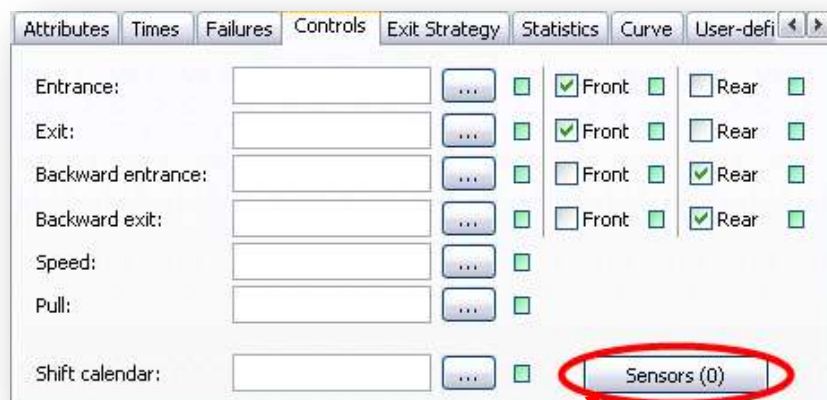


A hátul aktivált kilépési kontroll egyszer hívódik csak meg.

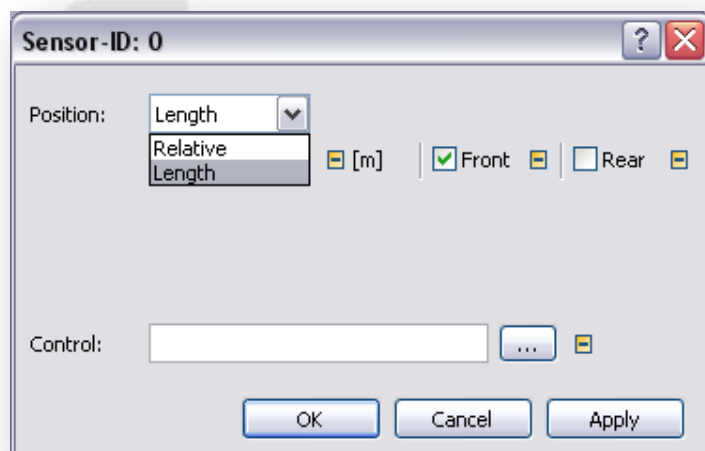
Nem írja felül az alapértelmezett kilépési stratégiát.



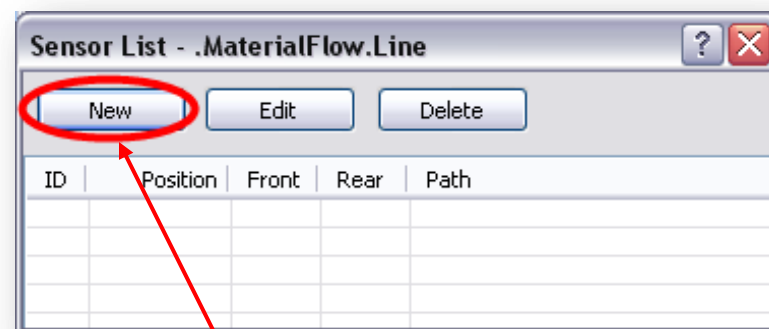
Az MU nem változtatja meg a helyzetét, pl. nem fordul el maga körül, és nem mozog az elejével a szimuláció mozgásával ellentétes irányba.



Kattintson a **Sensors** gombra a szenzorok listájának megnyitásához.



A hosszorientált objektumokhoz szenzorokat lehet a hosszuk mentén tetszőleges helyen definiálni.



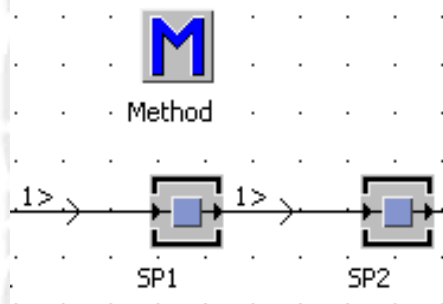
Kattinon a **New** gombra a **Sensor** párbeszédablak megjelenítéséhez. Adja meg a beállításokat az ablakban.

A **move** metódus mozgatja az MU-t az egyik objektumról a másikra. A Plant Simulation a teljes MU-t a hely orientált objektumra mozgatja. A Plant Simulation az MU-t a hossz-orientált objektumra a megadott sebességgel mozgatja.

SP2-re mozgatás:

```
@.move;
```

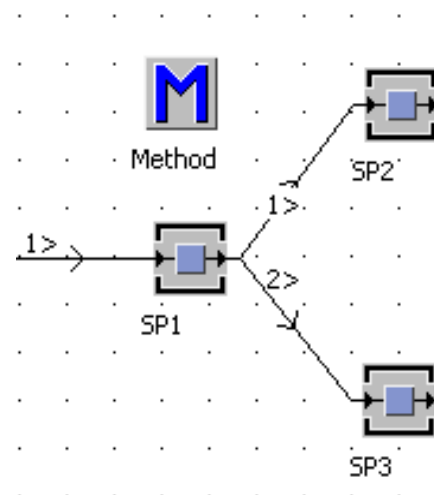
```
@.move (SP2) ;
```



SP2-re mozgatás: SP3-ra mozgatás:

```
@.move (SP2) ;    @.move (SP3) ;
```

```
@.move (1) ;    @.move (2) ;
```



insert

Az **insert** metódus a teljes MU-t átmozgatja a célobjektumra. A célobjektumnak elég helyel kell rendelkeznie az MU számára.

Példa: `@.insert(track, 3.3) ;`

transfer

A **transfer** metódus az MU-t a következő anyagáram objektumra mozgatja. Ez a metódus teljesen eltávolítja az MU-t a jelenlegi objektumról, akkor is, ha a successor nem rendelkezik elég helyel.

Példa: `@.transfer(track, 3.3) ;`

A **move**, **insert**, **transfer** metódusok visszatérése true, ha az MU átmozgatódott, false, ha nem.

Metódusok használata:

`succ(<integer>)`

`pred(<integer>)`

objektumok predecessor - ának és/vagy successor - ának tesztelésére. Ez hasznos kilépési stratégiák modellezésekor.

A meghívott objektum útvonalával kezdve a hivatkozást, a metódus annak successor - ához vagy predecessor - ához fér hozzá, visszaadva a hozzá kapcsolódó objektumot. A szám a *Connector* sorszáma.

Példa:

```
@.move(Milling.pred(1));
```

```
@.move(?.succ(1).succ(1));
```

```
if Milling.pred(1).name = "CutToSize" then
```

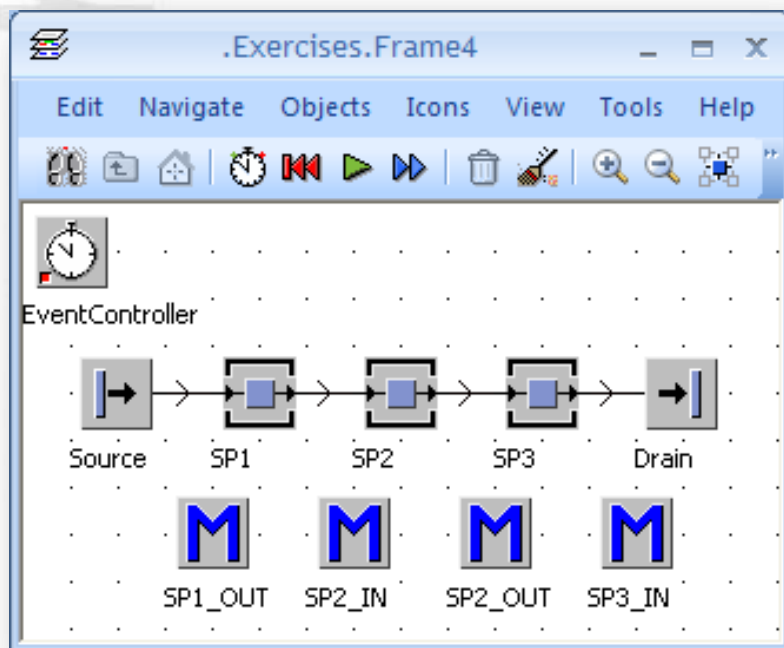
```
...
```



3a. lecke: SingleProc be- és kilépési kontrollok

Illessze be a az ábrán látható metódusokat, és figyelje meg az eredményt a konzol ablakban:

1. Adja hozzá a **Frame4 Frame** - et az *Exercises* mappához. Adjon meg 5 percet az MU-k készítési időközéneként.
2. Írja be a print **self.name** kódot a metódusokba, és válassza ki az objektumok kontrolljait. Ne feledje el, hogyan működik az elől aktivizált kilépési kontroll.



Figyelje meg a metódus hívásakor:

SP1_out: elől aktivizált kilépési kontroll
SP1

SP2_in: belépési kontroll SP2

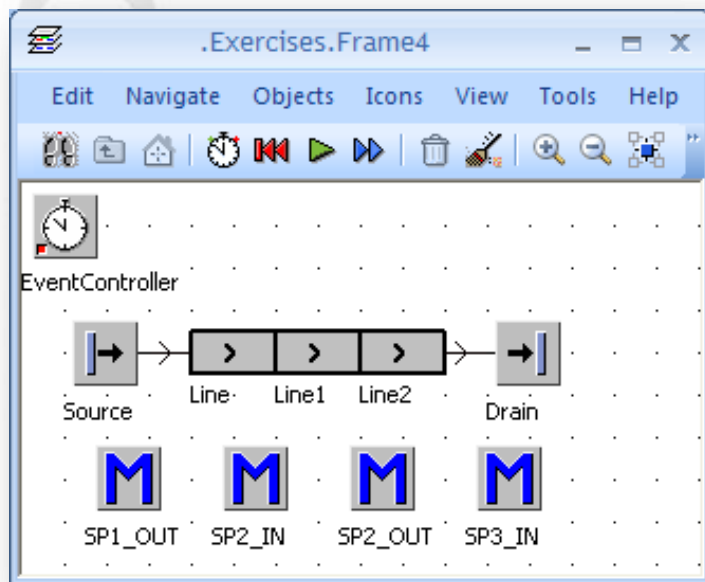
SP2_out: hátul aktivizált kilépési kontroll
SP2

SP3_in: belépési kontroll SP3

3b. lecke: Line be- és kilépési kontrollok

Figyelje meg, hogyan viselkednek a belépési és kilépési kontrollok a hosszorientált objektumokra.

1. Másolja át a **Frame4** *Frame* tartalmát és másolja be azt a **Frame5** *Frame*-be.
2. Cserélje ki a *SingleProc* objektumokat *Lines* objektumokkal és adja meg a metódusokat a lentieknek megfelelően.
3. Indítsa el a szimulációt és figyelje meg, hogy mit mutat a konzol ablak.



SP1_out: elől aktivizált kilépési kontroll SP1

SP2_in: hátul aktivizált belépési kontroll SP2

SP2_out: hátul aktivizált kilépési kontroll SP2

SP3_in: elől aktivizált belépési kontroll SP3



7. fejezet

Kifejezések

Válassza ki a Class Library helyi menüjéből a **Show Attributes and Methods** parancsot az osztály attribútumainak/metódusainak megjelenítéséhez.

A kiválasztott példány attribútumait/metódusait a *Frame* menüjének **Objects > Show Attributes and Methods** parancsával lehet megjeleníteni.

Name	Signature	Value	i.	W
addObserver	(string,method)			
attributeWatchable	(string) : boolean			
BookPnt	length	0.4	ni	*
BookPntL	length	0.4	ni	*
BookPntW	length	0.4	ni	*
BookPos	length	0		
ChangePathCtrl	method	VOID	ni	
ChannelID	string	3d	ni	*
ChildNo	(integer) : object			
Class	object	VOID		
closeDialog	([boolean]) : boolean			
closeImg	(boolean) : boolean			
Collided	boolean	false		*
connectAutomatically	boolean			
ConstructorCtrl	method	VOID	ni	
ConveyingDirection	integer	0	ni	*
copyToClipboard				
create	(object[,length]) : object			
createAttr	(string,string) : boolean			
createIcon	([string,integer,integer]) : i...			
createObject	(object,integer,integer[,stri...			
CreationTime	time	Creation...		
CurrIcon	string	Operational	ni	*
CurrIconNo	integer	0	ni	*
CurrIconTransparent	boolean	true	ni	
delete				
deleteAttr	(string) : boolean			

Value (érték). Csak az attribútumoknak van értéke. Dupla kattintás szerkesztéshez.

Azt jelzi, hogy a Plant Simulation tudja-e mutatni az attribútum értékét vagy állapotát (pl. gyakran változó értékeket nem)

Azt mutatja, hogy az érték örökölt (i) vagy nem örökölt (ni)

Az attribútum vagy metódus neve

Az attribútum adattípusa, vagy a paraméterek adattípusa

Értéket az értékadás művelet ($:=$) jobb oldalára megadott értékkel lehet adni.

$\langle \text{kifejezés} \rangle := \langle \text{érték} \rangle;$

Példa:

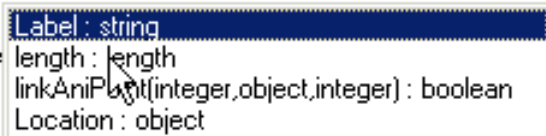
```
buffer.capacity := 8;  
singleProc.Pause := true;  
@.curricon := "TableTop";
```

Ha szövegnek ad értéket, akkor idézőjelek közé kell tenni az adott szöveget.

Az automatikus kiegészítés a forráskód gépelése során használható: be kell gépelni az első vagy első két karakterét az objektumnak, attribútumnak vagy metódusnak, és az **Edit > Auto Complete** menüpontot kiválasztva vagy a **Ctrl + Space** billentyűket lenyomva működik az automatikus kiegészítés.

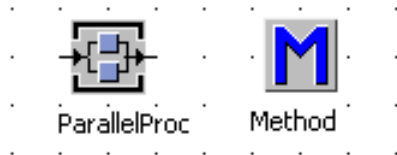
A Plant Simulation kiegészíti a szót. Ha szükséges hozzáadja a megfelelő karaktereket (pont, vessző, bezáró zárójel, értékadás művelet, stb.).

```
is
do
  Line.1
  print
  @.move
end;
```



Ha a név nem egyedi, a Plant Simulation egy listát nyit, ahonnan kiválasztható egy elem. A le-fel nyilakkal lehet lépkedni, az **Enter** billentyűvel pedig kiválasztani a kívánt kifejezést.

4. lecke: Objektum attribútumának módosítása metódussal



1. Adja hozzá a **ChangeAttributes** *Frame* - et az *Exercises* mappához.
2. Illesszen be egy *ParallelProc* és egy *Method* objektumot.
3. Módosítsa a *Method* segítségével a *ParallelProc* megfelelő attribútumait:

- **Processing time:** 120 sec (feldolgozási idő 120 másodperc)
- **X-Dimension:** 5, **Y-Dimension:** 3 (x irányú méret: 5, y irányú méret: 3)
- Zárolja a *ParallelProc* belépését (**Entrance locked**)
- Ellenőrizze le a **Show Attributes and Methods** ablakban, hogy az értékek a megfelelő módon megváltoztak.

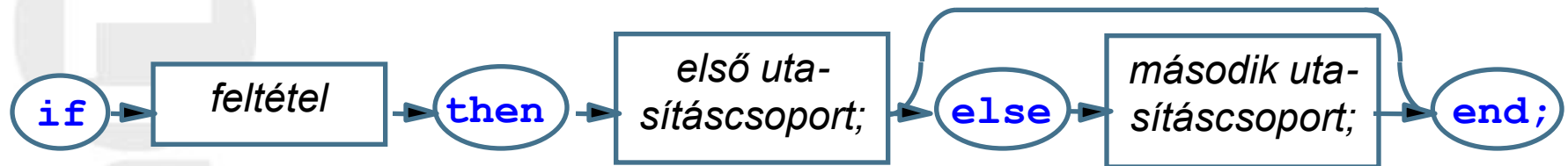
Módosítsa a *ParallelProc* objektum attribútumait a **Show Attributes and Methods** ablakban.

```
1: is
2: do
3:     parallelProc.procTime := 120;
4:     parallelProc.xDim := 3;
5:     parallelProc.yDim := 5;
6:     parallelProc.entranceLocked := true;
7: end;
```



8. fejezet

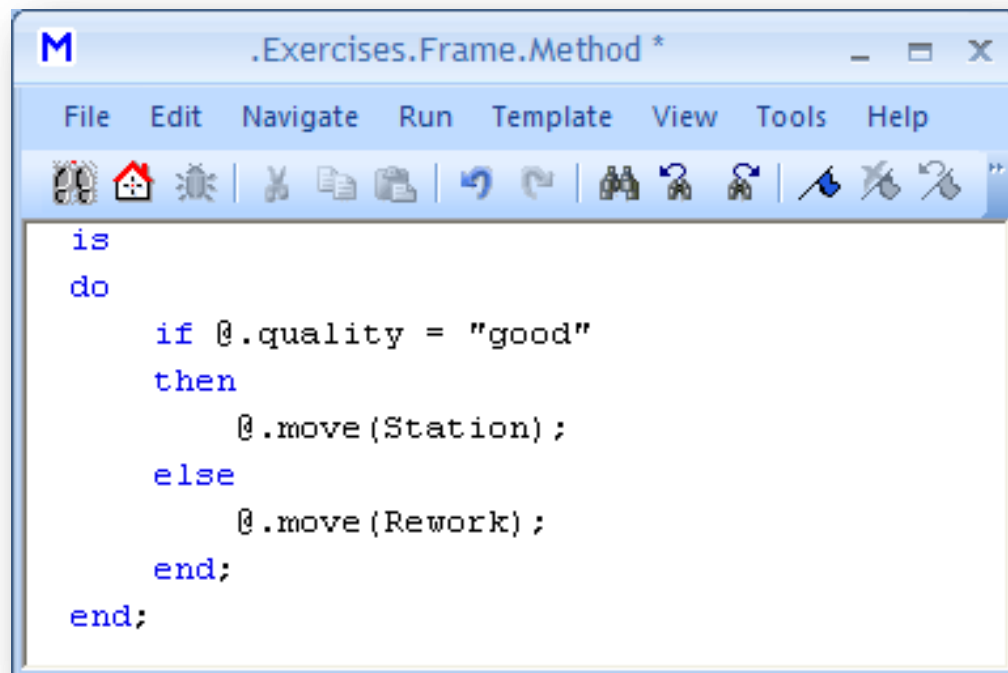
Feltételes kifejezések - if, inspect



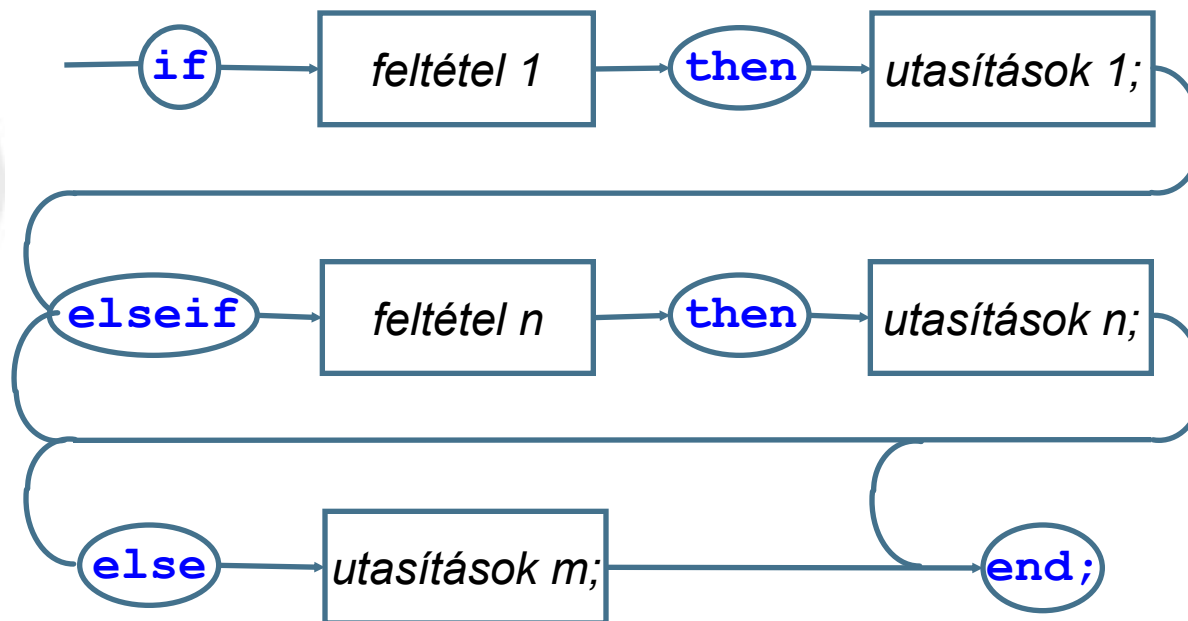
Az **if** kifejezést akkor érdemes használni, ha a metódus további futása egy feltétel eredményétől függ.

Ha a feltétel eredménye true (igaz), akkor a metódus az első utasításcsoporttal folytatja a futást, ha a feltétel eredménye false (hamis), akkor a metódus a második utasításcsoporttal folytatja a futást.

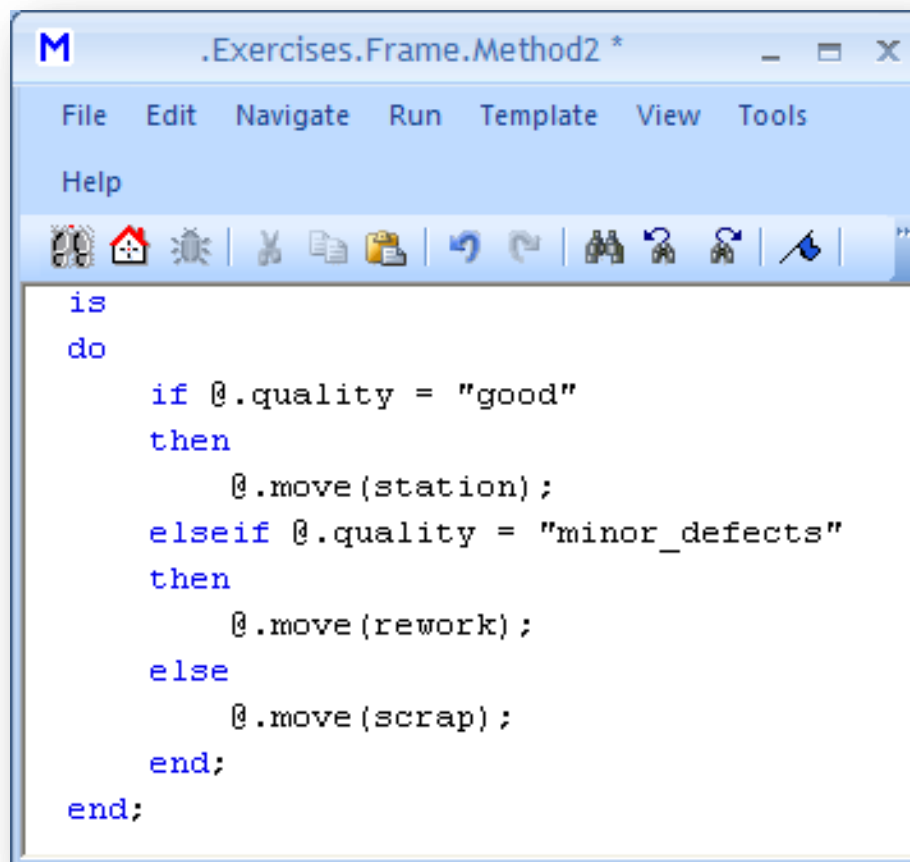
Ha nem szükséges, a második utasításcsoport elhagyható.



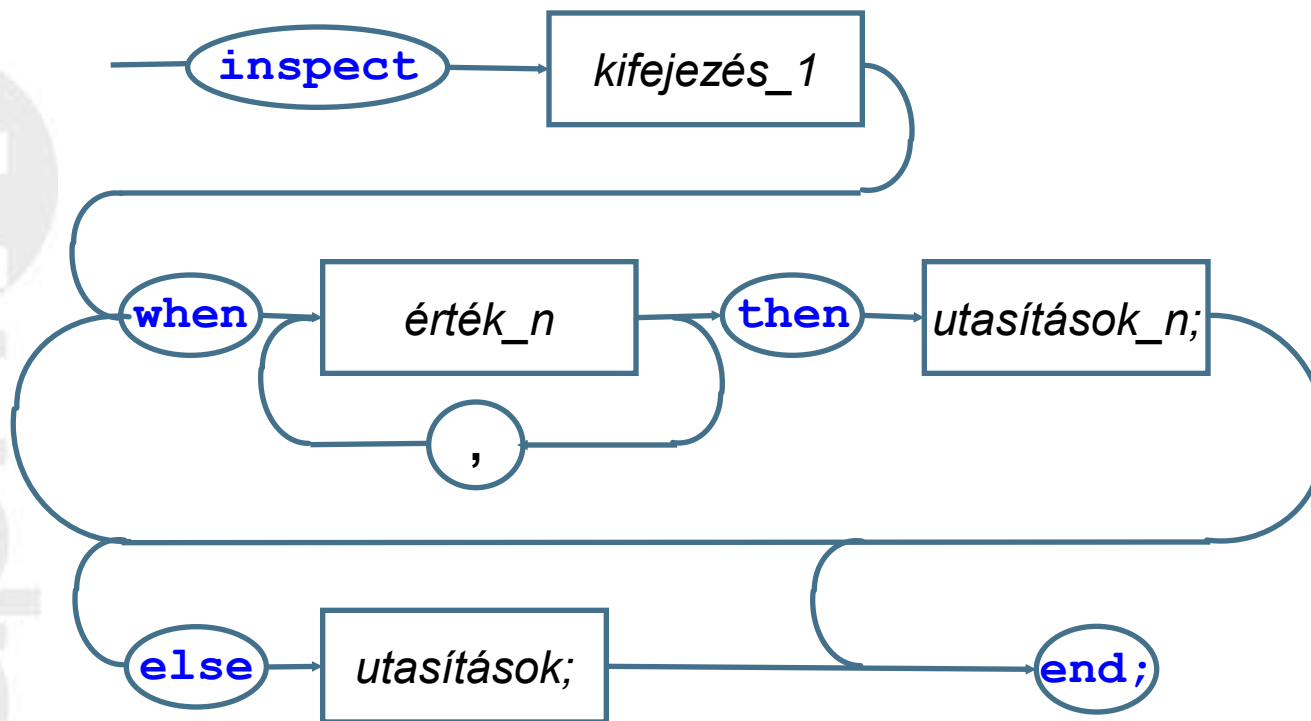
```
is
do
    if @.quality = "good"
    then
        @.move(Station);
    else
        @.move(Rework);
    end;
end;
```



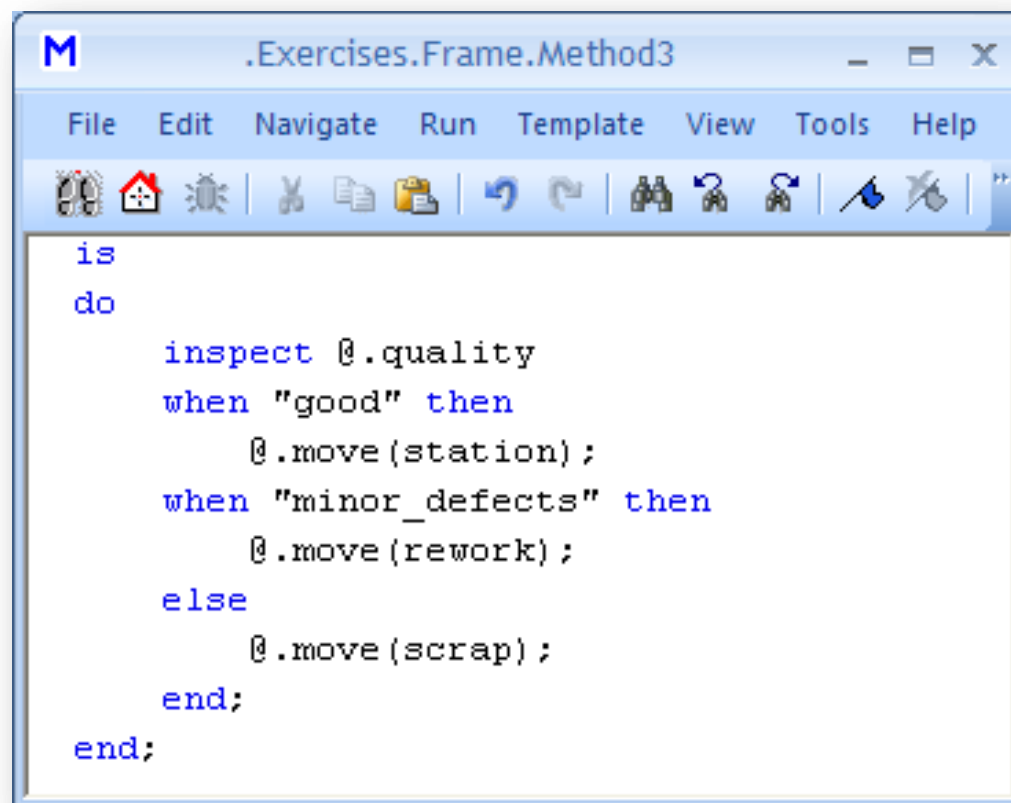
Az **if then elseif** kifejezés használatával a metódus különböző feltételekre tud reagálni. A Plant Simulation kiértékeli a feltételeket – egyiket a másik után – addig, amíg az egyik visszatérése true (igaz) lesz. Ez után lefuttatja a feltételhez tartozó utasításokat. Ha nincs igaz ág, akkor az opcionális **else** ág fut le. Ha **else** ág nincs, akkor a Plant Simulation folytatja a kódot a lezáró **end** kifejezés után.



```
is
do
    if @.quality = "good"
    then
        @.move(station);
    elseif @.quality = "minor_defects"
    then
        @.move(rework);
    else
        @.move(scrap);
    end;
end;
```



Az **inspect** kifejezés lehetővé teszi több választási lehetőség közüli választás egyszerűbb leírását. Így nem kell hosszú **if-then-elseif** utasításláncokat írni.

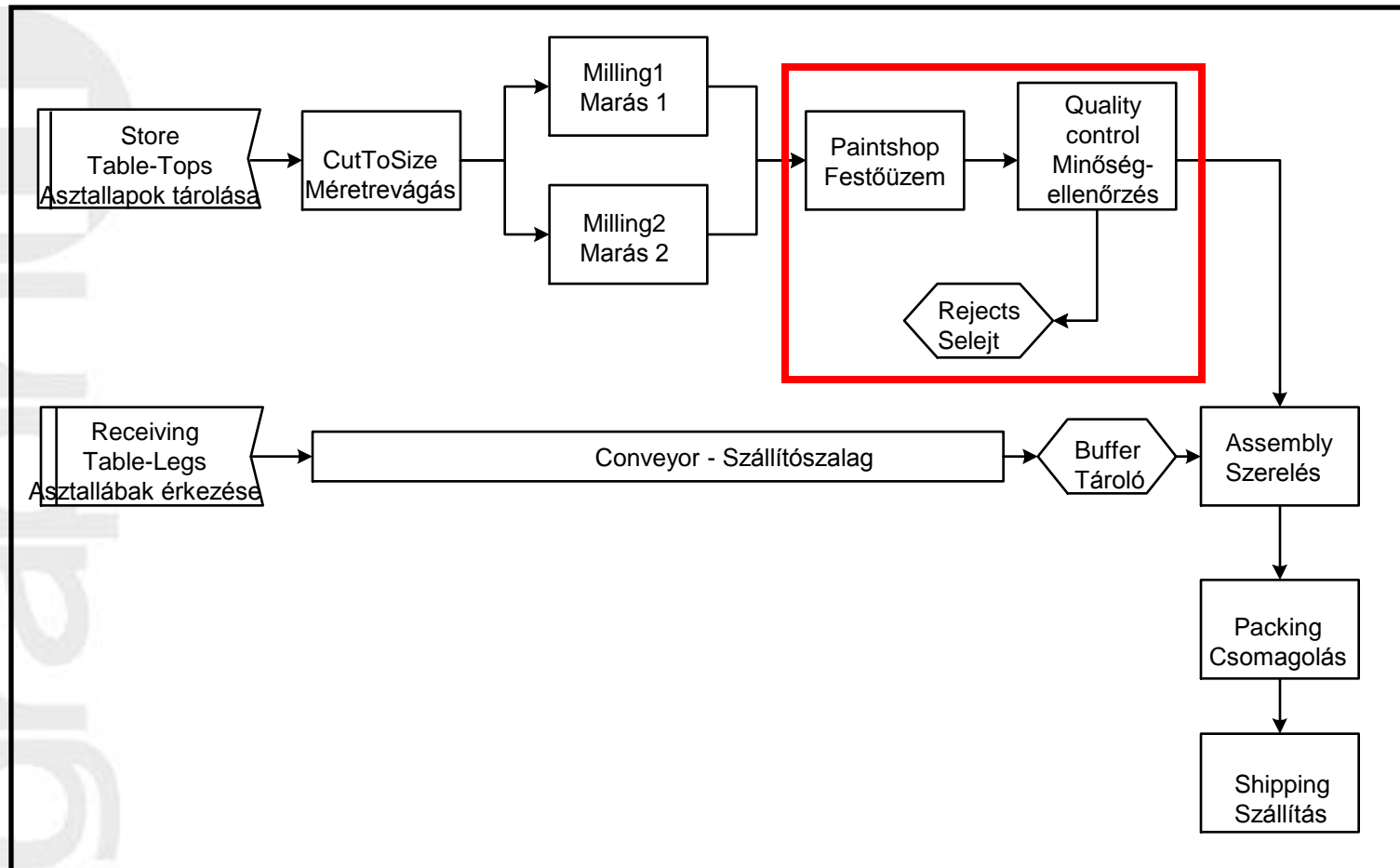


```
is
do
  inspect @.quality
  when "good" then
    @.move(station);
  when "minor_defects" then
    @.move(rework);
  else
    @.move(scrap);
  end;
end;
```



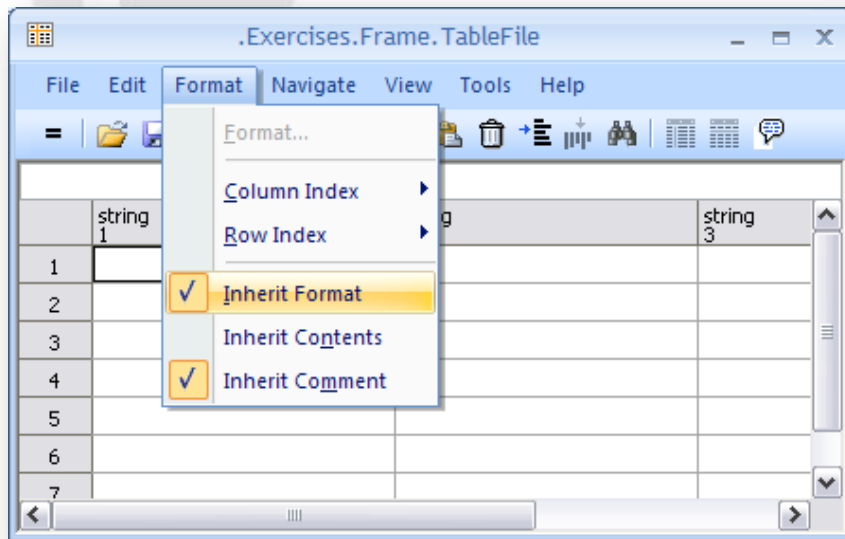
9. fejezet

Táblázatok és globális változók



A *TableFile* (táblázat) egy több oszlopos lista.

Az öröklődést szabályozó opciók a **Format** (Formátum) menüben:



Inherit Format (Formátum öröklése) vagy

Inherit Contents (Tartalom öröklése) parancsokkal kapcsolhatók ki.

Az **Inherit Contents** opció kikapcsolása után módosítható a tábla tartalma.

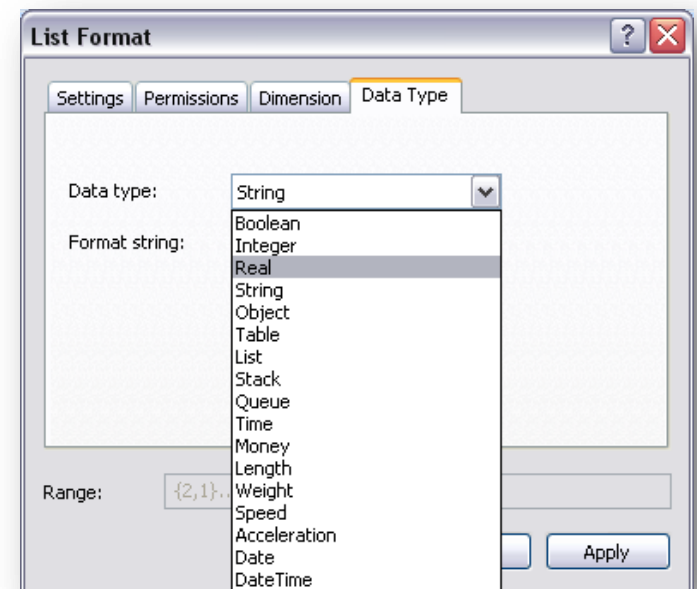
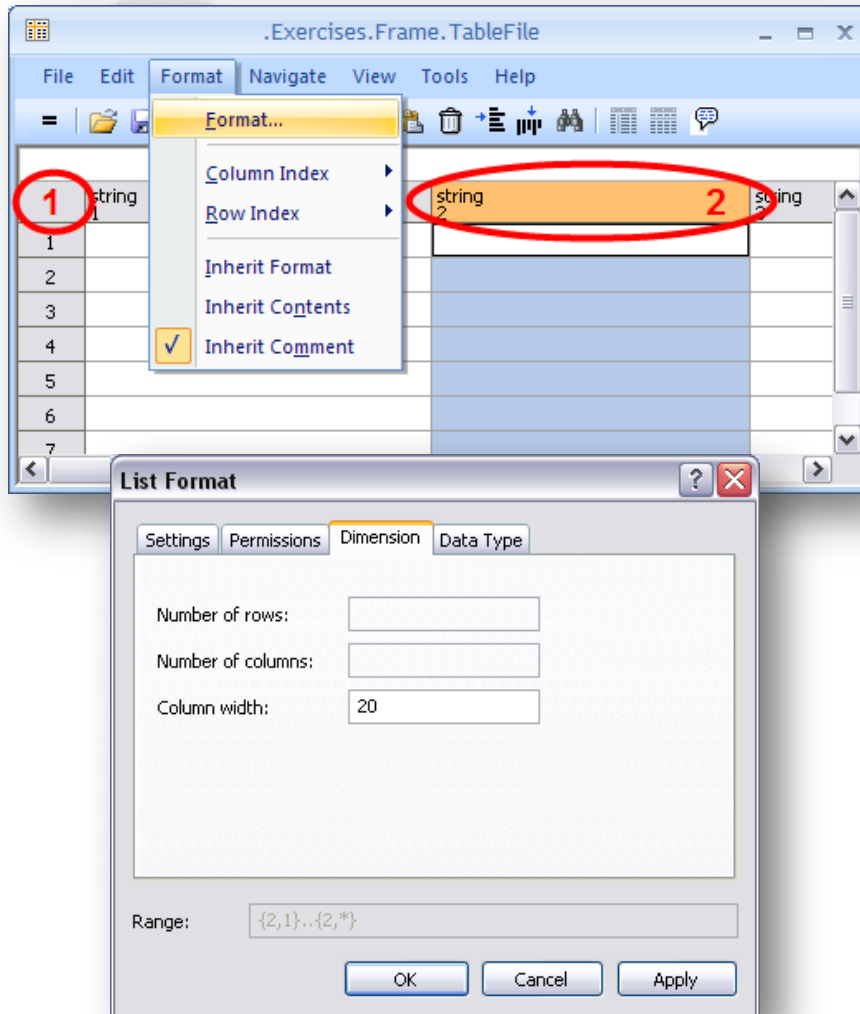
Az **Inherit Format** opció kikapcsolása után módosítható a tábla formátuma. Ez az opció kikapcsolja a **Inherit Contents** opciót is.

A **View > Comment** (nézet > megjegyzés) menüpontot kiválasztva a táblázattal kapcsolatos tudnivalók dokumentálhatók.

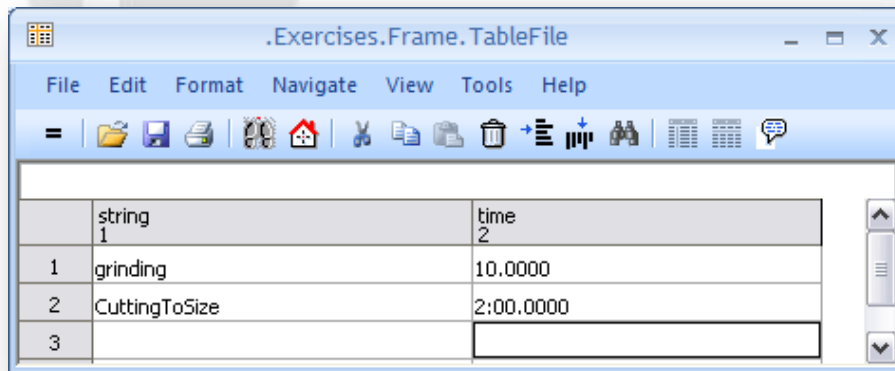
Az adattípus kiválasztása előtt válassza ki a teljes táblázatot (1) vagy a kívánt oszlopot (2). Válassza ki a

Format > Format > Data Type menüpontot és válassza ki az adattípust.

Adja meg a táblázat **Dimension** (méret) értékét.



A táblázatok esetében a Plant Simulation először az oszlopot, aztán a sort hívja meg:



	string 1	time 2
1	grinding	10.0000
2	CuttingToSize	2:00.0000
3		

Olvasás táblázatból:

```
print tablefile[1,2];
```

Írás a táblázatba:

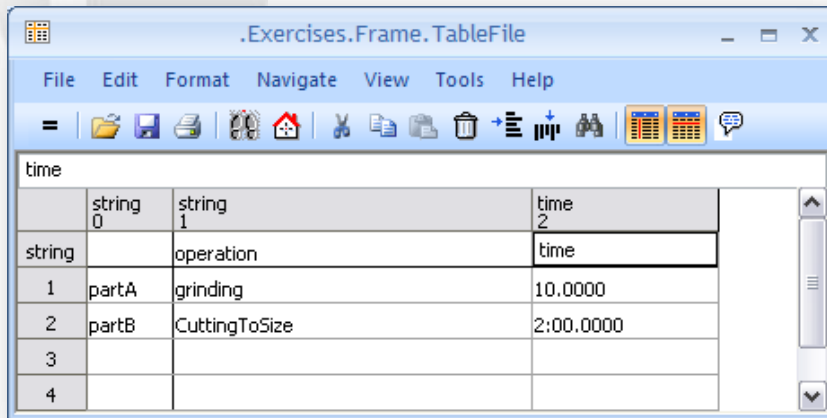
```
tablefile[1,3] := "grinding";
```

```
tablefile[2,3] := 180;
```

oszlop sor

Így lehet táblázat cellájának értékét attribútumhoz rendelni:

```
<object>.proctime := tablefile[2,1];
```



The screenshot shows a window titled ".Exercises.Frame.TableFile" with a menu bar (File, Edit, Format, Navigate, View, Tools, Help) and a toolbar. Below the toolbar is a table with the following data:

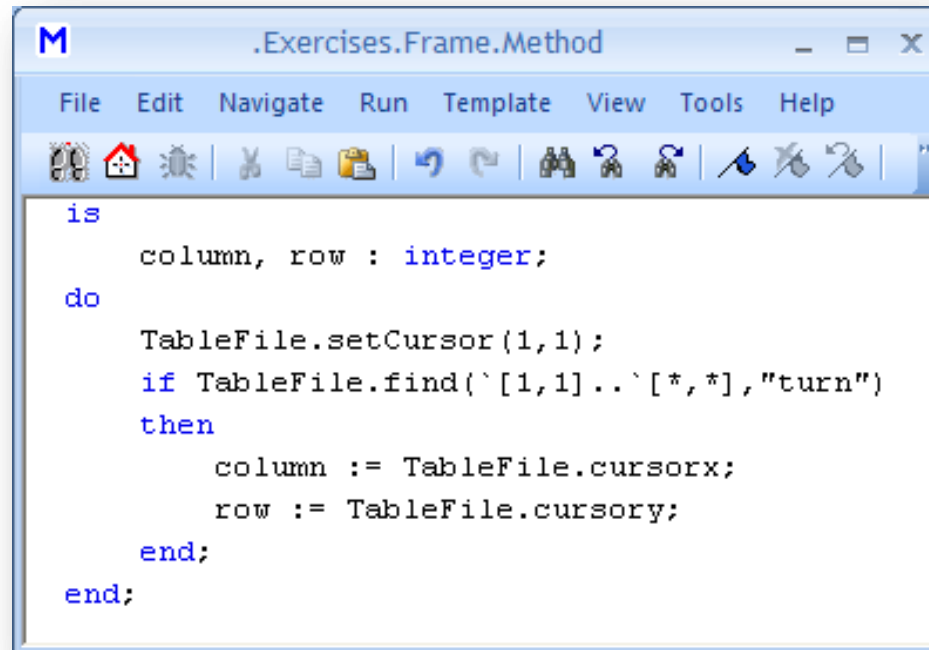
	string 0	string 1	time 2
string		operation	time
1	partA	grinding	10.0000
2	partB	CuttingToSize	2:00.0000
3			
4			

Egy cella meghívható az oszlop és sorindexével is.

A példában hozzárendeljük az anyagáram objektum feldolgozási idejéhez annak a cellának a tartalmát, amely a 2. oszlop, 1. sorában van a táblázatnak:

```
<object>.proctime := tablefile["time","partA"];
```

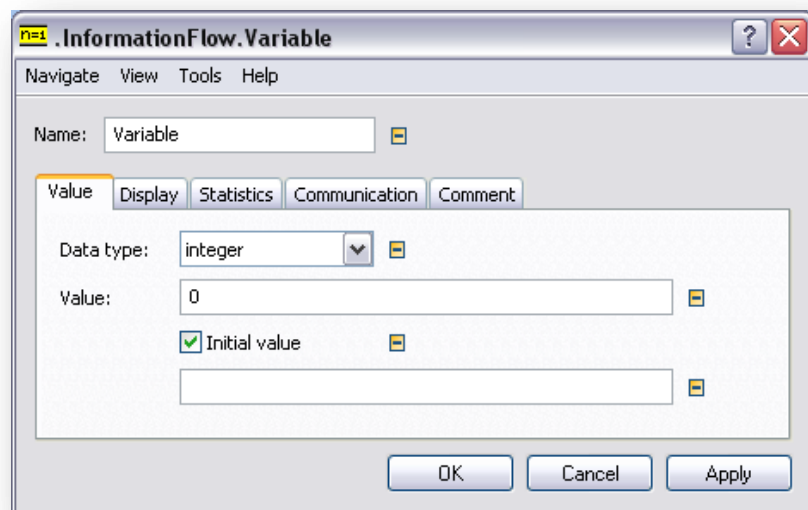
<code><TableFile>.setCursor(1,1) ;</code>	a táblázat kurzorát a megadott cellához állítja
<code><...>.find(`[1,1]..`[*,*],<value>) ;</code>	érték keresése az adott tartományban
<code><...>.CursorX, <...>.CursorY;</code>	kurzor adott oszlophoz (CursorX) vagy sorhoz (CursorY) állítása
<code><...>.xDim; <...>.yDim;</code>	visszatér az utolsó oszlop/sor számával
<code><...>.sort(3,"ascending") ;</code>	táblázat sorba rendezése egy vagy több oszlop alapján növekvő vagy csökkenő sorrendben
<code><...>.meanValue(`[1,*]) ;</code>	az összes számtani értékkel tér vissza (real és integer) az adott tartományban
<code><...>.delete(`[1,1]..`[*,*]) ;</code>	tartomány vagy a teljes tartalom törlése
<code><local_variable>.create;</code>	táblázat készítése lokális változóként, tartalom nélkül




```
is
    column, row : integer;
do
    TableFile.setCursor(1,1);
    if TableFile.find(`[1,1]..`[*,*],"turn")
    then
        column := TableFile.cursorx;
        row := TableFile.cursory;
    end;
end;
```

A Plant Simulation a táblázatot a kurzor pozíciójában kezdi el keresni: legyen a kurzor az első cellában.

A Plant Simulation visszatérése true (igaz), ha megtalálta az értéket, és a kurzort abba a cellába helyezi. A visszatérés false (hamis), ha nem találta meg az értéket.



Jellemzők:

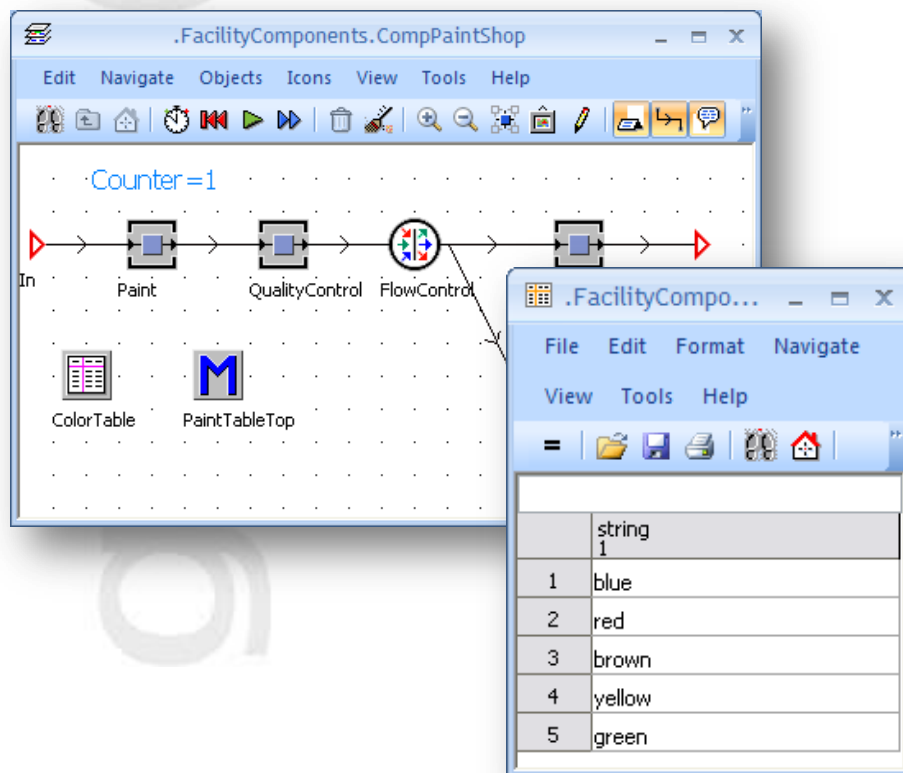
- Ikon: 
- Kapacitás: 0
- Információs áram objektum

A *Variable* (változó) objektum értékek tárolására szolgál a szimuláció hosszabb idejére. Ugyanúgy meghívható, mint bármely más objektum. Az értékek lehetséges tartománya az adattípustól függ.

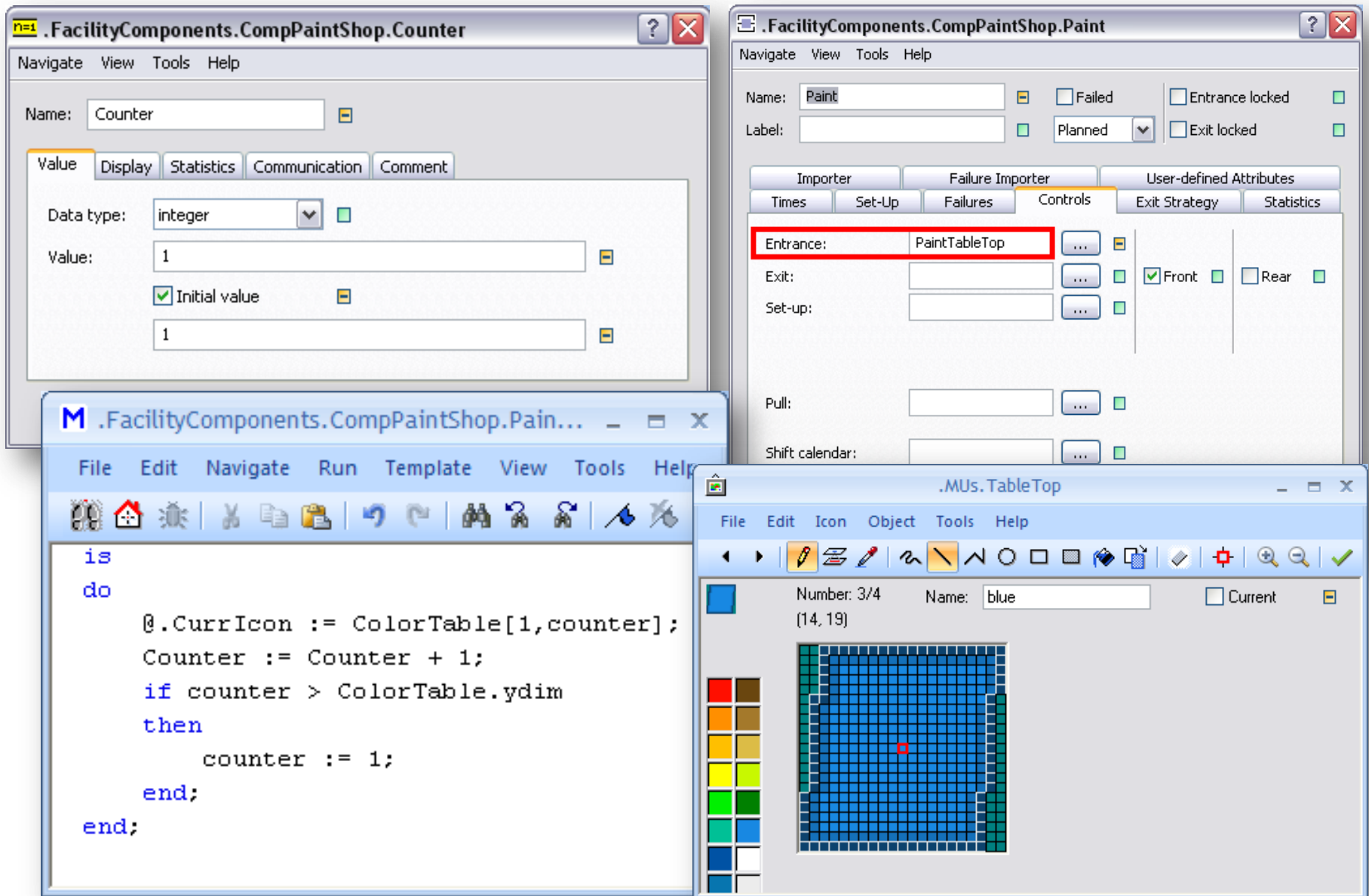
A *Variable* meghívható más objektumok vagy metódusok által.

A szimuláció inicializálása során alapértékre lehet állítani a változót, az **Initial value** opcióval, ekkor meg kell adni a kezdőértékét a változónak.

1. Nyissa meg a **CompPaintshop** *Frame* - et a *FacilityComponents* mappában. Adjon hozzá egy *TableFile* objektumot, és legyen a neve *ColorTable*. Adjon hozzá egy *PaintTableTop* nevű metódust.
2. Adja meg az asztalszíneket a *ColorTable* objektumba. A színek megegyeznek az ikonszerkesztőnél látható állapotikon színekkel.



3. Melyik kontrollba kell megadni a metódus meghívását?
4. Programozza le a metódust, amelyikben a Plant Simulation lefesti az asztallapokat a táblázatban lévő színekkel. Szedje szét a problémát kisebb lépésekre.
5. Ha szükséges illessze be a még hiányzó objektumokat.



FacilityComponents.CompPaintShop.Counter

Name: Counter

Data type: integer

Value: 1

☒ Initial value

1

FacilityComponents.CompPaintShop.Paint

Name: Paint

Label:

☐ Failed ☐ Entrance locked ☐ Exit locked

☒ Planned

Importer Failure Importer User-defined Attributes

Times Set-Up Failures Controls Exit Strategy Statistics

Entrance: PaintTableTop

Exit:

Set-up:

Pull:

Shift calendar:

☒ Front ☐ Rear

.MUs.TableTop

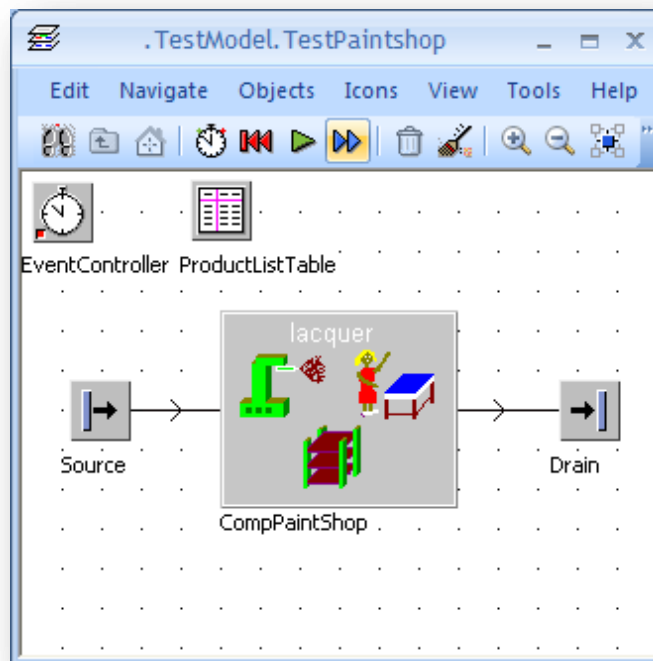
File Edit Icon Object Tools Help

Number: 3/4 (14, 19) Name: blue ☐ Current

```

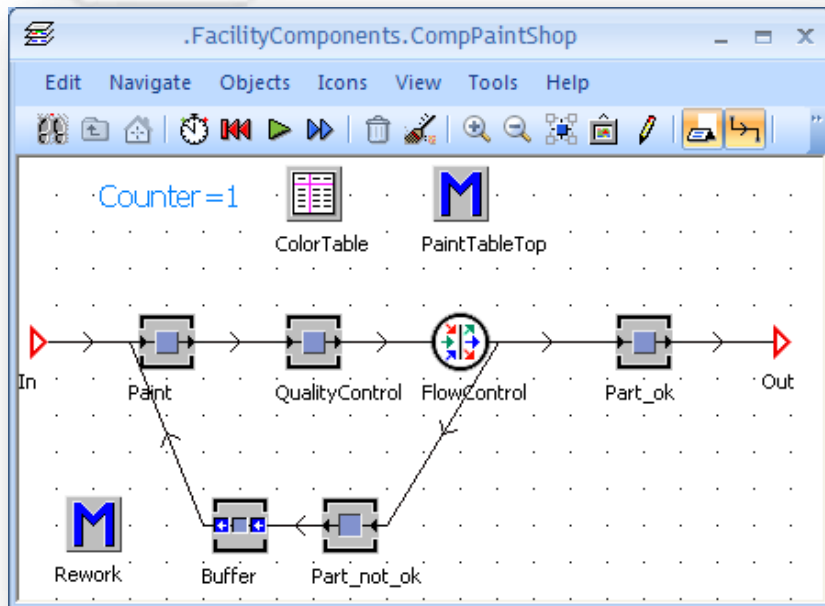
is
do
    @.CurrIcon := ColorTable[1,counter];
    Counter := Counter + 1;
    if counter > ColorTable.ydim
    then
        counter := 1;
    end;
end;
    
```

1. Nyissa meg a **TestPaintshop** *Frame*-et a **TestModel** mappában. Amint ki lett bővítve a *Paintshop Frame* a *Paintshop Frame* osztályban (a **FacilityComponents** mappában), a teszt *TestPaintshop Frame* mutatja a változásokat.
2. Indítsa el a szimulációt, és figyelje meg, hogyan festődnek az asztallapok a *Paintshop Frame*-ben.



Most bezárjuk a festési ciklust, és minden hibásan festett asztallapot visszaküldünk a *Paintshop* állomásra:

1. Nyissa meg a **CompPaintshop** *Frame* osztályt.



2. Törölje a *Drain* objektumot, és illesszen be egy *Buffer* objektumot 4 alkatrész kapacitással.

3. Zárja be a kört a *Paintshop* állomással egy *Connector* objektum beillesztésével.

4. Illesszen be egy metódust, aminek a programja az asztallap objektum *Quality* attribútumát GOOD - ra állítja ha az a *Part_not_ok* állomásra érkezik. Emellett az MU nevét is megváltoztatja *tableTop_good* - ra.

```
is
do
    @.Quality := "good";
    @.name := "good";
end;
```

.FacilityComponents.CompPaintShop.Part_not_ok

Navigate View Tools Help

Name: Part_not_ok ☐ Failed ☐ Entrance locked ☒

Label: ☒ Planned ☐ Exit locked ☒

Importer Failure Importer User-defined Attributes

Times Set-Up Failures Controls Exit Strategy Statistics

Entrance: Rework ☐

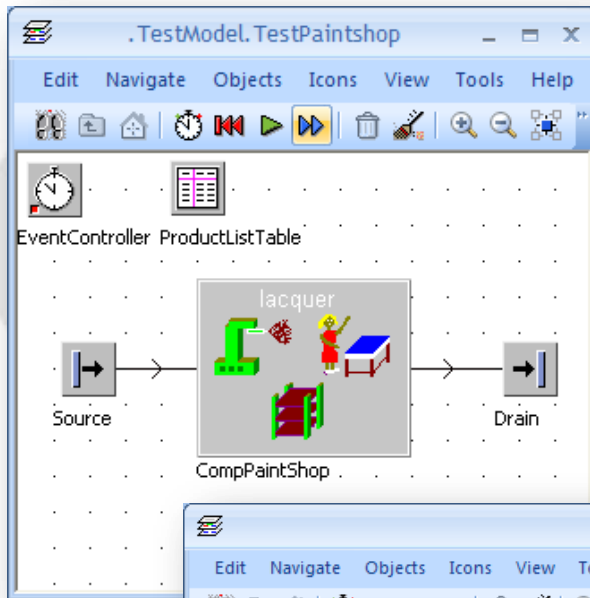
Exit: ☒ ☒ Front ☐ Rear ☒

Set-up: ☒

Pull: ☒

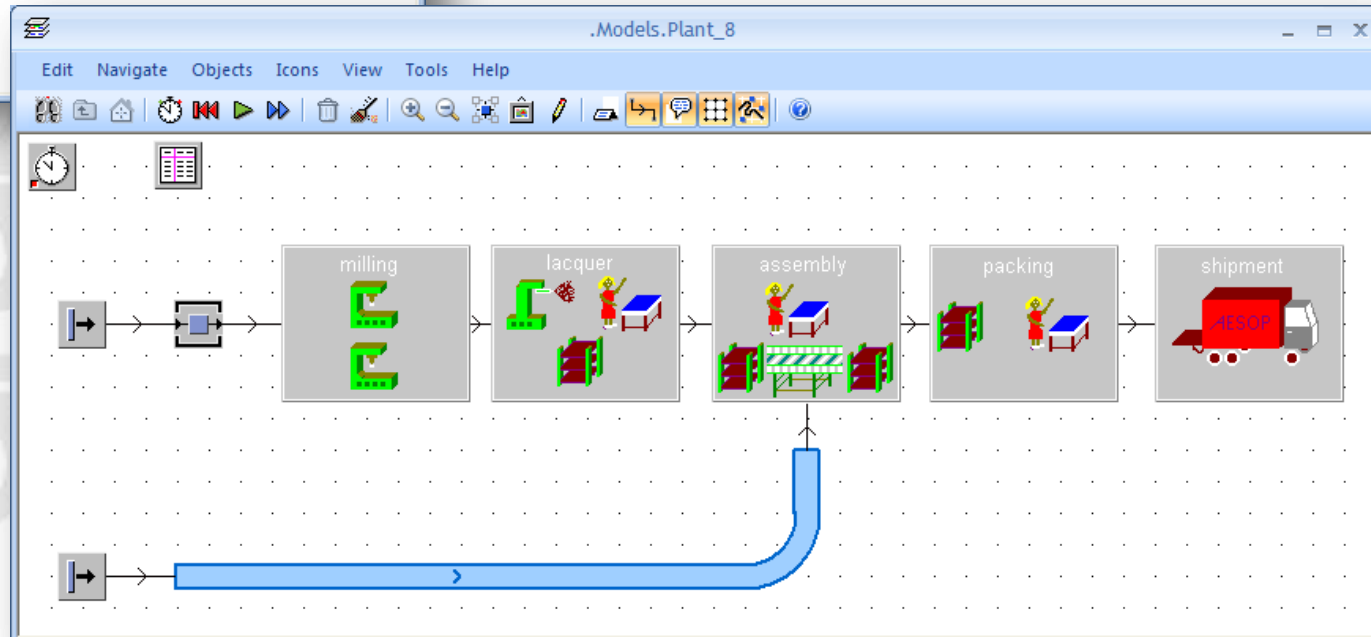
Shift calendar: ☒

OK Cancel Apply



Győződjön meg arról, hogy a **TestPaintshop** *Frame* megfelelően működik.

Aztán győződjön meg arról, hogy a **Paintshop** *Frame* a megfelelő módon viselkedik a **Plant_8** *Frame* - en.





A *StackFile*, *QueueFile* és *CardFile* objektumok **egy oszlopos listák**. Nem lehet **üres cellájuk**. Különböző jellemzőkkel rendelkeznek a cellák adatainak olvasására, beszúrására, stb.

StackFile (hasonló a verem adathoz)

A hozzáférés LIFO módszerrel történik (Last In First Out). Az új elemek a tetejére kerülnek, és az elsőként hozzáadott cellát távolítja el.

QueueFile (hasonló a sorhoz)

A hozzáférés FIFO módszerrel történik (First In First Out). Az elemeket abban a sorrendben menti el, ahogy a beszúrásuk történik. Az az elem kerül eltávolításra, amely a legrégebben vár a sorban.

CardFile (hasonló a listához)

Véletlenszerű hozzáféréssel a cellapozíciót használja. Eltávolításkor a követő cellák feljebb mozognak.

<code><...>.push(<value>)</code>	érték beszúrása a <i>StackFile</i> vagy <i>QueueFile</i> legfelső cellájába
<code><...>.insert(3,<value>)</code>	érték beszúrása a harmadik sorban lévő cellába a <i>CardFile</i> objektumnál, a többi cellát lefelé mozgatva eggyel
<code><...>.[1]</code>	a <i>StackFile</i> utolsó cellájának tartalmát távolítja el. A többi cella eggyel feljebb mozog. Eltávolítja az első celláját a <i>QueueFile</i> objektumnak
<code><...>.[3]</code>	a <i>CardFile</i> harmadik sorában lévő cella tartalmát olvassa ki és távolítja el a tartalmát, a további cellák egy pozíciót felfelé mozognak
<code><...>.pop</code>	a <i>StackFile</i> utolsó cellájának tartalmát olvassa ki. <i>QueueFile</i> első cellájának tartalmát olvassa ki. A kiolvasott cellákat eltávolítja
<code><...>.read(3)</code>	a <i>CardFile</i> harmadik sorának tartalmát olvassa ki a cella eltávolítása nélkül
<code><...>.dim</code>	visszatér az utolsó sor sorszámával



10. fejezet

Műveletek Adattípusok konverziója

A logikai műveletek igaz vagy hamis (true, false) értékkel térnek vissza.

Használhatók az **AND**, **OR** és **NOT** műveletek.

NOT	NOT TRUE → FALSE NOT FALSE → TRUE
=	FALSE = FALSE → TRUE TRUE = TRUE → TRUE
/=	FALSE /= TRUE → TRUE

AND	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

OR	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

Visszatérési érték

+ összeadás

- kivonás

* szorzás

// egész osztás

\\ modulusz

/ osztás

= egyenlő

/= nem egyenlő

> nagyobb, mint

< kevesebb, mint

>= nagyobb vagy egyenlő

<= kisebb vagy egyenlő

} integer

} integer $17//5 = 3$

$17\\5 = 2$

→ integer, real


} boolean

Visszatérési érték



+	összeadás	}	real
-	kivonás		
*	szorzás		
/	osztás		
=	egyenlő	}	boolean
/=	nem egyenlő		
>	nagyobb, mint		
<	kevesebb, mint		
>=	nagyobb vagy egyenlő		
<=	kisebb vagy egyenlő		
==	nagyjából egyenlő		
<= =	kevesebb vagy nagyjából egyenlő	}	
>= =	nagyobb vagy nagyjából egyenlő		

Visszatérési érték



+	összeadás	→	string
=	egyenlő	{	boolean
/=	nem egyenlő		
==	nagyjából egyenlő		
	toLowerCase	{	string
	toUpperCase		
	copy		
	incl	{	integer
	omit		
	strlen		
	pos		

Metódus	Visszatérés
to_str(<...>)	string
num_to_bool(<integer>)	boolean
bool_to_num(<boolean>)	real
str_to_num(<string>)	real
str_to_bool(<string>)	boolean
str_to_time(<string>)	time
str_to_date(<string>)	date
str_to_datetime(<string>)	datetime
str_to_length(<string>)	length
str_to_weight(<string>)	weight
str_to_speed(<string>)	speed
str_to_obj(<string>)	object

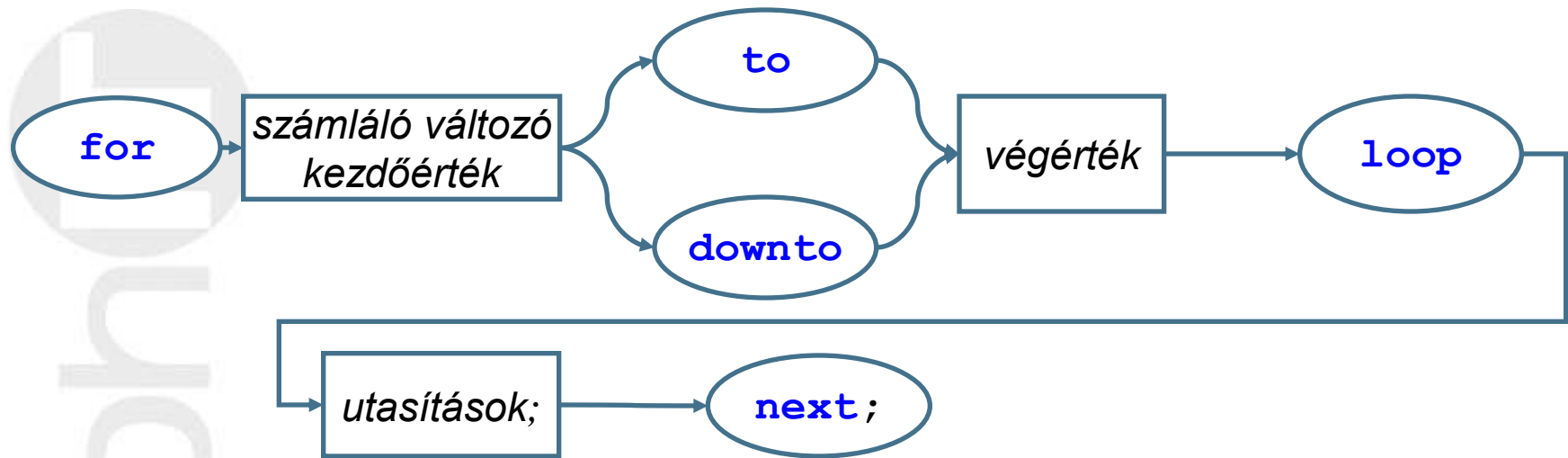
A különböző adattípusokkal a megfelelő műveleteket lehet végrehajtani. Nem minden műveletet lehet minden adattípussal végrehajtani.

Mielőtt a különböző adattípusokon műveleteket végezne, egyforma típusúvá kell konvertálni azokat. Csak ez után tudja a Plant Simulation végrehajtani a műveleteket rajtuk.

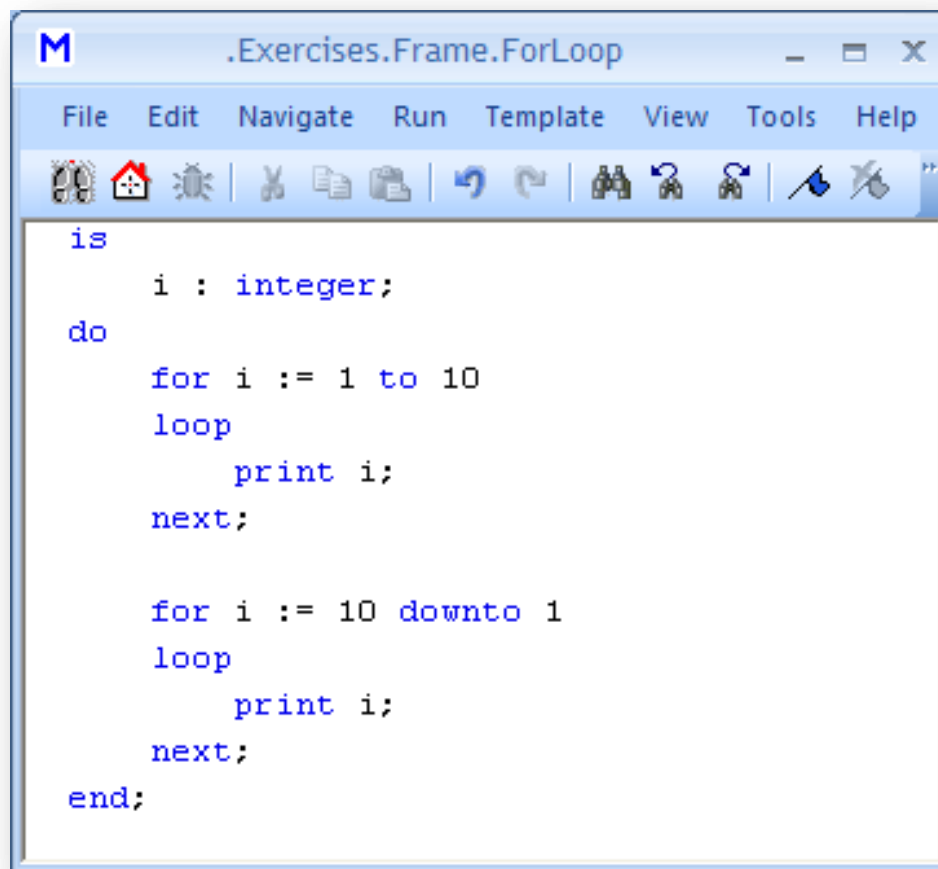


11. fejezet

Ciklusok



A **for** ciklus egy tartományon halad végig a kezdőérték és a végérték között. A számláló változónak integer adattípusúnak kell lennie és megadja a ciklus kezdetét. Amikor a ciklus egy lépése lefut, a Plant Simulation hozzáad 1-et a számláló változóhoz (a **to** beállítással) vagy levon 1-et (a **downto** beállítással). Addig fut a ciklus, amíg a számláló változó értéke el nem éri a végértéket.

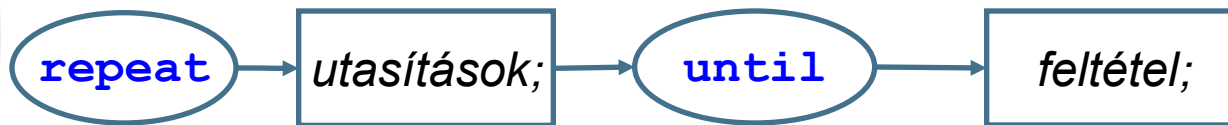


```
is
  i : integer;
do
  for i := 1 to 10
  loop
    print i;
  next;

  for i := 10 downto 1
  loop
    print i;
  next;
end;
```

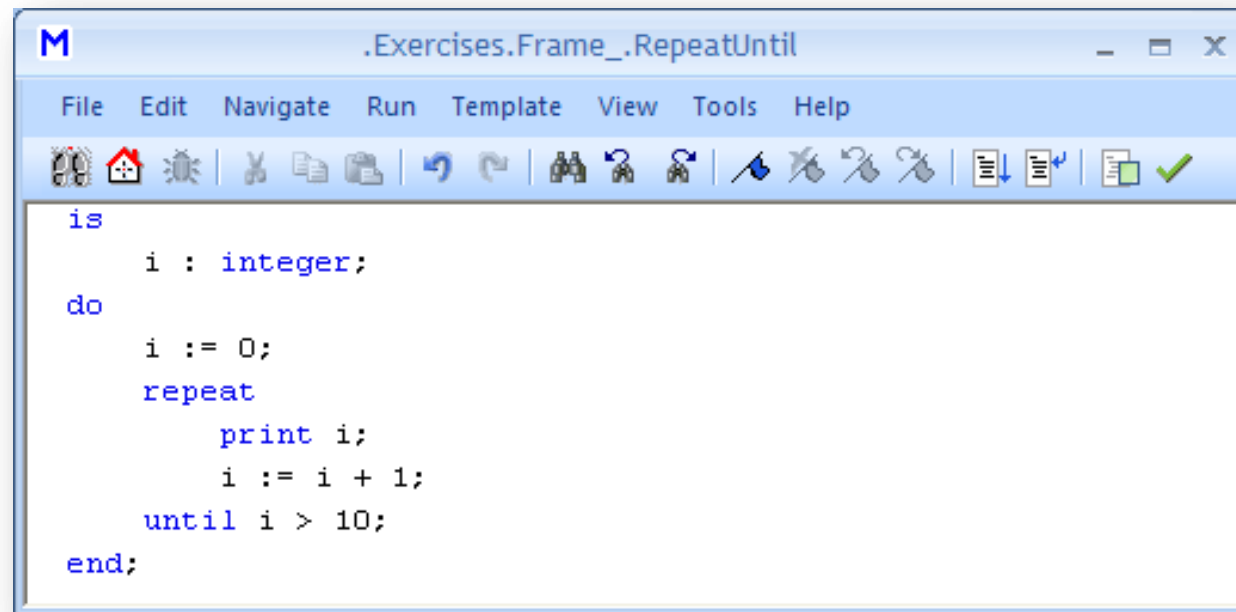
i: a számláló változó

1: lépték



A Plant Simulation a **repeat** ciklust legalább egyszer lefuttatja. A Plant Simulation akkor lép ki a ciklusból, amikor a feltétel értéke true (igaz).

Ha a feltétel soha nem válik igazzá, akkor a Plant Simulation végtelenül futtatja azt. Egy végtelen ciklus a **Shift+Alt+Ctrl** billentyűvel szakítható meg. Ekkor megjelenik a hibakereső.

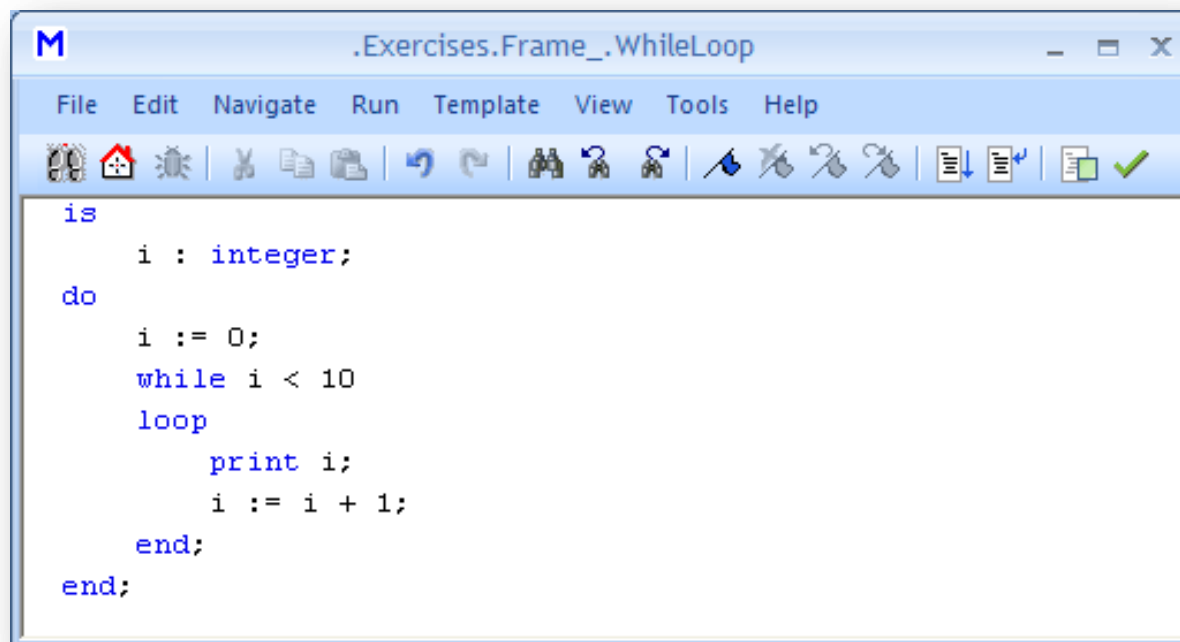


```
is
  i : integer;
do
  i := 0;
  repeat
    print i;
    i := i + 1;
  until i > 10;
end;
```



A Plant Simulation addig futtatja a **while** ciklust, amíg a feltétel értéke true (igaz).

Amikor az utasítások teljesen feldolgozásra kerülnek, a Plant Simulation még egyszer leellenőrzi, hogy a feltétel igaz.



The screenshot shows a GraphIT IDE window titled ".Exercises.Frame_.WhileLoop". The window has a menu bar with "File", "Edit", "Navigate", "Run", "Template", "View", "Tools", and "Help". Below the menu bar is a toolbar with various icons for file operations, navigation, and execution. The main editor area contains the following code:

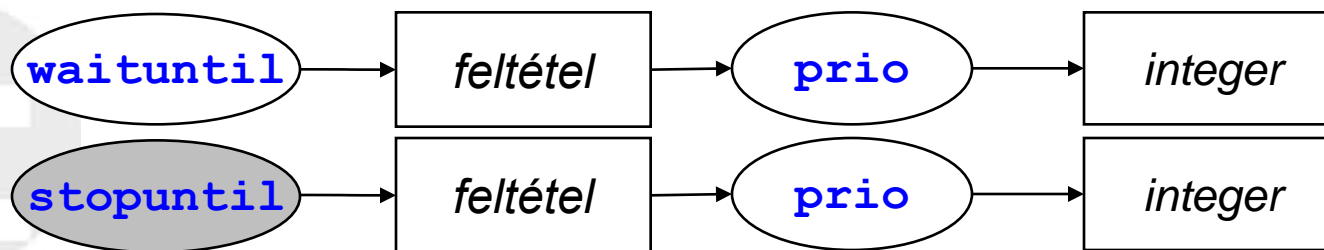
```
is
  i : integer;
do
  i := 0;
  while i < 10
  loop
    print i;
    i := i + 1;
  end;
end;
```

1. Készítsen egy **Loops** nevű *Frame*-et az **Exercises** mappában.
2. Adjon hozzá egy metódust és programozzon egy olyan ciklust, amely 1-től 10-ig halad végig értékeken.
3. Írassa ki a számláló változó értékét a konzolon.
4. Lépkedjen végig a metóduson. A *Watch Window* ablakban is meg lehet jeleníteni a számláló változót.



12. fejezet

Feltételes várakozások - waituntil,
stopuntil



A **waituntil** utasítás egy metódus futásának ideiglenes felfüggesztésére szolgál – egy feltételnek megfelelően.

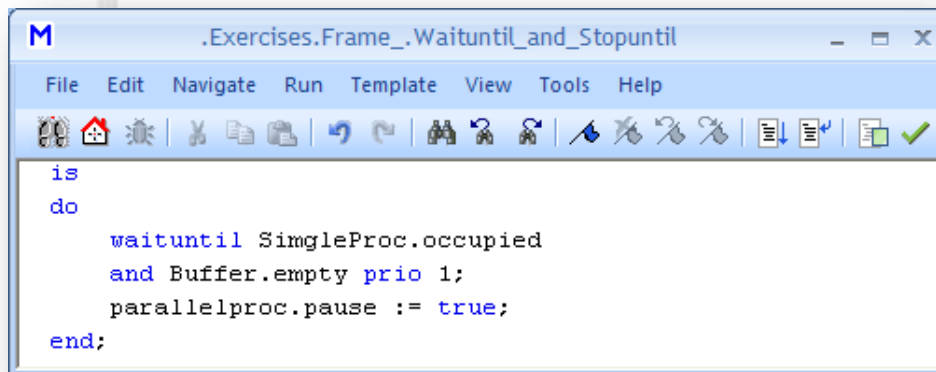
Ha a feltétel nem igaz, akkor a Plant Simulation megszakítja a metódus futását ott. Ideiglenesen felfüggeszti a metódus futását, de minden jellemzőjét (változók, stb.) elmenti. Ez után figyeli a feltételt, miközben a szimuláció futása folytatódik.

Amint a feltétel igazzá válik, megszakítja a futását a metódusoknak és ezt a metódust folytatja, ott ahol abbahagyta.

A feltételnek ehhez boolean típusúnak és true értékűnek kell lennie.

Amikor több metódus vár ugyanannak a feltételnek a teljesülésére, a Plant Simulation egyszerre aktivizálja újra azokat, és abban a sorrendben futtatja le őket, ami a **prio** kifejezésben megadott.

A **waituntil** utasítással ideiglenesen felfüggesztett metódusok esetében a fordító újra kiértékeli a feltételt, mielőtt a különálló metódusokat lefuttatná.



A **stopuntil** utasítással ideiglenesen felfüggesztett metódusok esetében a Plant Simulation nem értékeli ki újra a feltételt az egyes metódusok futtatása előtt. Mindet egyszerre aktivizálja újra és a megadott prioritás szerinti sorrendben futtatja le őket.

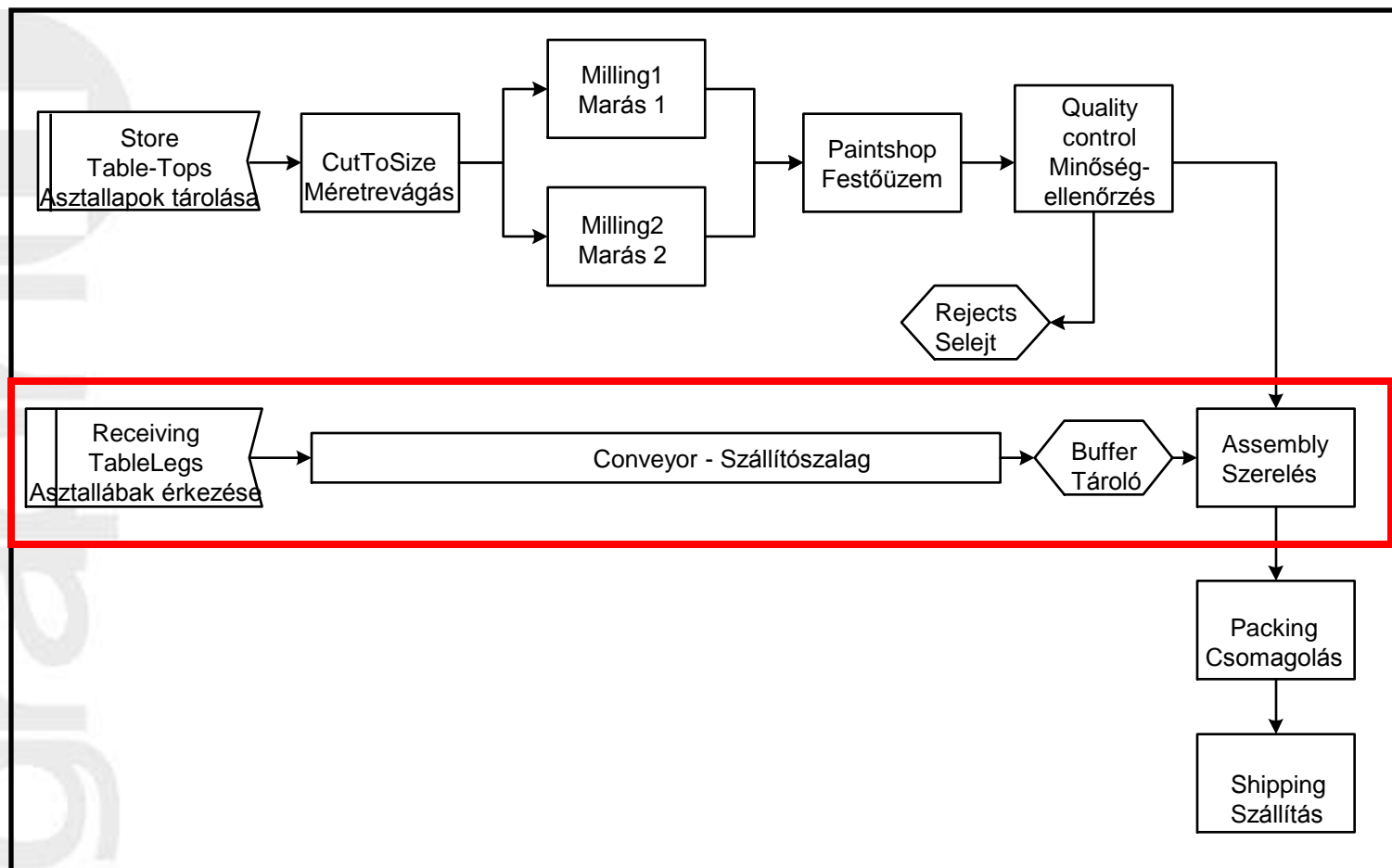
Az egy időben ideiglenesen felfüggesztett metódusok maximális száma (**Maximum number of suspended methods**) az **Options > Simulation** menü alatt adható meg



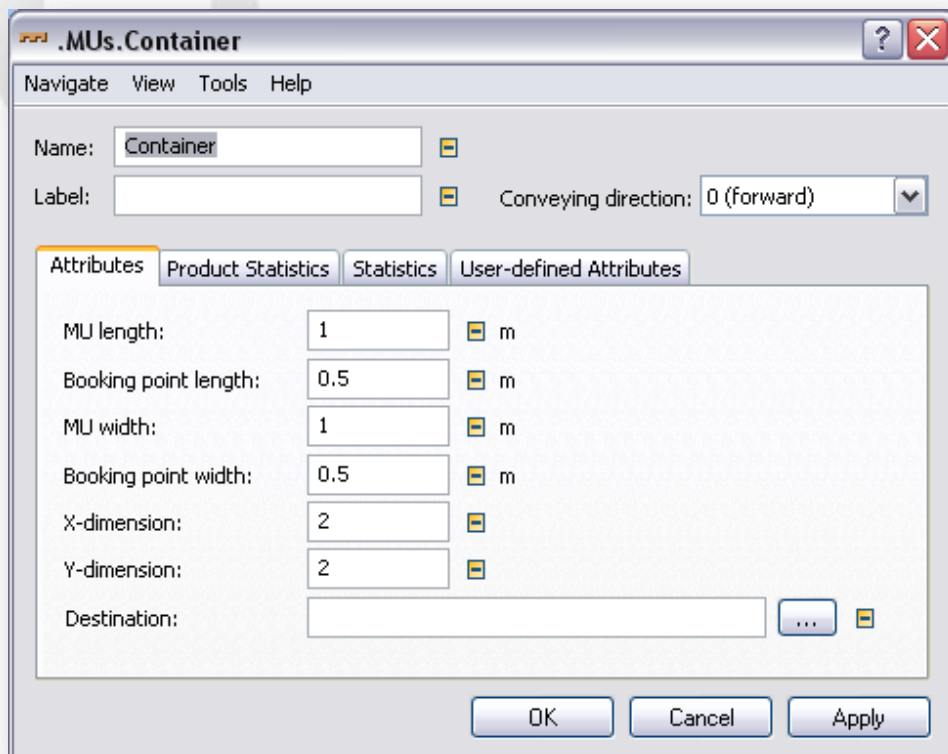
13. fejezet

Egy objektum tartalma

Fel- és lerakodás



A *Container* objektum mátrix-alapú tárolóhelyekkel rendelkezik.



The screenshot shows a software dialog box titled ".MU.Container". It has a menu bar with "Navigate", "View", "Tools", and "Help". Below the menu bar, there are input fields for "Name:" (containing "Container") and "Label:", followed by a "Conveying direction:" dropdown menu set to "0 (forward)". Below these are four tabs: "Attributes" (selected), "Product Statistics", "Statistics", and "User-defined Attributes". The "Attributes" tab contains several labeled input fields: "MU length:" (value 1), "Booking point length:" (value 0.5), "MU width:" (value 1), "Booking point width:" (value 0.5), "X-dimension:" (value 2), "Y-dimension:" (value 2), and "Destination:" (empty). Each of the first six fields has a small icon to its right. At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

Rá lehet **rakodni és szállítani** bármilyen MU-t. Ezzel lehet modellezni pl: palettákat, stb.

Meg lehet adni az x és y irányú **kapacitását**. A *Container* nem vizsgálja a rá töltött objektum hosszát.

A **cont** metódus használható az anyagáram objektumok és a *Container* tartalmának lekérdezésére. Formátuma: **<object>.cont**.

```
@.move (MillingMachine) *  
saw.cont.move (MillingMachine)
```

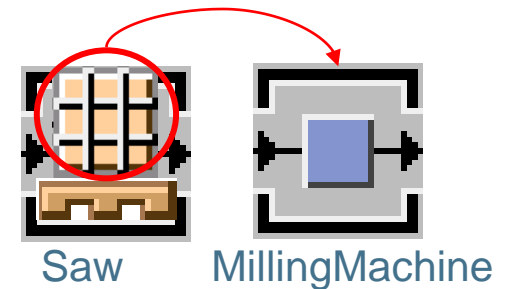
* @, amikor az aktuális MU hívja meg a metódust



A **cont** metódus rendszerint a következő MU - val tér vissza, amelyik kész az objektumból való kilépésre, akkor is, ha másik MU - n található is (pl. *Container*).

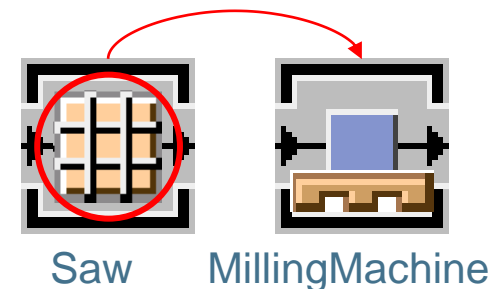
A felrakodott *Container* tartalmához először az állomás tartalmát, majd a rajta lévő *Container* tartalmát lekérdezve lehet.

```
@.cont.move (MillingMachine)  
Saw.cont.cont.move (MillingMachine)
```



A **cont** metódus használható egy másik MU *Container* - re mozgatásához:

```
@.move (MillingMachine.cont)  
Saw.cont.move (MillingMachine.cont)
```



Ez akkor is igaz, ha egymáson lévő *Container* - ekről van szó. (... **.cont.cont.cont...**)

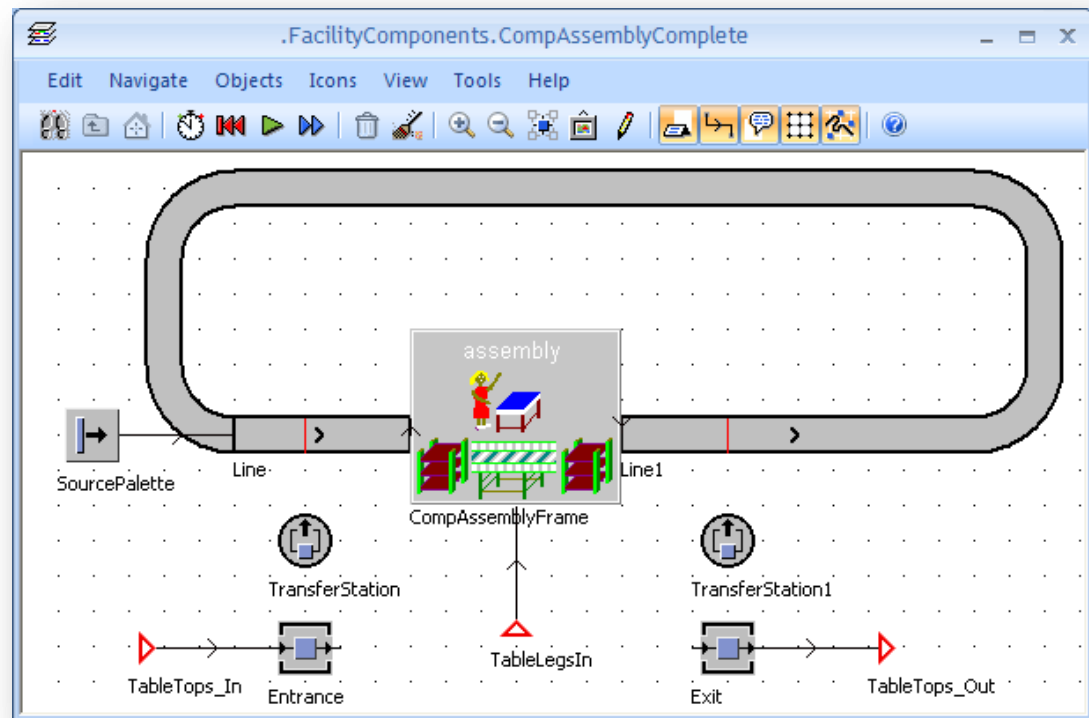
- Mielőtt egy objektum tartalmához fordulna, győződjön meg, hogy az objektumnak van tartalma. Használja valamelyik állapotjelző attribútumot az anyagáram objektum ellenőrzésére.
- Az anyagáram objektumnak különböző állapotai lehetnek, amelyeket le lehet kérdezni és be lehet állítani attribútumokkal. A megfelelő metódus visszatérése true (igaz), ha az objektum abban az állapotban van, és false (hamis), ha nem.

<code><object>.empty;</code>	visszatérése TRUE, ha nincs MU az állomáson, FALSE ha van MU rajta
<code><...>.full;</code>	visszatérése TRUE, ha minden állomása a <i>ParallelProc</i> - nak vagy a <i>Buffer</i> - nek foglalt, FALSE ha nem ez a helyzet
<code><...>.occupied;</code>	visszatérése TRUE, ha legalább egy MU van az állomáson, FALSE, ha nincs MU rajta
<code><...>[2,1].occupied;</code>	visszatérése TRUE, ha van MU a megadott állomáskoordinátákon
<code><...>.cont.finished;</code>	visszatérése TRUE, ha az MU feldolgozási ideje eltelt és az MU kész elhagyni az objektumot
<code><...>.ready;</code>	visszatérése TRUE ha az objektum megtelt, és van kilépésre kész MU. A kombinációja az occupied és cont.finished metódusoknak (Megjegyzés: nem figyelhető meg waituntil paranccsal)


Szerelési (assembly) ciklust kell készíteni. Az asztallapok felkerülnek a palettákra, amely elviszi azokat a szerelési állomásra. Itt az asztallábak hozzáadódnak az asztallapokhoz. Ezután az összeszerelt asztalok elhagyják a szerelési állomást, a paletták visszamennek a szerelési állomásra, újra használatra.

A végrehajtáshoz a következőket kell tenni:

1. *Container* felrakodás
2. *Container* lerakodás
3. Fel- és lerakodási ciklus
4. A szerelési folyamat



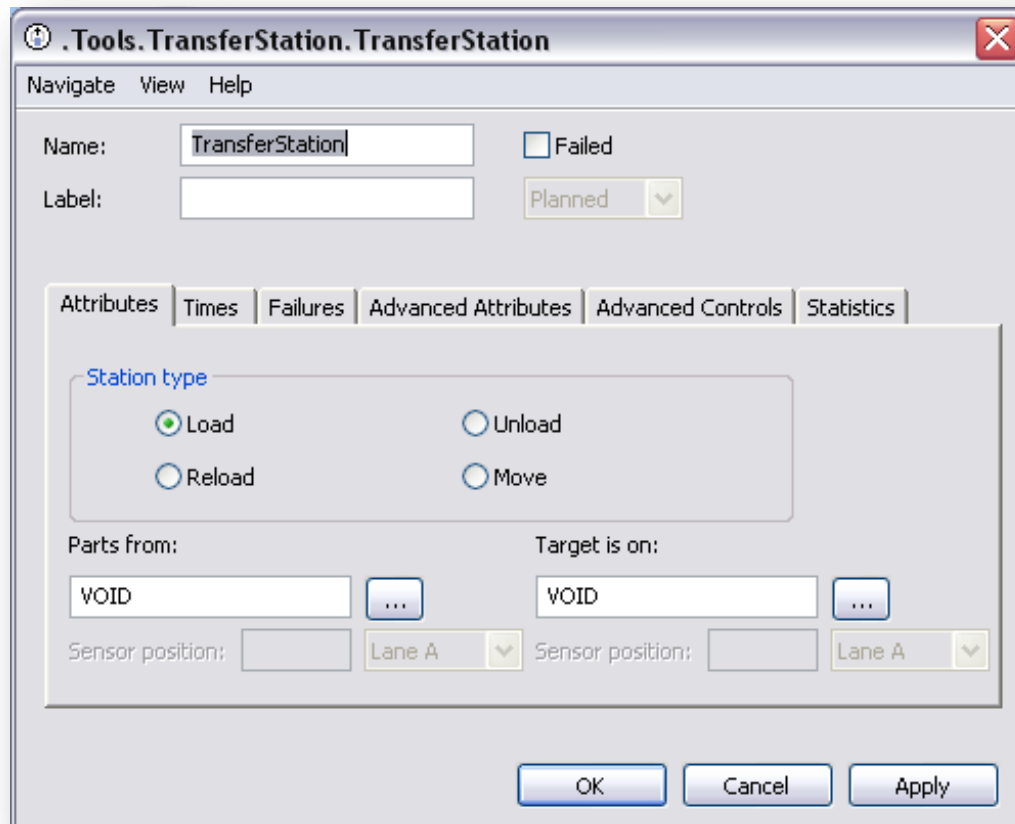
Jellemzők:

- Ikon: 
- Kapacitás: 1
- Aktív anyagáramlás objektum

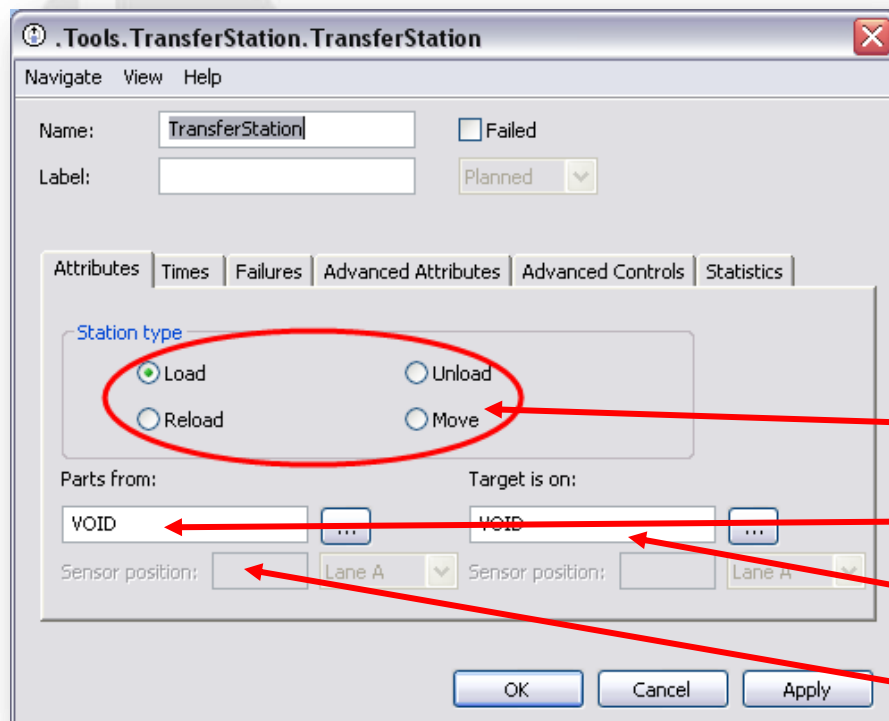
A TransferStation az MU-k egyik objektumról egy másikra történő mozgatását végzi.

A mozgatás típusa lehet:

- Load (felrakodás),
- Unload (lerakodás),
- Reload (feltöltés),
- Move (átmozgatás).



Sensor segítségével a TransferStation az MU-kat pl. egy SingleProc - ról egy Line objektumra rakodja.



Meg kell határozni:

- az állomás típusát,
- a forrást (Parts Station),
- a célállomást (Target Station), ill.
- hosszorientált objektum esetében a Sensor pozícióját.

Megadható a feldolgozási idő.

Tools.TransferStation.TransferStation

Navigation View Help

Name: TransferStation ☐ Failed

Label: Planned

Attributes Times Failures Advanced Attributes Advanced Controls Statistics

Processing time: 1:00 DDD:HH:MM:SS.XXXX

Recovery time: 0 DDD:HH:MM:SS.XXXX

Lekérhetők az állomás statisztikai adatai is.

Attributes Times Failures Advanced Attributes Advanced Controls Statistics

Working:	0.00 %	Blocked:	0.00 %
Waiting:	0 %	Failed:	0.00 %
Waiting for parts:	0.00 %	Paused:	0.00 %
Waiting for target:	0.00 %	Unplanned:	0.00 %
Entries:	0	<button>Statistics Table</button>	


Tools.TransferStation.TransferStation.StatusTable

waiting

	Value	Frequency	Duration	%_Frequency	%_Duration	Mean Duration	Standard
1	waiting	1	0.0000	100.00	0.00	0.0000	

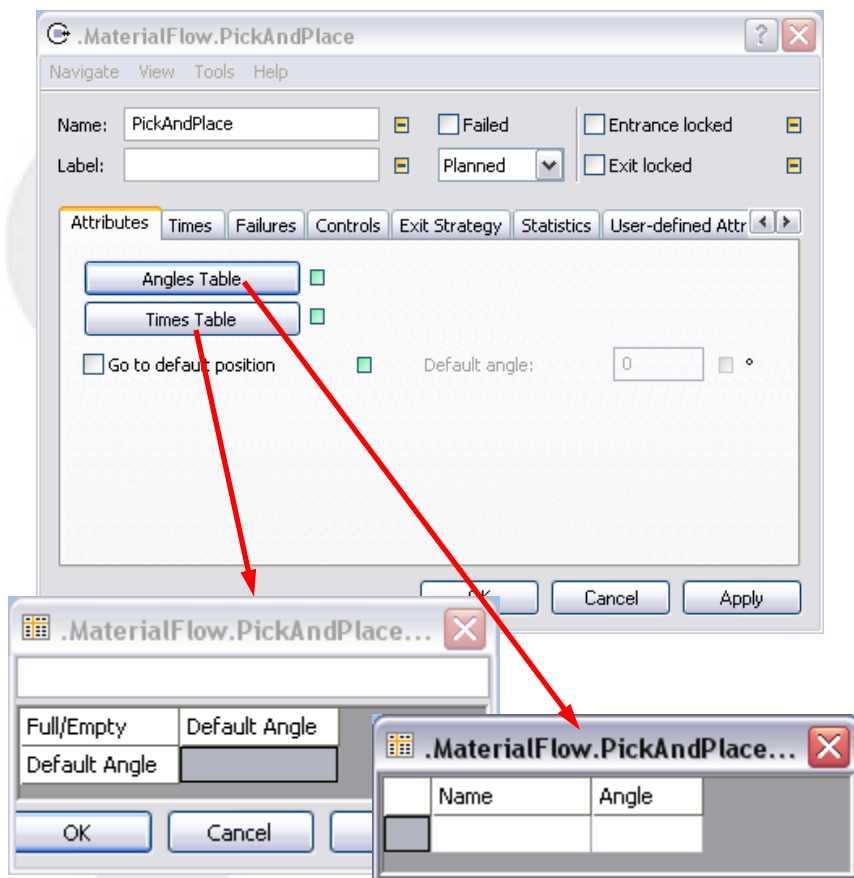
OK Cancel Apply

Jellemzők:

- Ikon: 
- Kapacitás: 1
- Aktív anyagáramlás objektum

A **PickAndPlace** nevű objektum a darabok egyik állomásról egy másikra történő elhelyezését végzi.

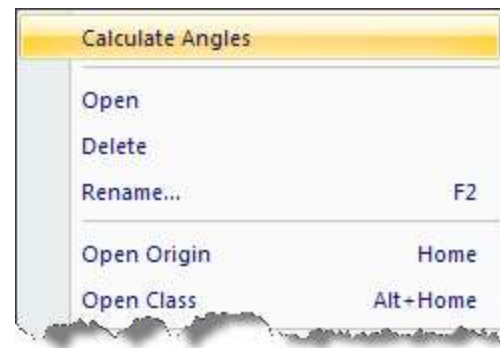
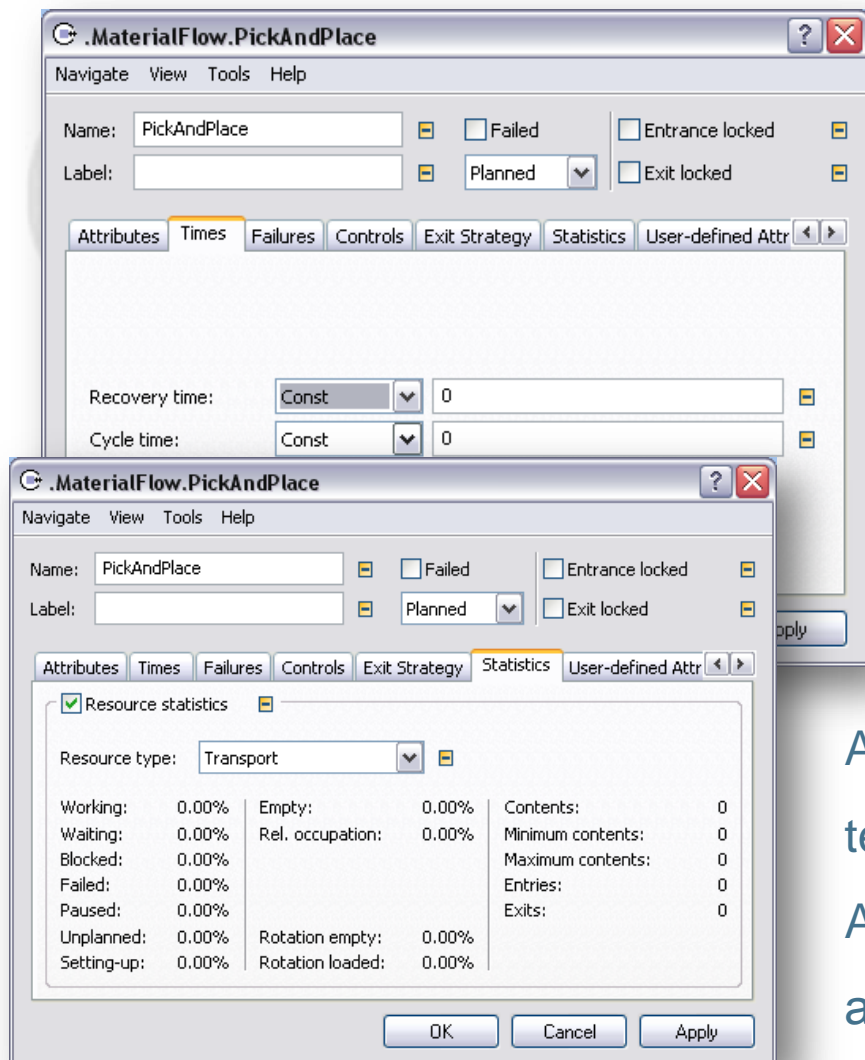
Az **Angles Table** (szög táblázat) tartalmazza a robot körül elhelyezkedő állomások szögét a robothoz képest.



A **Times Table** (idők táblázata) pedig azt tartalmazza, hogy az egyes állomások között mennyi időt vesz igénybe a robot forgása.

Mindkét táblázat egyéni igényeknek megfelelően testreszabható.

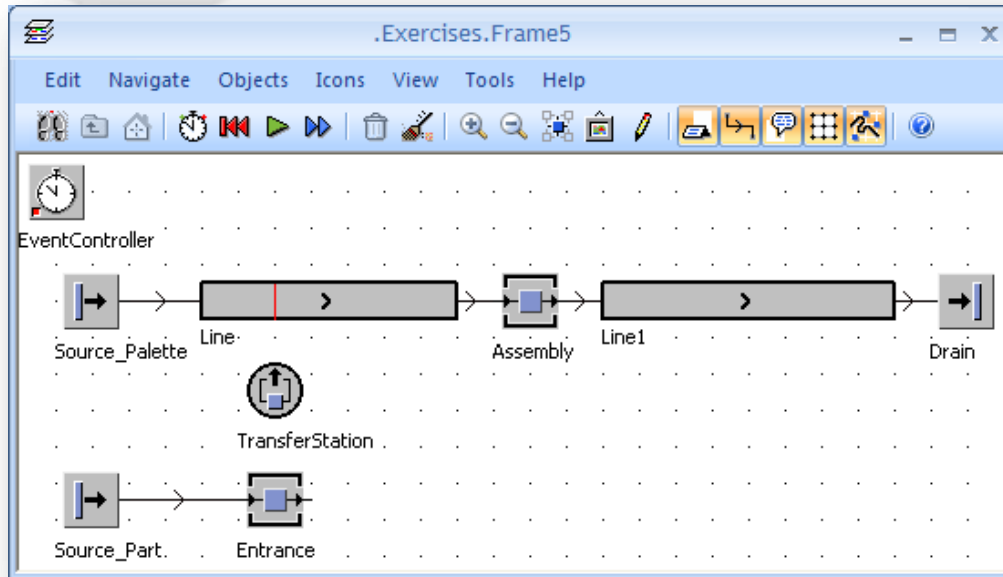
A **PickAndPlace** robot a (jobb klikk) **Calculate Angles** parancs segítségével számítja ki a szomszédos objektumok elhelyezkedési szögét.



A *Times* fül alatt a helyreállítási- és a teljes ciklusidő adható meg.

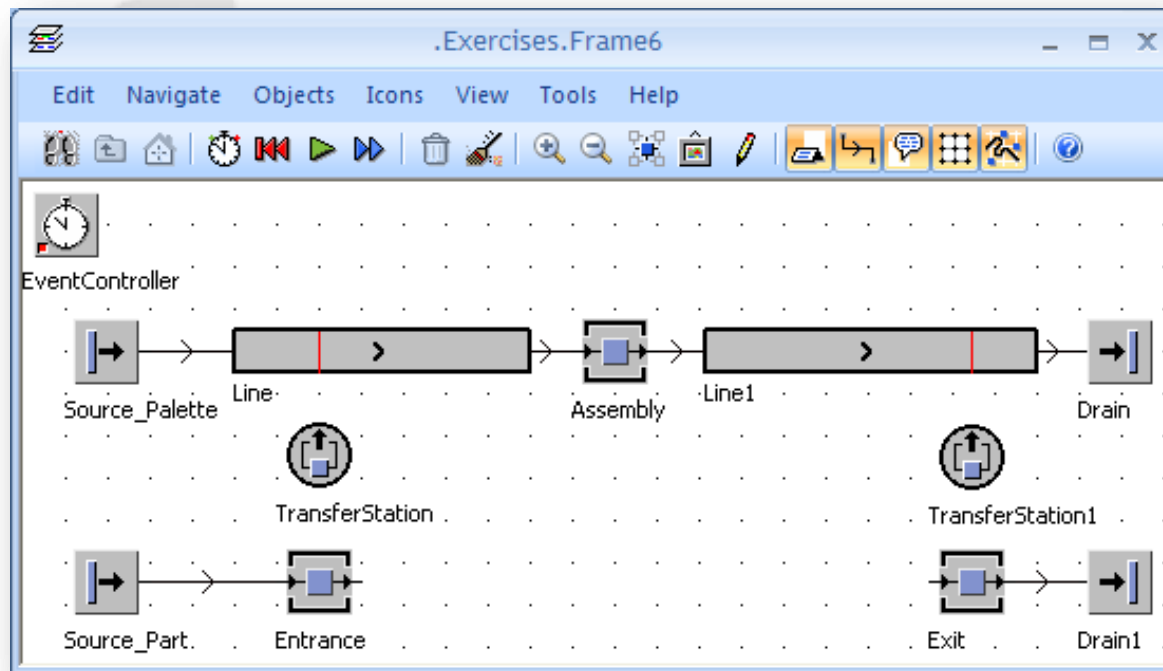
A *Statistics* fül alatt (mint minden anyagáram objektumnál) található a robotra vonatkozó statisztikák.

1. Adja hozzá a **Frame_5** nevű *Frame* - et az **Exercises** mappához.
2. Illessze be az ábrán látható állomásokat. A **SourcePart** *Source* hozza létre az asztallapokat, a **SourcePalette** *Source* pedig a palettákat.
3. Illesszen be két szállítószalagot (*Line*) és kapcsolja össze azokat a többi elemmel.



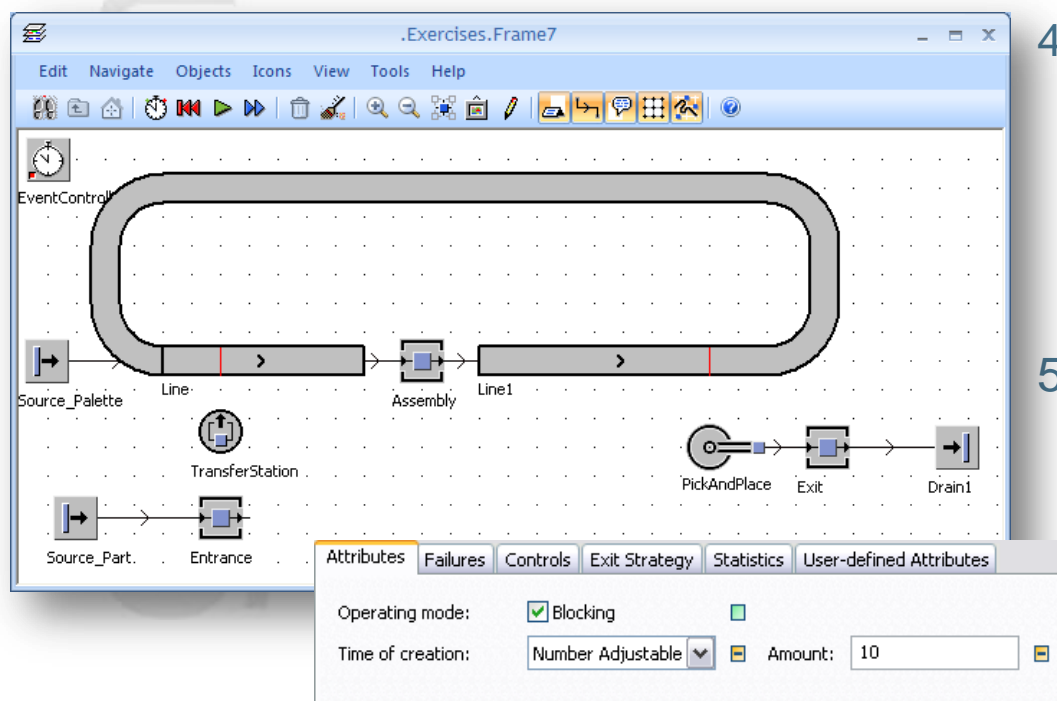
4. Illesszen be egy *TransferStation* objektumot, húzza rá az *Entrance* objektumot és a kapcsolódó szállítószalagot. Az *Entrance* a *Part Station*, a szállítószalag a *Target Station*, a szenzor helye pedig 2m-nél van.
5. Tesztelje a felrakodást. Ne feledje, hogy mindegyik palettára egy asztallapnak kell kerülni.

1. Készítsen másolatot a **Frame5** Frame - ből az Exercises mappában **Frame6** néven.
2. Adjon hozzá egy további **TransferStation**, és **Drain**, illetve egy **Exit** állomást.
3. Készítsen egy második Line szállítószalagot.
4. A *Target Station* (célállomás) az Exit, a *Part Station* a Line1, a szenzor legyen 8m környékén.



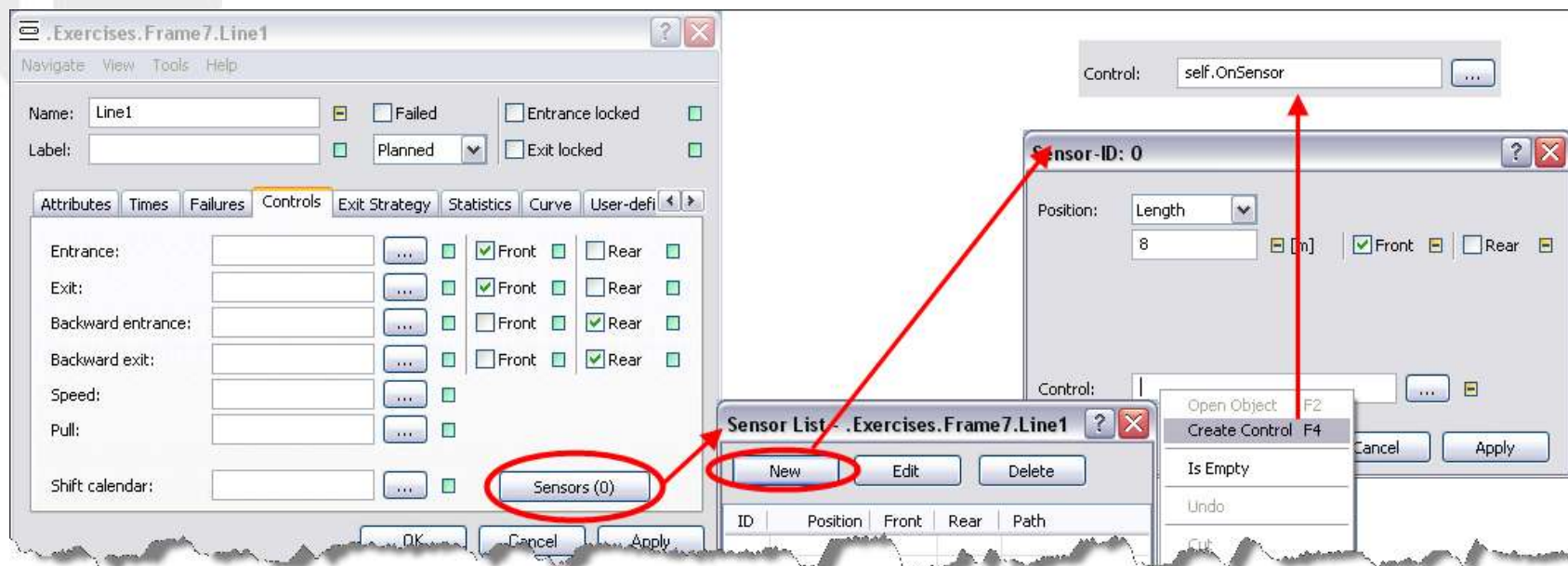
5. Engedje meg az MU - knak a feltorlódást a szállítószalagon, és ellenőrizze a lerakodás működését.
6. Adja meg a fel/lerakodás idejét (pl. 10 sec) a Times fülön a TransferStation állomások ablakában.

1. Készítsen másolatot a **Frame6** *Frame* - ből az **Exercises** mappában **Frame7** néven.
2. Törölje a **Drain** komponenst, és alakítsa át a második Line elemet teljes körre (jobb egérgattintással lehet pontokat hozzáadni).
3. Törölje a **TransferStation1** komponenst, és illesszen be a helyére egy **PickAndPlace** robot komponenst, majd kösse össze a Line1 és **Exit** állomásokkal.

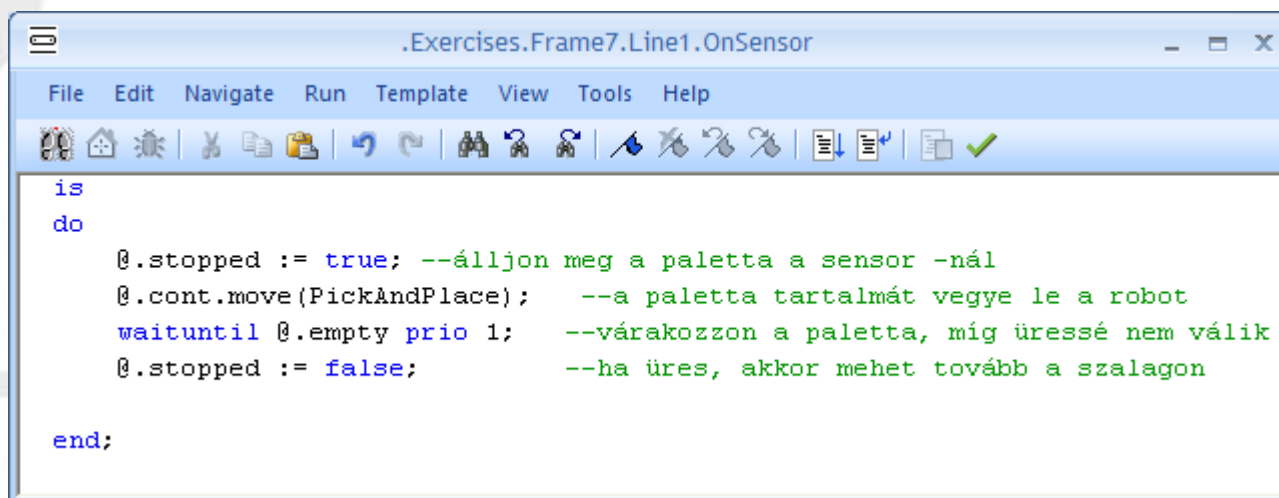


4. Az **Exit** állomás szerepe csupán az, hogy levegye az asztallapokat a konténerről (*Container*). Ezért a feldolgozási ideje legyen nulla.
5. A **Source_Palette** 10 darab palettát készít egyszer (használja a **Number adjustable** opciót).

6. Győződjön meg arról, hogy az MU-k fel tudnak torlódni.
7. A szállítószalagról való lerakodást a **PickAndPlace** robot fogja végezni. Ennek érdekében be kell állítani egy **Sensor** – t a **Line1** komponensen (kb. 8 m-nél).



8. A Control mezőbe kattintva jobb egérgombbal, válassza ki a Create Control parancsot. Az **OnSensor** nevű metódus fogja szabályozni a Sensor – t.



```
is
do
  @.stopped := true; --álljon meg a paletta a sensor -nál
  @.cont.move(PickAndPlace); --a paletta tartalmát vegye le a robot
  waituntil @.empty prio 1; --várakozzon a paletta, míg üressé nem válik
  @.stopped := false; --ha üres, akkor mehet tovább a szalagon
end;
```

A hosszorientált objektumok tartalmához (pl. a rajtuk lévő MU - khoz és konténerekhez) az alábbi metódusokkal lehet hozzáférni:

<code><object>.cont;</code>	a következő kilépésre kész MU - val tér vissza.
<code><...>.mu(3);</code>	a 3-as számú kilépésre kész MU - val tér vissza.
<code><...>[x,y].cont;</code>	a mátrix [x,y] pozíciójában lévő MU - val tér vissza.

Például:

<code><singleproc>.mu(1).move;</code>	az 1-es számú MU-t a kilépési állomásra mozgatja.
<code>@.move(<parallelproc>[2,2]);</code>	az MU-t az objektum [2,2] pozíciójára mozgatja.

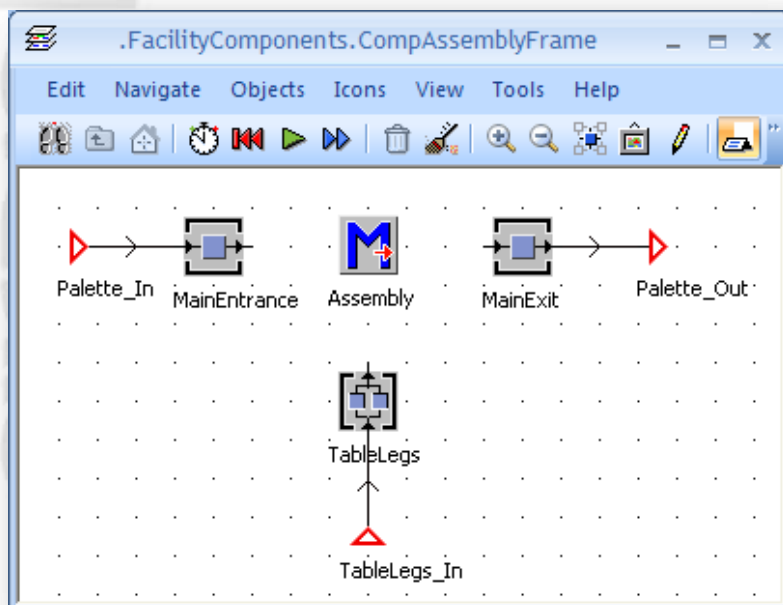


Csak olyan objektum tartalmához lehet hozzáférni, amelyek éppen tartalmaz MU - kat!

8a. lecke: A lábak kapcsolása az asztallapokhoz

Ebben a *Frame*-ben a szerelési folyamatot modellezzük. Az asztallapok palettákon érkeznek, lerakódásra kerülnek, majd az asztallábak hozzájuk kapcsolódnak.

1. Készítsen egy **CompAssemblyFrame** *Frame* - et a **FacilityComponents** mappában és illessze bele az ábrán látható objektumokat.
2. A **ParalleProc** négy állomással rendelkezik a 4 asztalláb számára, és nincs feldolgozási ideje.



3. Programozza a metódust, amely az összeszerelést modellezi. Hogyan lehet a különböző eseményeket kezelni?
4. Készítsen ikont a Frame számára és animációs pontokat.

8a. lecke: A lábak kapcsolása az asztallapokhoz

The screenshot displays the graphIT software interface with three overlapping windows:

- Code Editor Window:** Titled `.FacilityComponents.CompAssemblyFrame.Assembly`. It contains the following code:

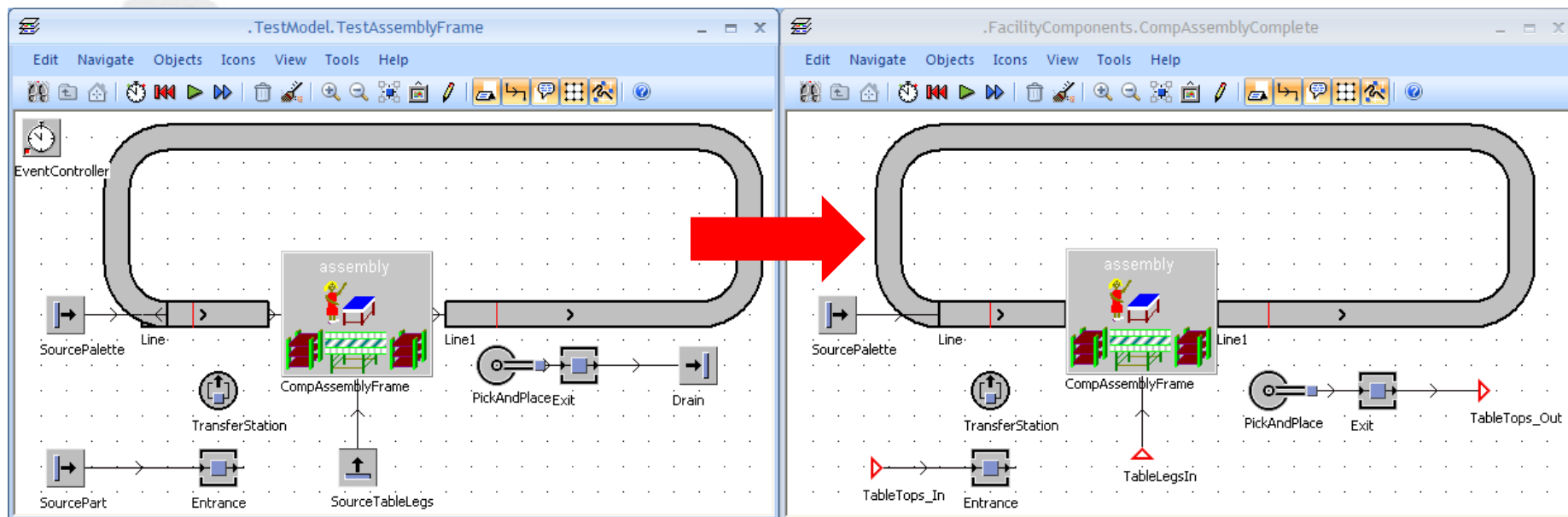
```
is
  i : integer;
do
  waituntil mainEntrance.occupied
  and mainExit.empty
  and TableLegs.full prio 1;

  mainEntrance.cont.move(mainExit);

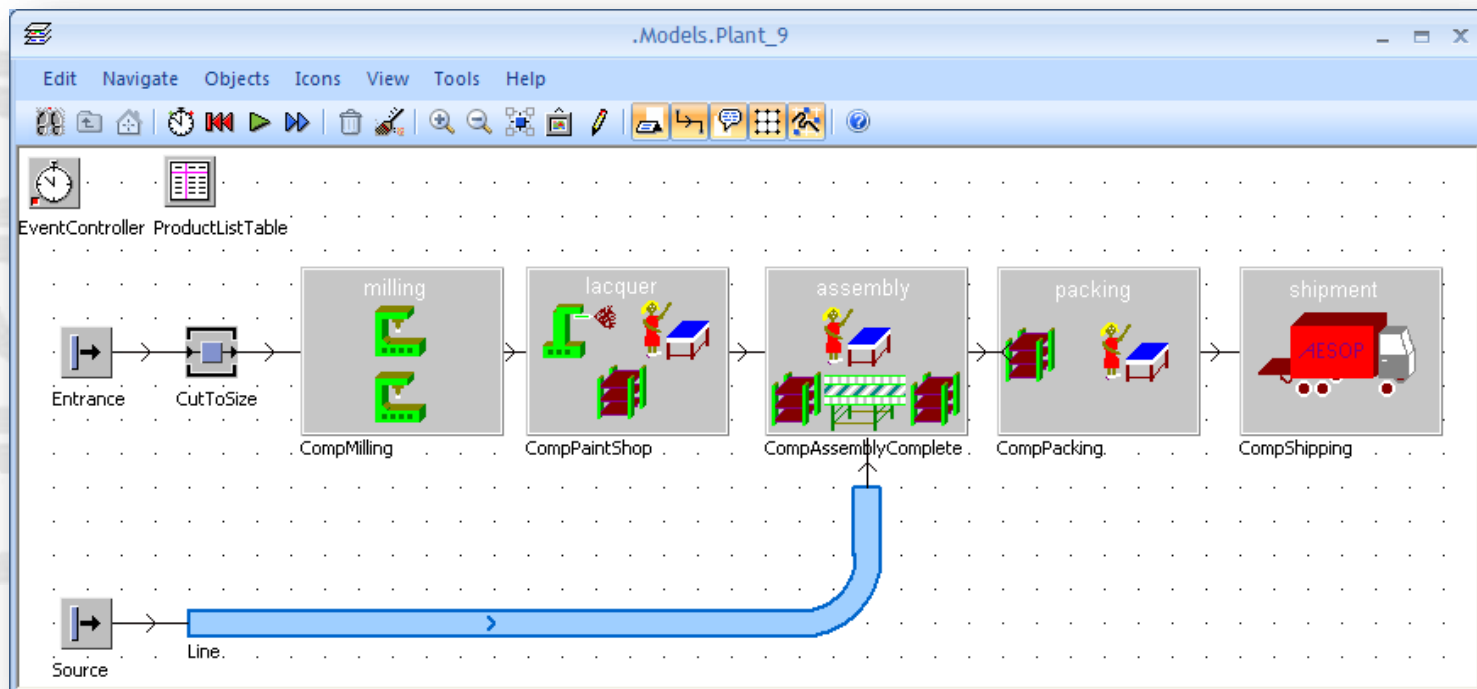
  for i := 1 to 4
  loop
    TableLegs.cont.move(mainExit.cont.cont);
  next;
end;
```
- MainEntrance Configuration Window:** Titled `.FacilityComponents.CompAssemblyFrame.MainEntrance`. It shows the 'Controls' tab. The 'Exit' field is set to 'Assembly' and is circled in red. Other fields include 'Entrance' (empty), 'Set-up' (empty), and 'Failed' (unchecked). Checkboxes for 'Entrance locked' and 'Exit locked' are also present.
- TableLegs Configuration Window:** Titled `.FacilityComponents.CompAssemblyFrame.TableLegs`. It shows the 'Attributes' tab. The 'X-dimension' and 'Y-dimension' fields are both set to '2' and are circled in red. Below the dimensions, the text $2 \times 2 = 4$ is displayed in red.

8b. lecke: Az Assembly és CompAssemblyComplete tesztje

1. Készítsen másolatot a **Frame7** Frame - ből, és legyen a másolat neve: **TestAssemblyFrame**.
2. Cserélje ki az *Assembly SingleProc* objektumot a **CompAssemblyFrame** nevű *Frame* - re, ami az imént készült el.
3. Ellenőrizze a szerelés funkcionalitását.
4. Cserélje ki a *SourcePart*, *SourceLegs* és *Drain* objektumokat *Interface* objektumokkal.



1. Készítsen másolatot a Plant_8 komponensből Plant_9 néven
2. Cserélje ki a **CompAssembly** nevű *Frame* komponenst a **CompAssemblyComplete** nevű *Frame* komponenssel a szimulációs modellben.
3. Indítsa el a szimulációt és figyelje meg, hogy a várt események jelennek-e meg a képernyőn.

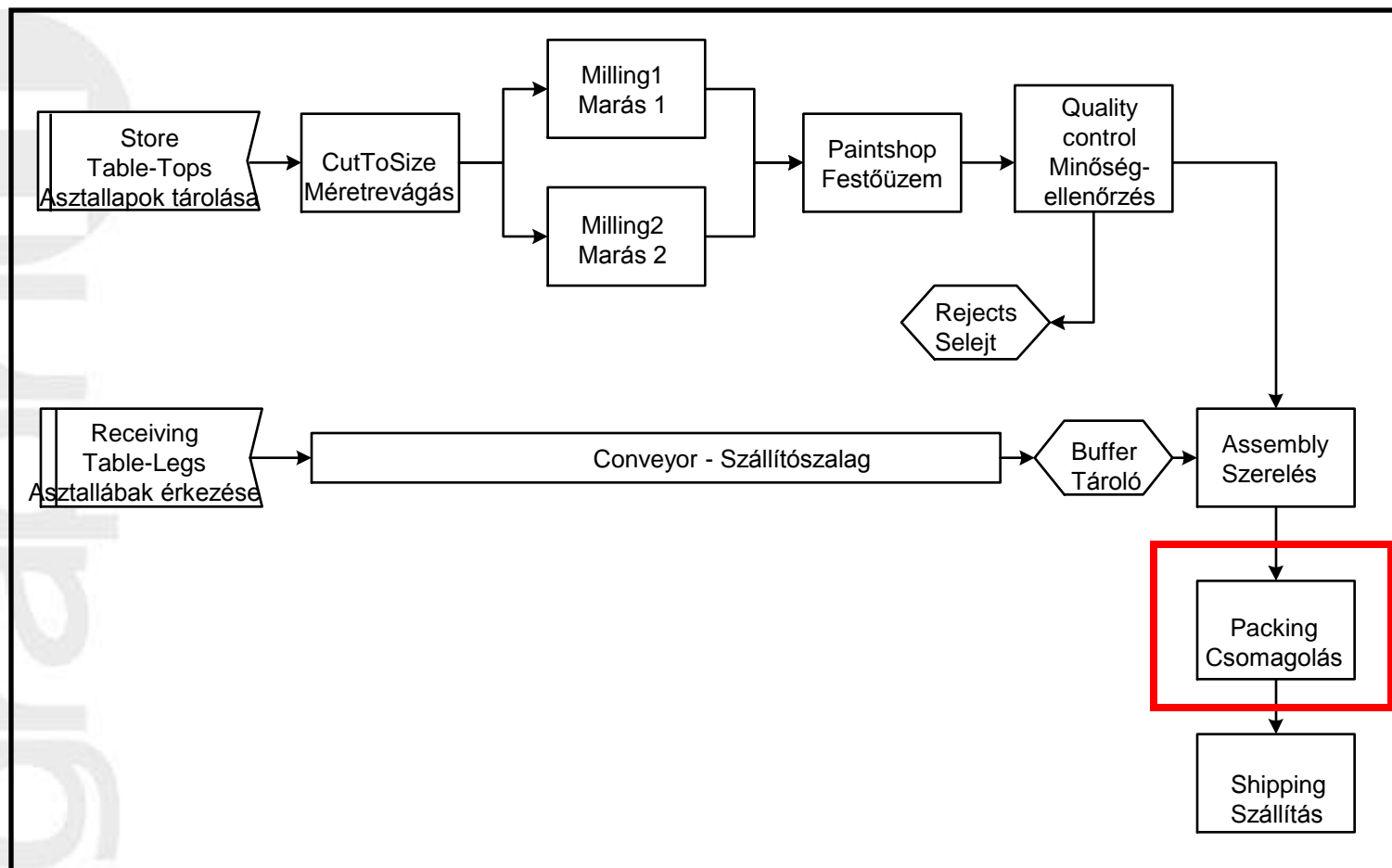


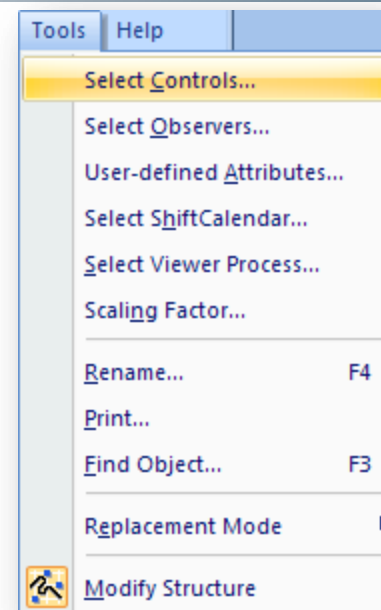
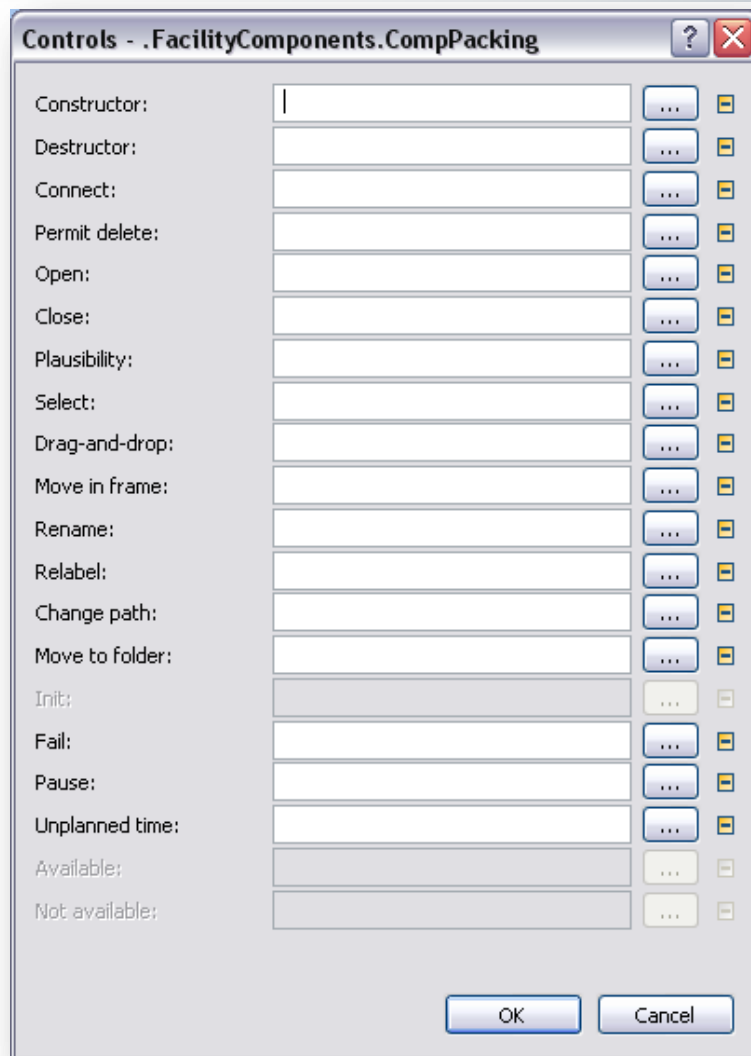


14. fejezet

Metódusok meghívása

Frame kontrollok

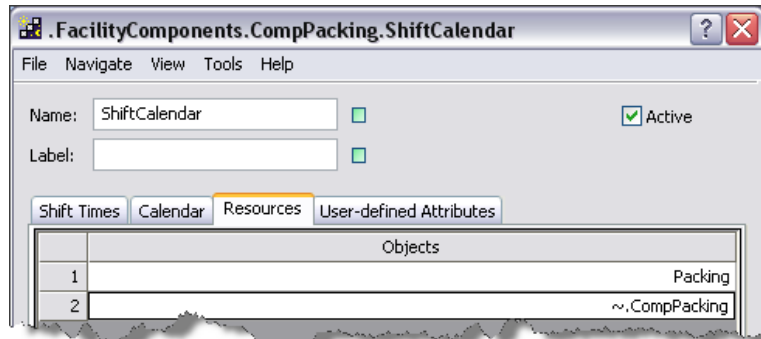




Válassza ki az

Tools > Select Controls

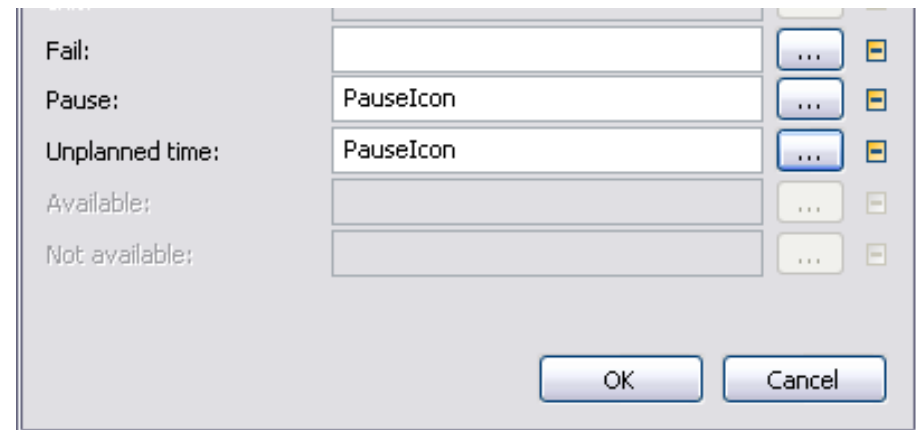
menüpontot és adja meg a metódusok neveit a kívánt helyeken. Ezek arra szolgálnak, hogy a *Frame* objektumok eseményeihez is lehessen készíteni függvényt.



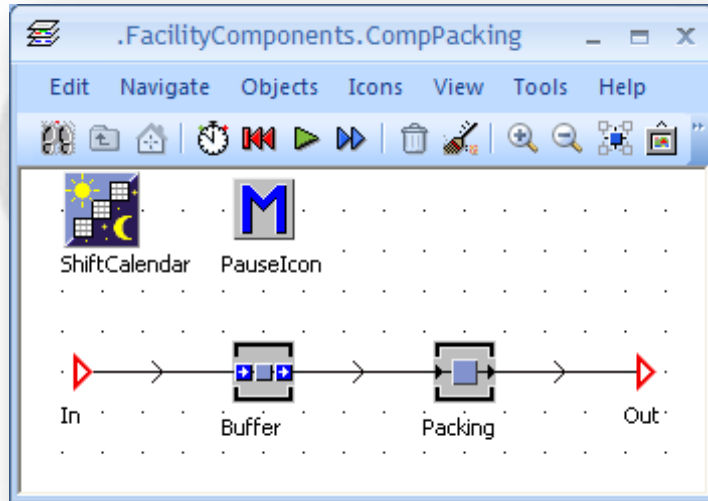
A *ShiftCalendar* használható *Frame* kezelésére is nem csupán egyes állomásokra. Ezért a *ShiftCalendar* *Resources* fülén a “~.” szimbólummal megadható a keret neve.

Ez után készíthető egy metódus, amely a frame ikonját megváltoztatja.

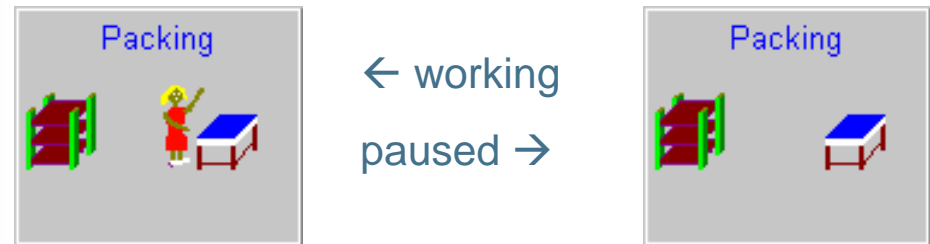
A metódusnak akkor kell aktivizálódnia, amikor a *ShiftCalendar* átállítja a frame *pause* vagy *unplanned* attribútumát. Az *unplanned* olyan idő, amikor nincs műszak, szünet van.



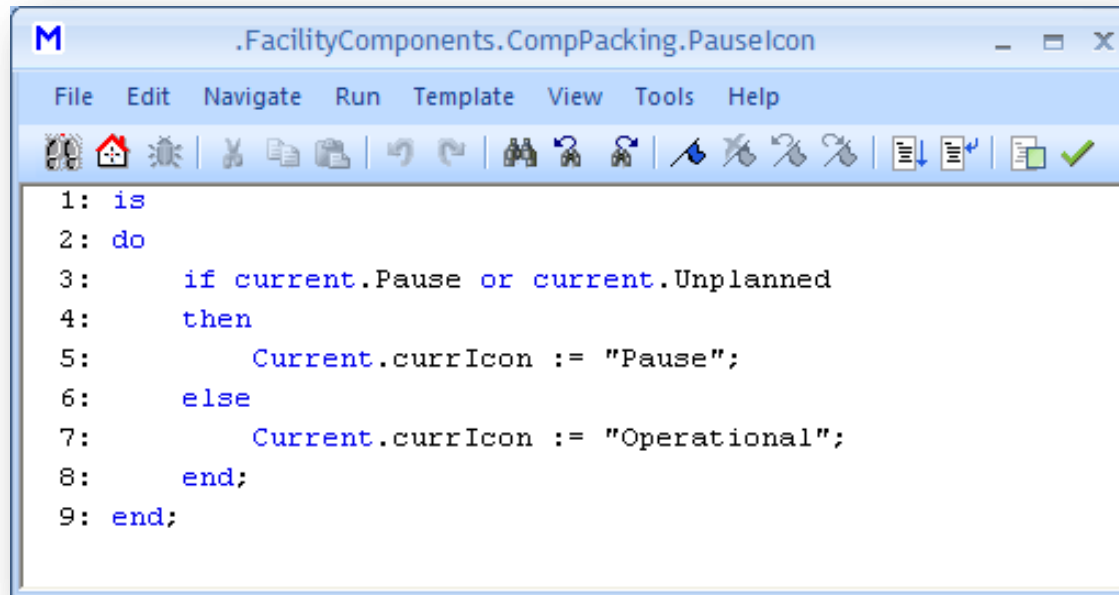
A szövegmezőben a metódus a **Pause/Unplanned time** esetén kontrollálja az állapotváltozások esetén a történéseket.



1. Készítsen egy másik ikont a **Packing Frame** számára az ikonszerkesztőben egy másik névvel.



2. Adja meg a **PauseIcon** metódust, mint a **CompPacking Frame** kontrollját.
3. Aktivizálja a *ShiftCalendar* objektumot a **CompPacking Frame** - hez, és a **PauseIcon** metódust a *Frame* pause eseményéhez.
4. Készítse el a megfelelő metódust, amely aktivizálja a megfelelő ikont, amikor a *Frame* állapota módosul.

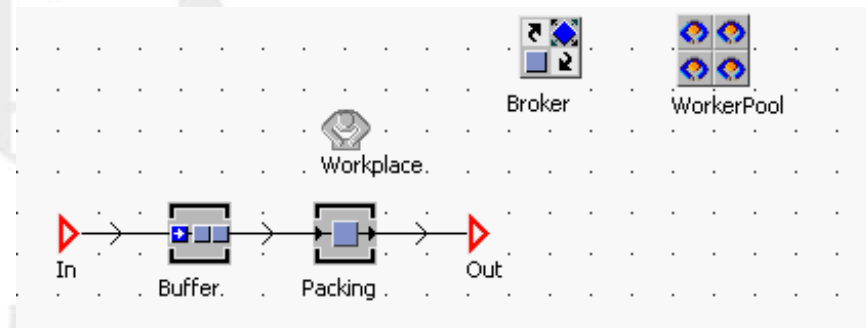


```
1: is
2: do
3:     if current.Pause or current.Unplanned
4:     then
5:         Current.currIcon := "Pause";
6:     else
7:         Current.currIcon := "Operational";
8:     end;
9: end;
```



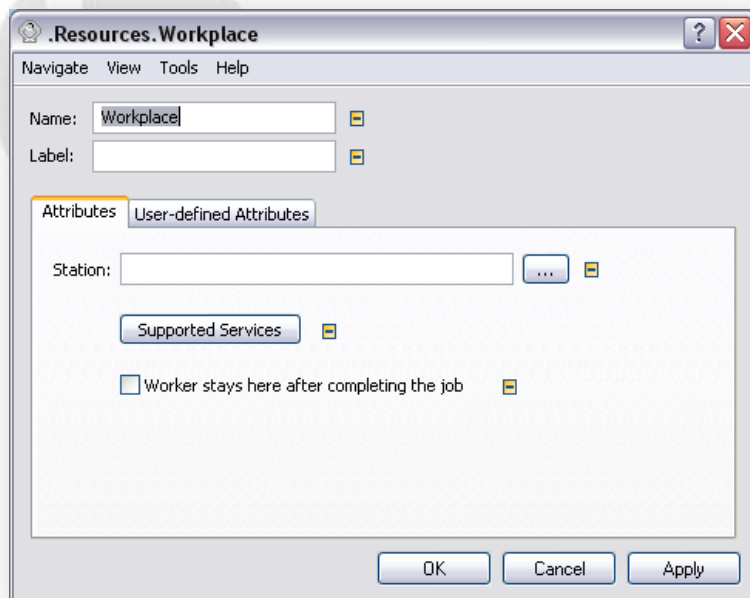

15. fejezet

Dolgozók (worker) hozzáadása



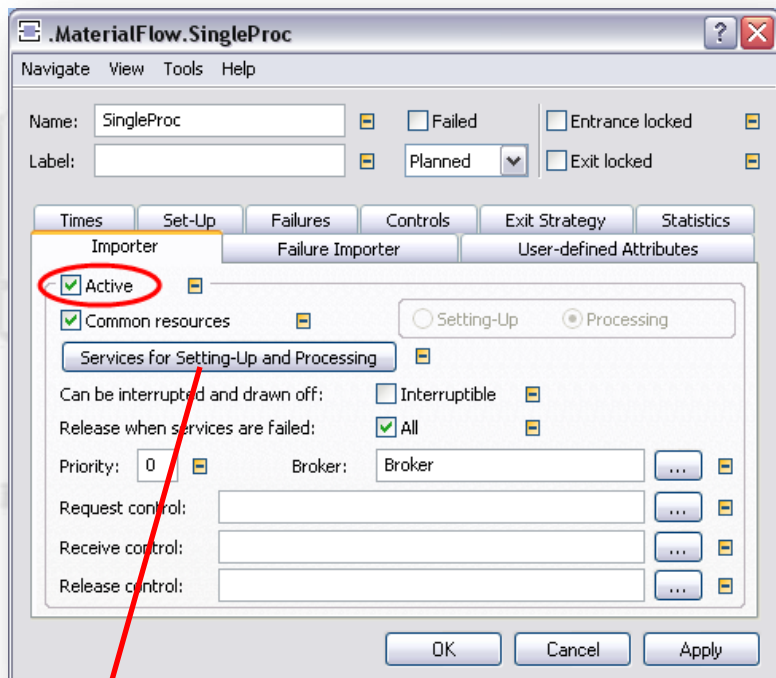
Ebben a részben a *Packing* komponenshez dolgozók (worker) objektumok kerülnek hozzáadásra.

Dolgozó kezeléséhez a kívánt helyen a következő komponensek szükségesek: **Workplace**, **Broker** és **WorkerPool**. Mindegyik objektum a **Resources** fülön található a Plant Simulation komponenspalettán.



Egy állomást a munkahelyhez (workplace) rendelni a következő módon lehet:

- Meg kell adni az állomás útvonalát a **Station** mezőben.
- Az állomást rá kell húzni a munkahelyre (húzd és ejtsd módszerrel).
- A munkahelyet az állomás közelében elhelyezve automatikusan hozzárendelődik az állomás a munkahelyhez.

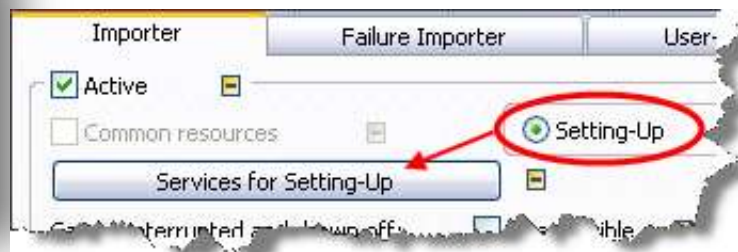
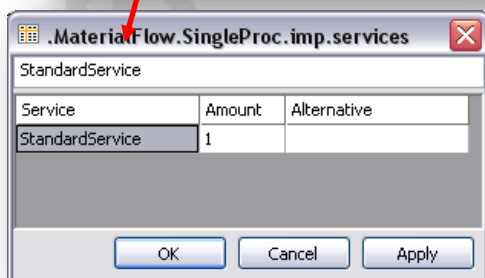


Meg kell nyitni az állomás ablakát (*SingleProc*, *ParallelProc*) és ki kell választani az **Importer** fület.

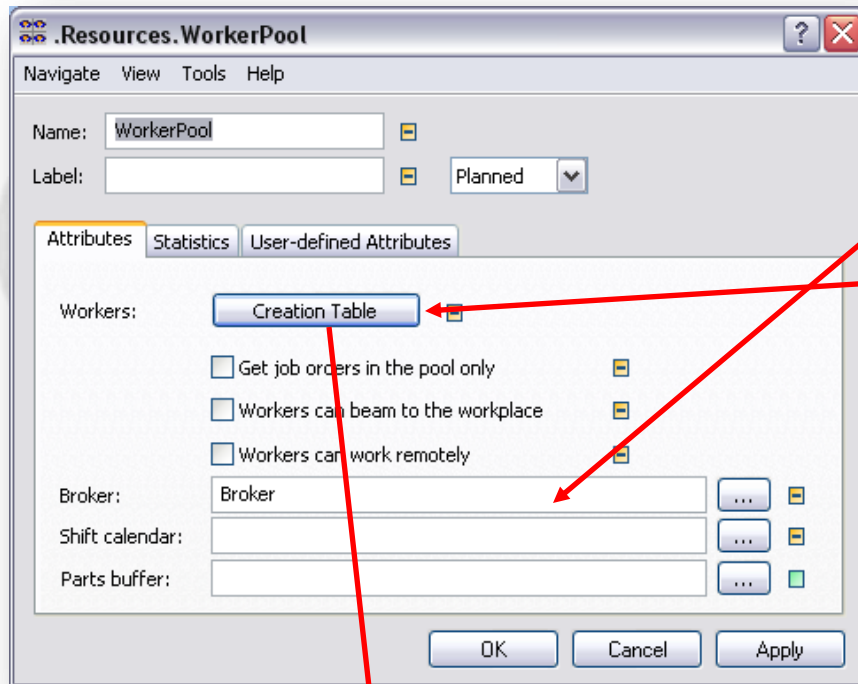
Aktivizálni kell az Importer - t.

Ki kell választani azt a Broker - t, amely a szolgáltatásokat kezeli.

Meg kell nyitni a Services táblázatot, és meg kell adni a megfelelő értéket az alternatívákra.



Különválasztható a Setting-up (beállítás) és a Processing (folyamatok).

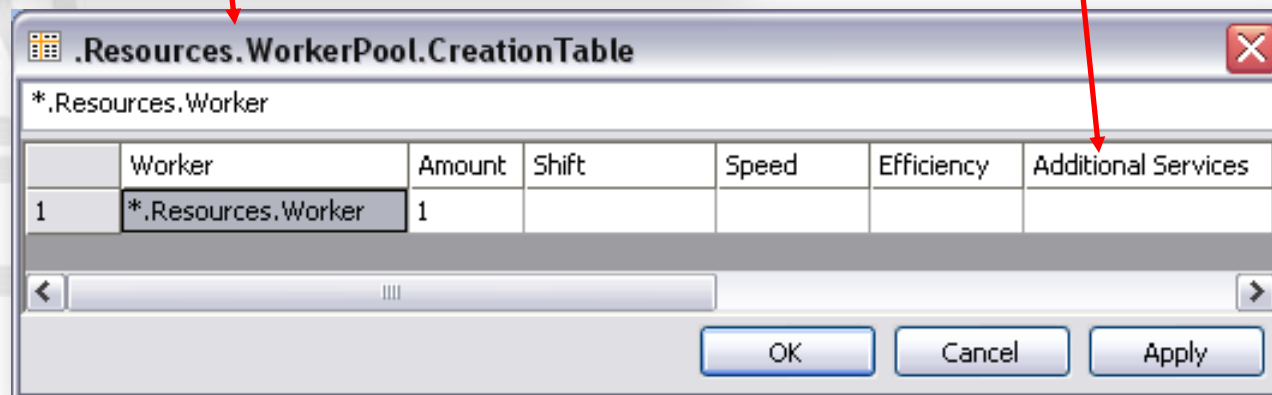


Meg kell nyitni a *WorkerPool* - t

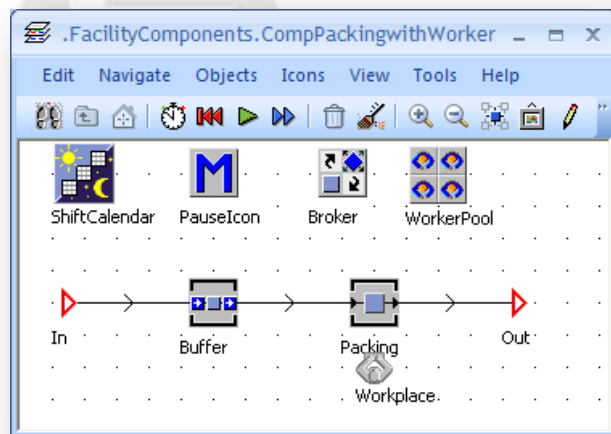
Meg kell adni a *Broker* objektumot.

Meg kell nyitni a **Creation Table** - t
(készítési táblát) a szükséges
dolgozók definiálására.

Meg kell adni a dolgozó által nyújtott
szolgáltatást.



	Worker	Amount	Shift	Speed	Efficiency	Additional Services
1	*,Resources.Worker	1				



1. Készítsen másolatot a CompPacking Frame -ből, legyen a neve CompPackingwithWorker.
2. Illesszen be egy WorkPlace, WorkerPool és Broker komponenst.
3. Kapcsolja össze őket (Broker és Packing és WorkerPool, Packing és WorkPlace, ShiftCalendar és WorkerPool)
4. A Packing állomásnál aktivizálja az Importer - t és kapcsolja össze a Broker komponenssel
5. Készítsen egy teszt környezetet, majd készítsen egy új Plant komponenst a dolgozóval csomagolást használva.

Ellenőrizze le, hogy mi történik, ha a Packing állomást nem kontrollálja a ShiftCalendar.

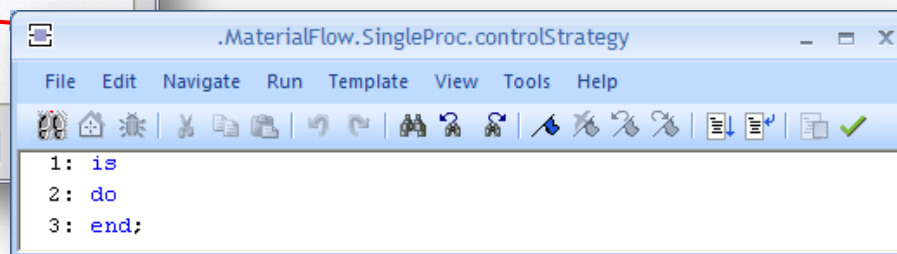
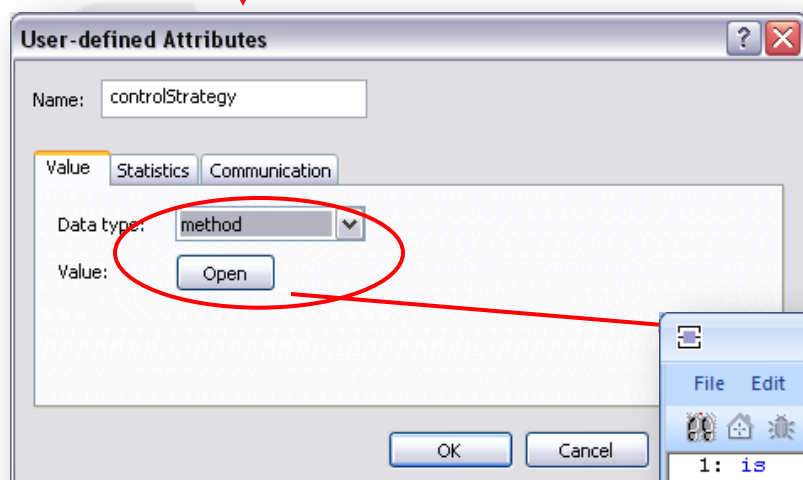
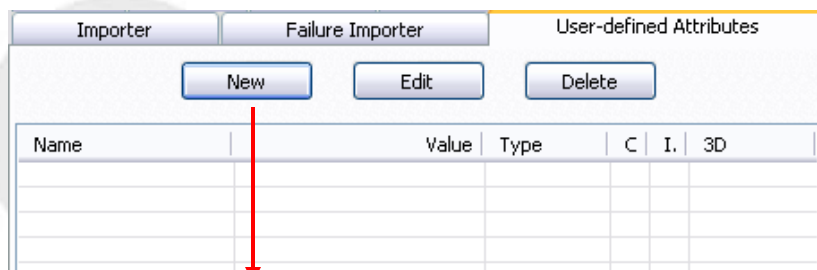


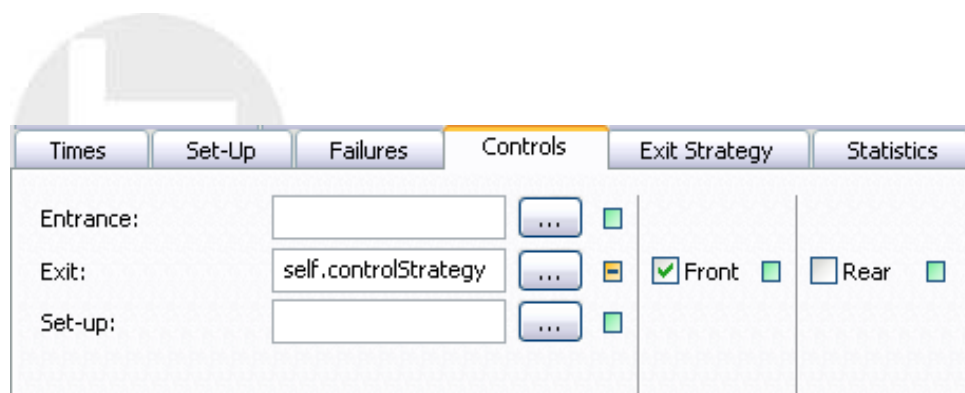
16. fejezet

Egyéb funkciók

Metódus felvehető egyedi jellemzőként is, ha az adattípus **method**-ként van megadva. Ezek olyan metódusok, amelyek az objektumban tároltak és kezeltek.

Az ilyen metódusokkal az adott anyagáram objektumra jellemző metódusok készíthetők.





A kifejezés:

`self.name_of _Method`

egyedi jellemzőként programozott
metódust hív meg.

Ez a metódus nem egy valódi *Method* objektum, de egy metódus típusú attribútuma egy objektumnak.

Ez a metódus az objektum része, ami megmarad az objektum mentésekor, másolásakor is.

- `callEvery(<path>, <method>, <argument>)`

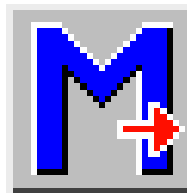
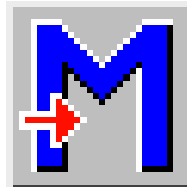
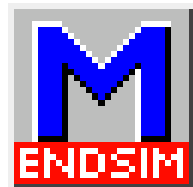
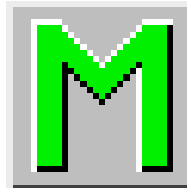
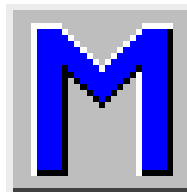
Meghívja azokat a metódusokat, amelyeket a `method` paraméter megad a `<path>` *Frame*-ben. Az argumentumok a metódus hívásának megfelelően adódnak át.

- `ref(<method>).methCall(<time/date/datetime>, <argument>)`

Meghívja a metódust a `<time>` által megadott idő elteltével. A `<datetime>` vagy `<date>` paramétereknél a Plant Simulation az *EventController* alapján számol. Az argumentumok a metódus hívásának megfelelően adódnak át.

- `wait(<real>)`

A hívások futásának láncát megszakítja a `<real>` paraméter által megadott másodpercig. Az *EventController* egy *MethWakeup* eseményt kap.



Jellemzők:

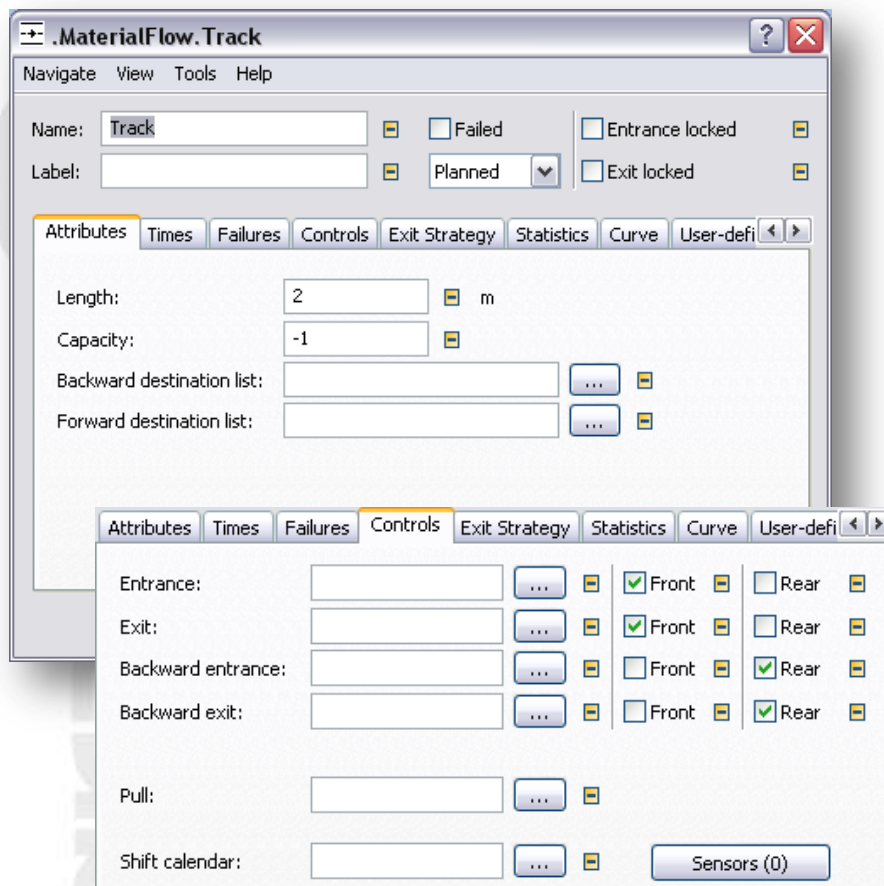
- Ikon: 

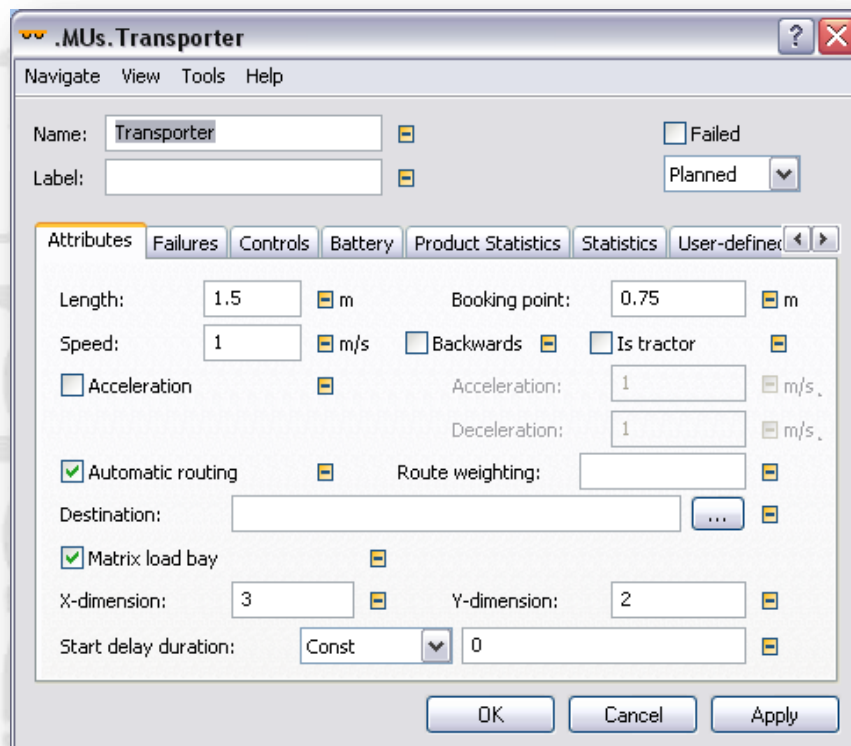
- Kapacitás: tetszőleges

- Hosszorientált anyagáram objektum

Megadható a hossza (**Length**) és a kapacitása (**Capacity**). Az egyetlen mozgó anyagáram objektum a *Transporter* ami közlekedni tud rajta. A *Container* és az *Entity* a helyén marad és nem mozog rajta.

A destination list tartalmazza azokat az objektumokat, amelyekhez a *Track* hozzá tud férni.

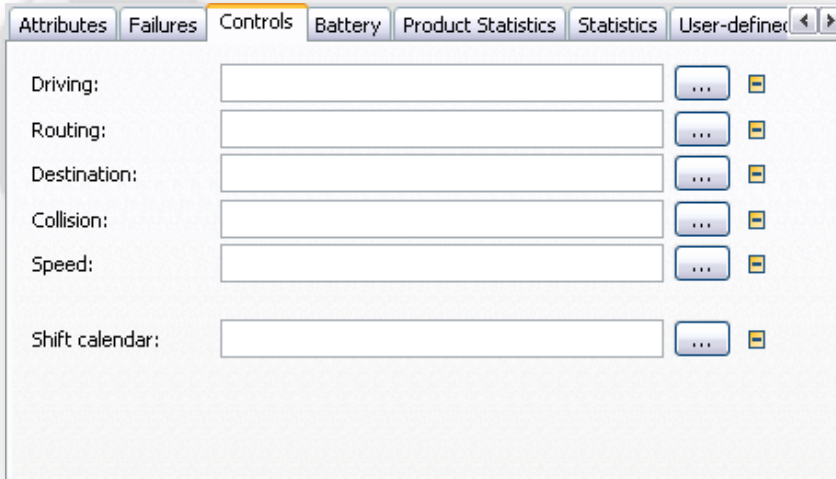




Megadható a gyorsulás (**Acceleration**) a végsebesség (**Final speed**) a szövegmezőkben.

Megadható, hogy a feltöltés módja: **Matrix load bay** vagy **Load bay length**, a kapacitás és a szenzorok helye.

A **Battery** fül alatt megadhatók azok a jellemzők, amelyek az üzemanyagcellával (akkumulátorral) működő *Transporter* objektumokat jellemzik.



A **Controls** fülön megadhatók azok a metódusok, amelyek a *Transporter* és a *Track* objektumokat kontrollálják.

Driving control: annak a *Track* objektumnak a megadása, amelyen a *Transporter* mozog.

Destination: ha nincs kilépési kontroll a *Track*en, akkor azt adja meg, hogy kilépési ágra mozog a *Transporter* a pálya végén.

Destination control: egyszer aktiválódik, amikor a *Transporter* elérte a célját.

Collision control: akkor aktiválódik, ha két *Transporter* összeütközik.

Speed control: akkor aktivizálódik, ha a *Transporter* eléri a végsebeségét.



A *Source* tud készíteni *Transporter* objektumokat az *Entity* vagy a *Container* objektumokhoz hasonlóan.

A Plant Simulation ezt a következő *Track* objektumra mozgatja az összekötésekkel vagy metódussal.

A *Transporter* emellett képes a *Track*-en mozgásra a saját energiájából. Ha a *Track* objektumok össze vannak kapcsolva, akkor az egyikről a másikra mozog.

Használja a következő formátumot a *Transporter* megfelelő objektumra helyezéséhez:

```
.MUs.transporter.create(track) ;
```

```
.MUs.transporter.create(track,3.1) ; -- a hely méterben a Tracken
```

A *Transporter* a *Track* - en azokkal a beállításokkal halad, amelyek a szimuláció futtatásakor meg lettek adva.

A *Transporter* megállítható, mozgása folytatható, és visszafelé is mozgatható.



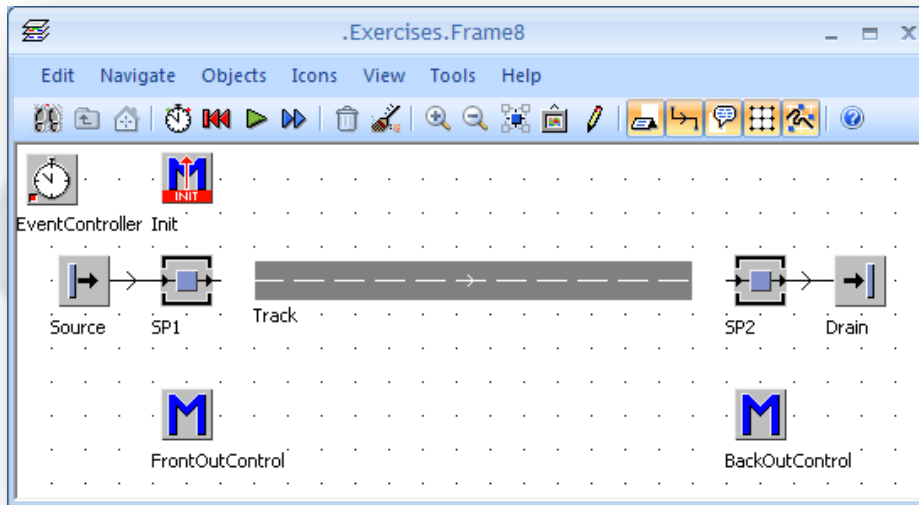
A *Track* `cont` metódusa használható a *Transporter* meghívására. A `@` azonosító a *Transporter* - t hívja meg:

```
@.stop;
```

```
@.continue;
```

```
@.backwards := true;
```

```
track.cont.stop; ...
```

A *Transporter* a *Conveyor1* által adott alkatrészek *Conveyor2*-höz szállítására szolgál, egyszerre egy alkatrészt szállítva.

1. Készítsen egy *Frame* -et és illessze bele az objektumokat az ábra szerint. A *Track* 10 méter hosszú.
2. Készítsen egy *Init* metódust, amely elkészíti a *Transporter* - t a *Track* - en a szimuláció indulásakor.
3. Melyik objektumnak kell ezt a szimulációs modellben kontrollálni? Programozza a **loading** és **unloading** metódusokat ezen események koordinálására.
4. Győződjön meg róla, hogy a metódusok megfelelően működnek, akkor is, ha az MU-k feltorlódnak a *Drain* előtt.

The screenshot displays the GraphIT software interface with four windows open, illustrating the configuration of a transporter system.

.Exercises.Frame8.FrontOutControl

```

1: is
2: do
3:   if @.empty then
4:     @.backwards := true;
5:   else
6:     waituntil sp2.empty prio 1;
7:     @.cont.move(sp2);
8:     @.backwards := true;
9:   end;
10: end;
    
```

.Exercises.Frame8.BackOutControl

```

1: is
2: do
3:   waituntil sp1.occupied and sp1.cont.finished prio 1;
4:   sp1.cont.move(@);
5:   @.backwards := false;
6: end;
    
```

.Exercises.Frame8.Track

Attributes: Name: Label:

Failed: ☐ Entrance locked: ☐ Exit locked: ☐

Planned: ☒

Attributes | Times | Failures | **Controls** | Exit Strategy | Statistics | Curve | User-defi

Entrance:	Exit:	Backward entrance:	Backward exit:	Front	Rear
<input type="text" value="FrontOutControl"/>	<input type="text" value="FrontOutControl"/>	<input type="text" value="FrontOutControl"/>	<input type="text" value="FrontOutControl"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="BackOutControl"/>	<input type="text" value="BackOutControl"/>	<input type="text" value="BackOutControl"/>	<input type="text" value="BackOutControl"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

.Exercises.Frame8.Init

```

1: is
2: do
3:   .MUs.Transporter.create(Track, 1.5);
4: end;
    
```



Tanfolyam

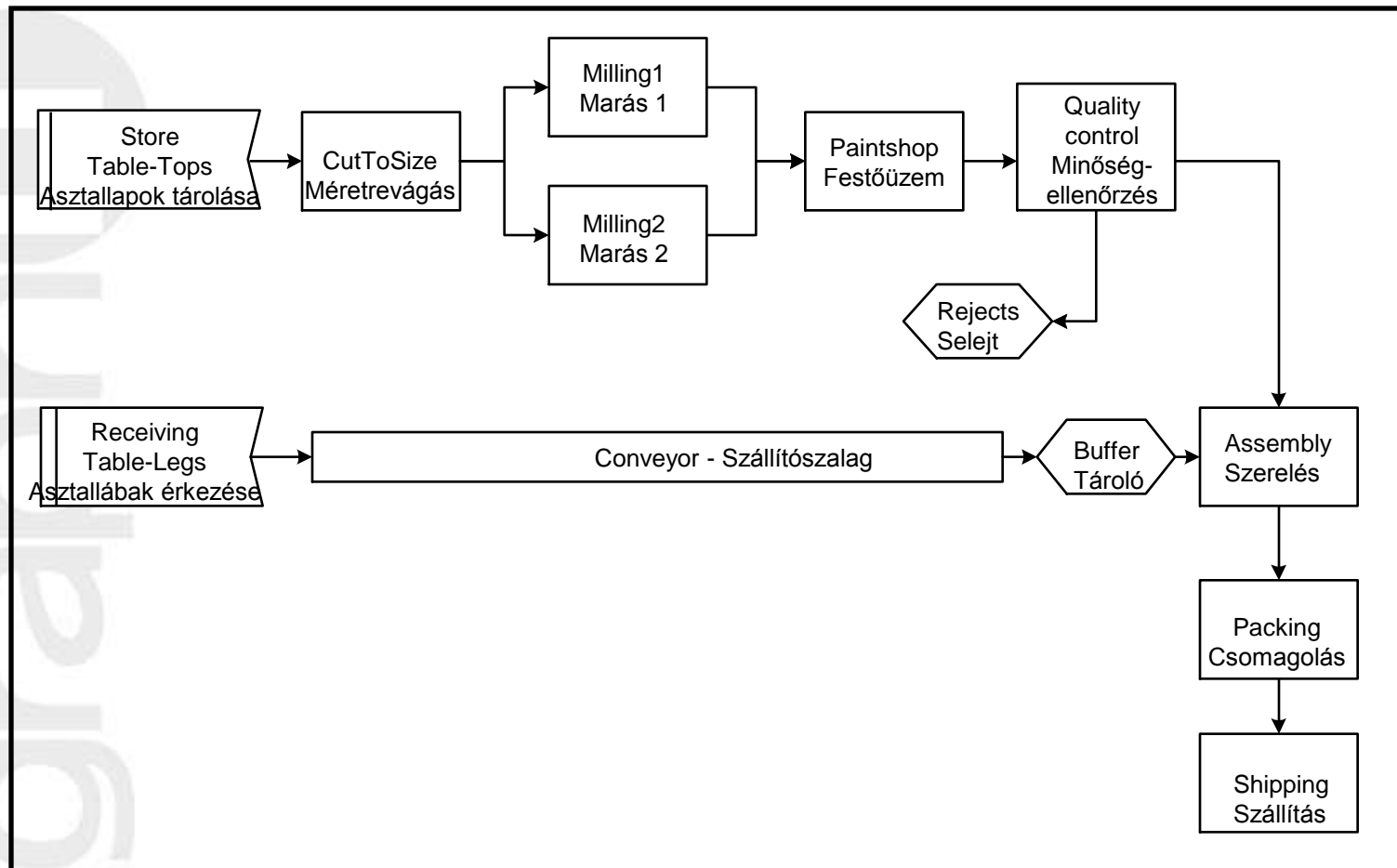
Plant Simulation 9

3. modul: Adatkezelés

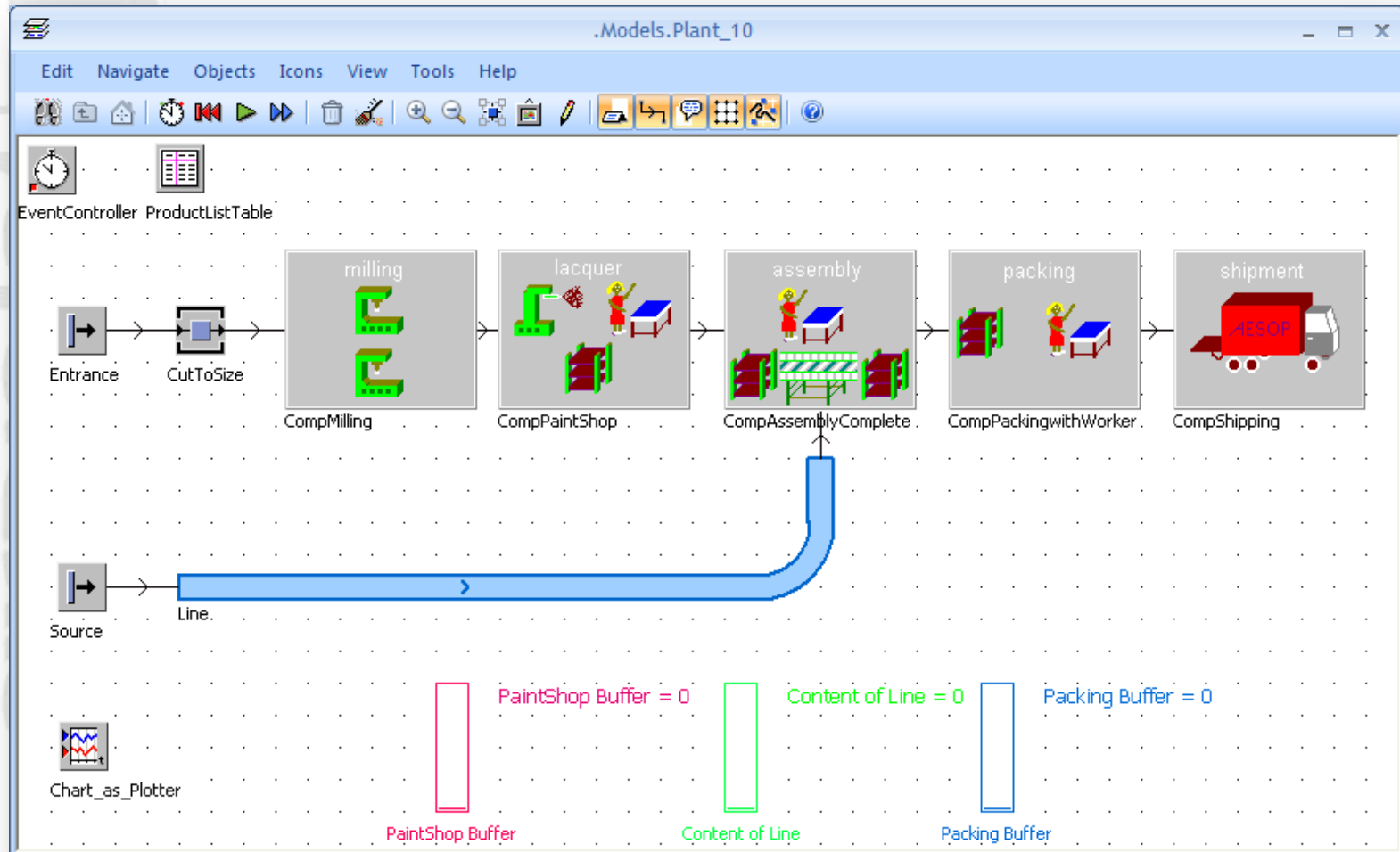


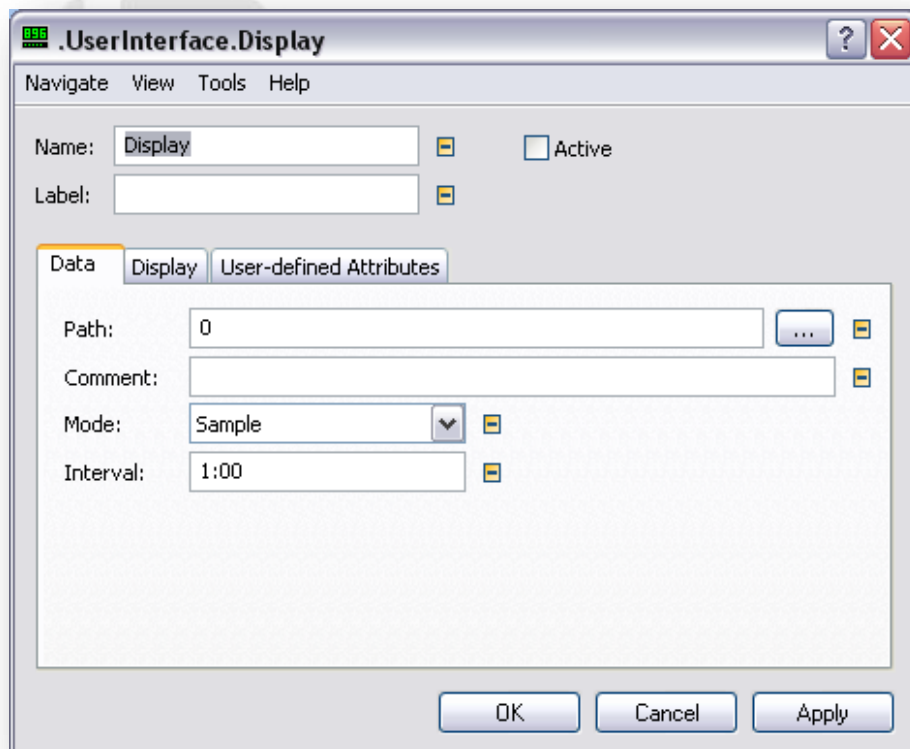
1. fejezet

Statisztika gyűjtő objektumok



Meg akarjuk vizsgálni a *Bufferek* terheltségét a gyárunkban. Ennek érdekében több *Gauge* objektumot és *Plotter* objektumot illesztünk a szimulációs modellbe.





Jellemzők :

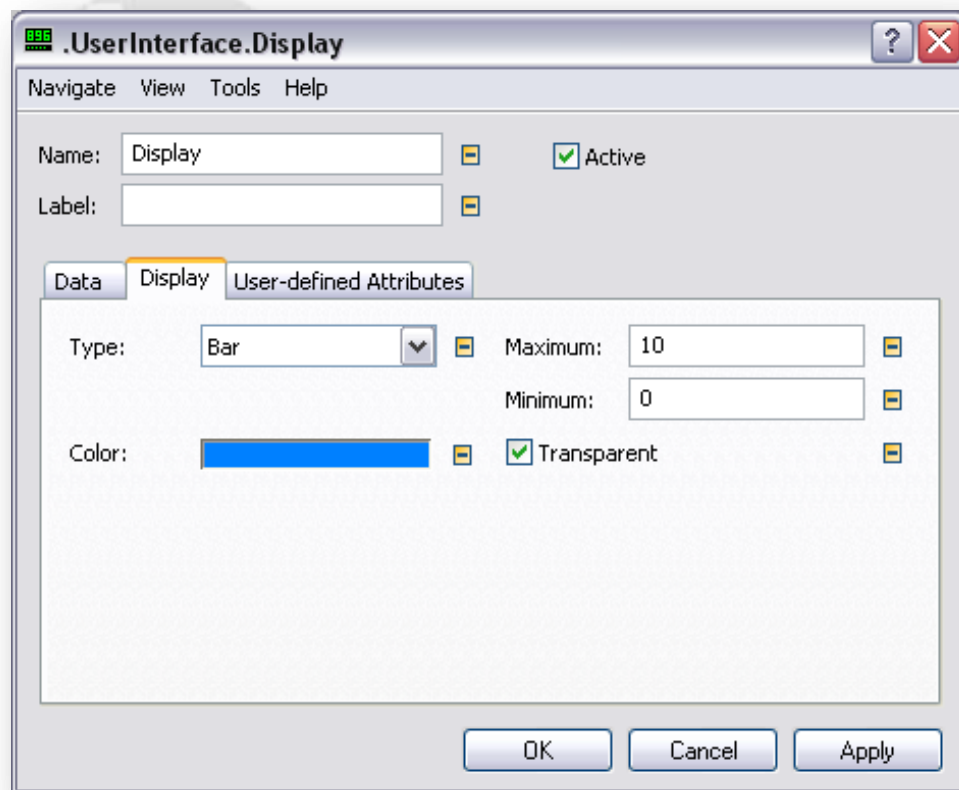
- Ikon: 
- Kapacitás: 0
- Aktív felhasználói felület objektum

A *Display* értékek szöveges, illetve kör- vagy oszlopdiagramm formájú megjelenítésére szolgál.

A **Path** mezőben a figyelni kívánt objektum és a figyelni kívánt attribútum adható meg.

A *Display* a **Comment** mezőben megadott szöveget a szimulációs modellben a *Frame* - en megjeleníti.

Megjegyzés: a **Path** megadása előtt az **Active** állapotot be kell állítani.



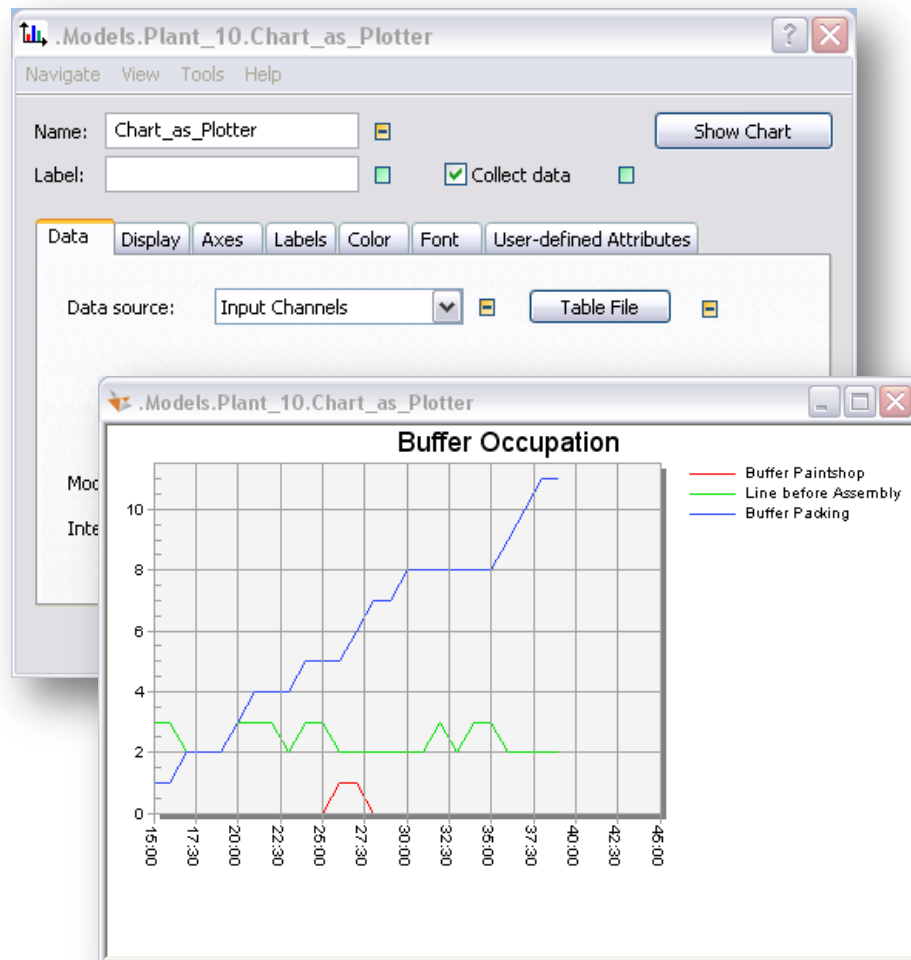
Először meg kell adni a megjelenítés típusát: **Text** (szöveg), **Bar** (oszlopdiagram) vagy **Pie** (kördiagram). Mostani célunkhoz a Bar megfelelő.

Diagramok esetében megadhatók a **Maximum** és **Minimum** értékek a megjelenítés áttekinthetőbbé tételére.

Továbbá megadható az összegyűjtött értékek megjelenítésére szolgáló szín (**Color**) is.

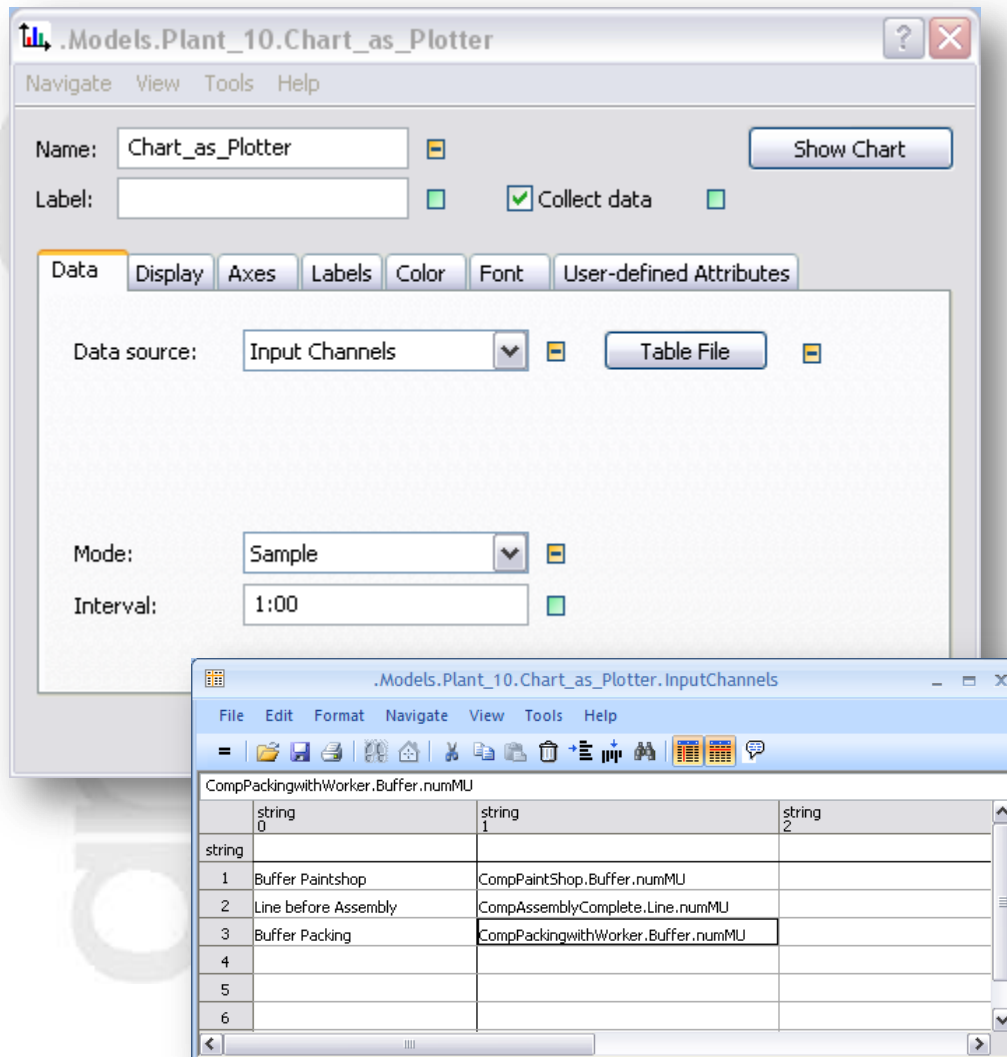
Jellemzők :

- Ikon: 
- Kapacitás: 0
- Aktív felhasználói felület objektum



A *Chart* értékek változását rögzíti és görbével ábrázolja.

Míg a *Display* egy változó aktuális értékét mutatja, addig a *Chart* több változó aktuális értékének mutatósára szolgál.

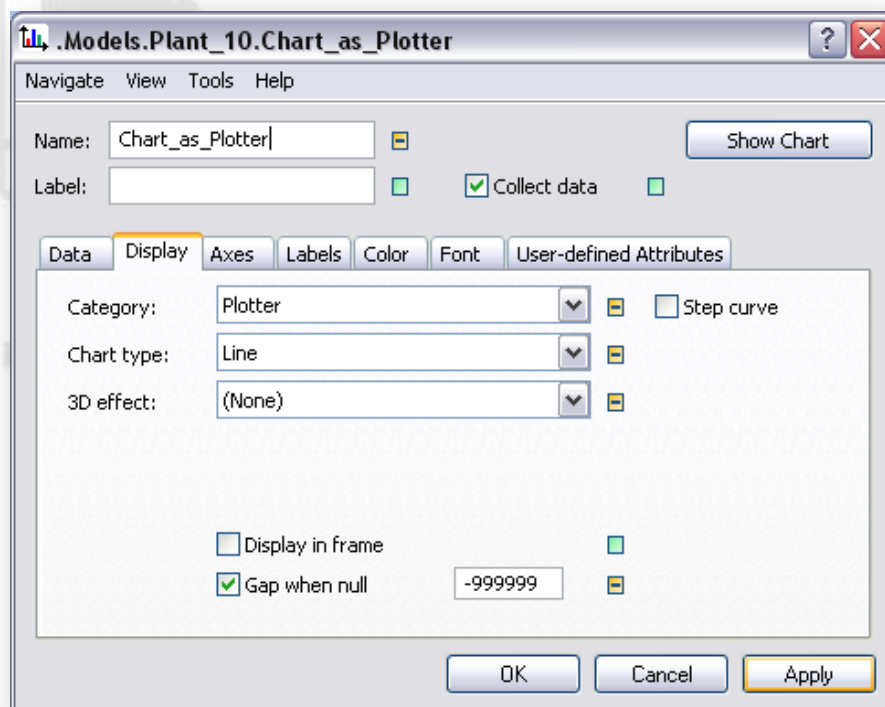


Data Source:

A table file-ban megadhatók az adatforrások az első oszlopban. A nulladik oszlopban használt értékek jelennek meg feliratként.

Mode:

Kiválasztható a **Plot**, a **Sample** vagy a **Watch** mód.



Category:

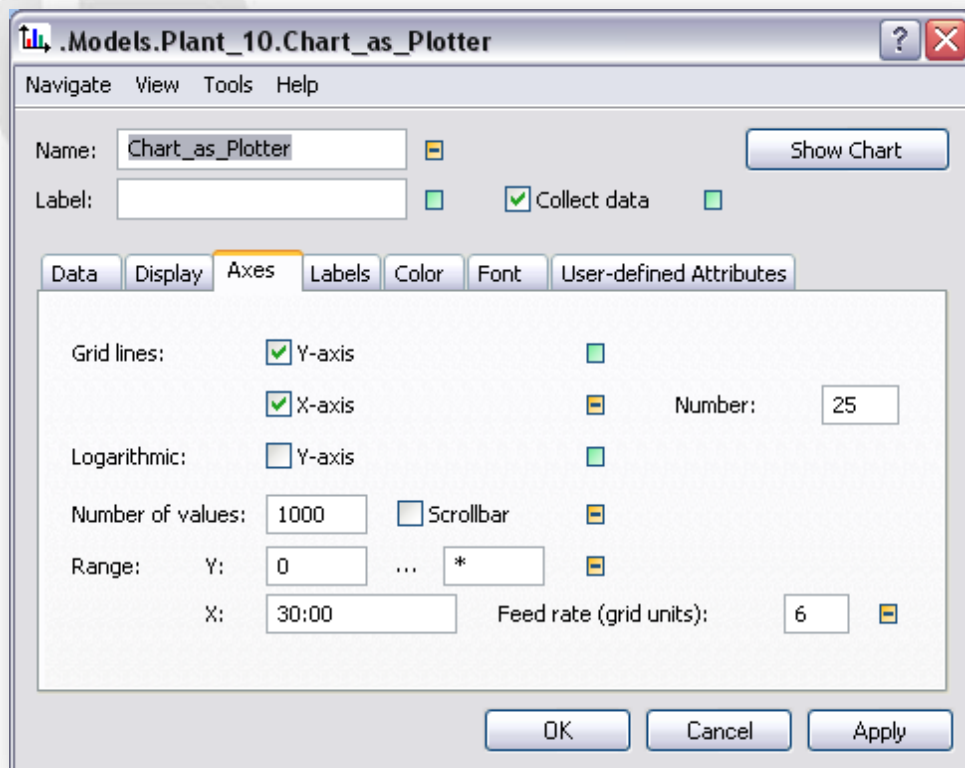
Legyen **Plotter** a grafikon kategóriája.

Chart Type:

Legyen **Line** a grafikon típusa

Data:

Legyen **in Column** az adatforrás.



Number of values:

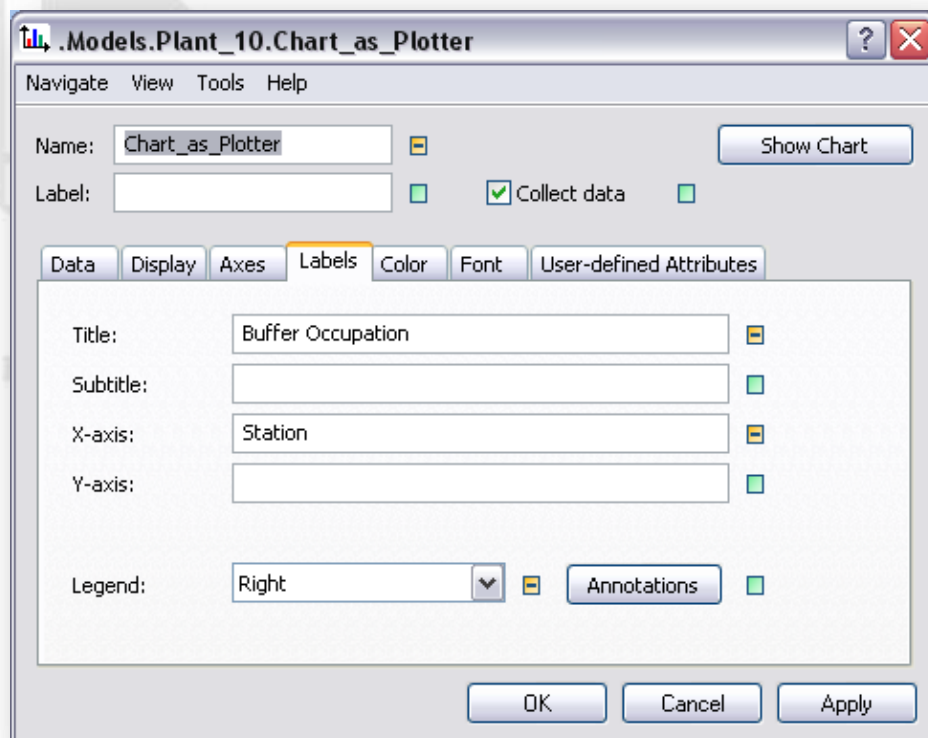
A kirajzoláshoz a grafikonban tárolt értékek száma.

Scrollbar:

Görgetősáv megjelenítése a grafikon számára, a tárolt értékek mentén lehet görgetni.

Range:

Az y tengely tartományának megadása. A maximális elemszám 100 lehet.

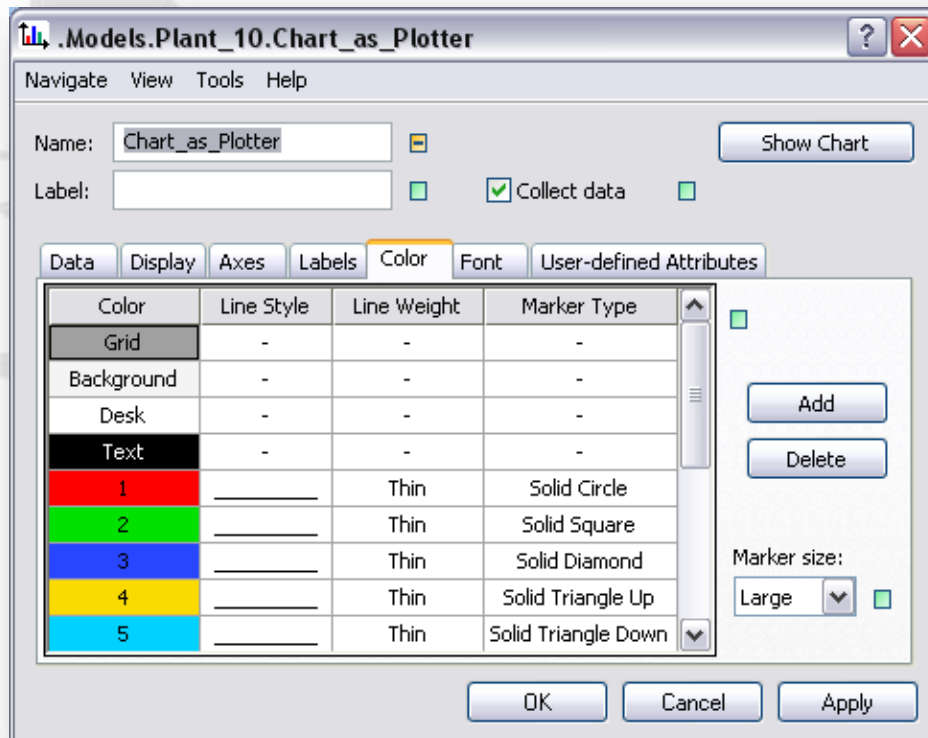


Title:

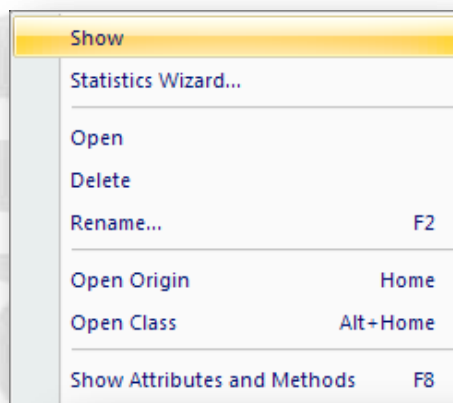
A grafikon címének megadása

Legend:

A felirat pozíciójának megadása
a lenyitható menüből kiválasztva.

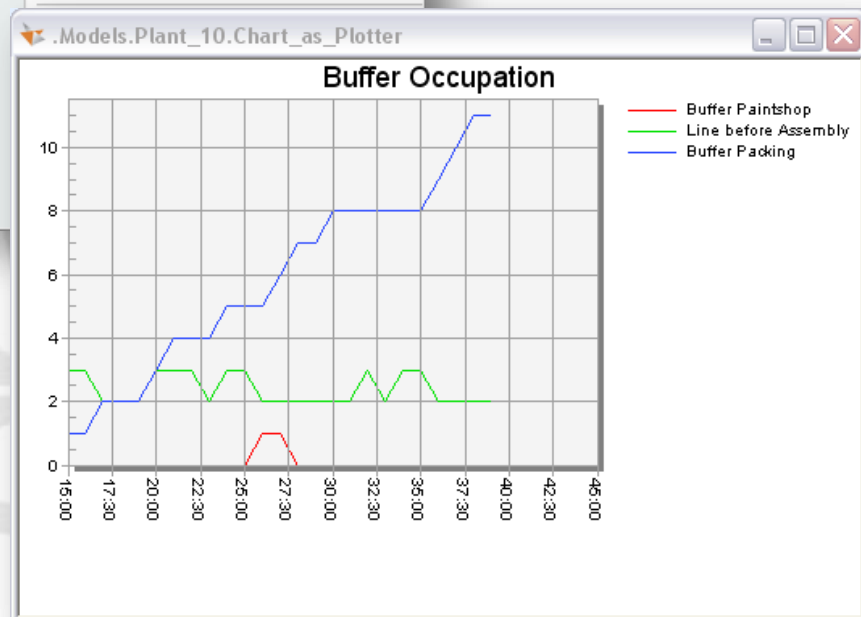


A rajzolósi vonalvastagság megadása az összes bemeneti csatornához a táblázatban.



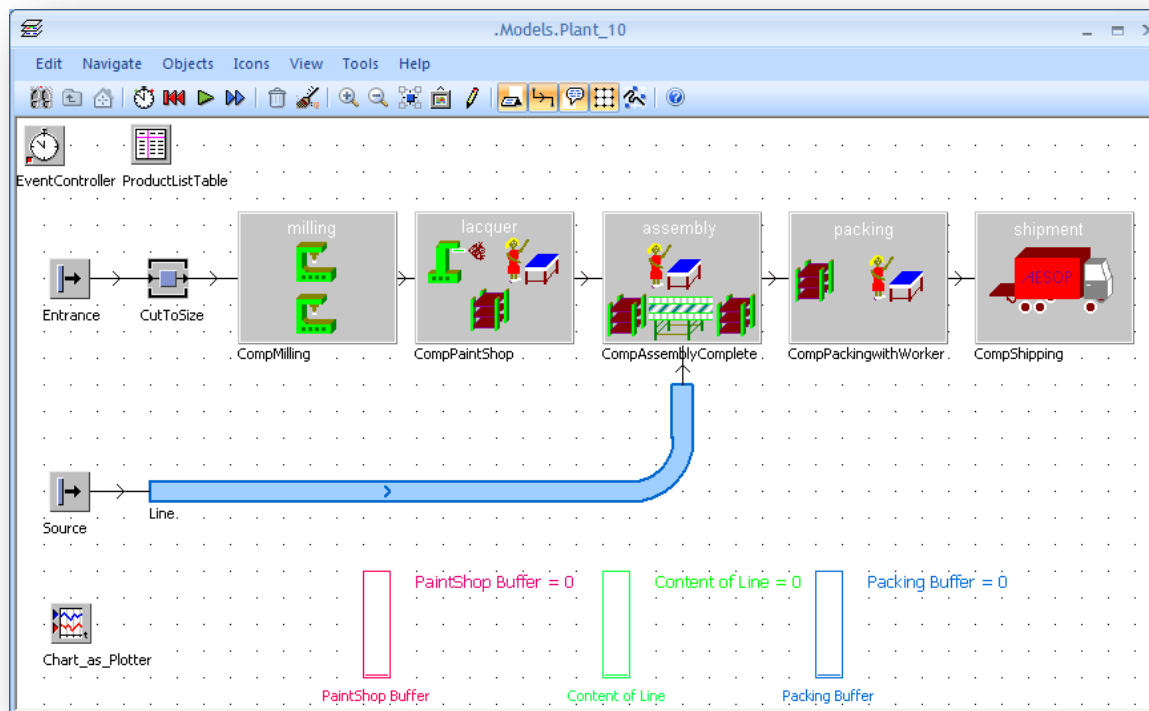
A megjelenítési ablak megnyitásához kattintson a Chart objektumra jobb egérgombbal és válassza ki a Show parancsot.

A *Chart* a megadott értékekkel jelenik meg.



Illesszen be néhány *Gauge* objektumot a *Frame*-be. A *Gauge* objektumok arra szolgálnak, hogy megjelenítsék az MU-k számát a komponensek pufferében a **Watch** módban oszlopgrafikonként, vagy szöveggént (**String**). Az objektum másolható és beilleszthető vágólapon keresztül, így csak a másolaton kell változtatni a beállításokat.

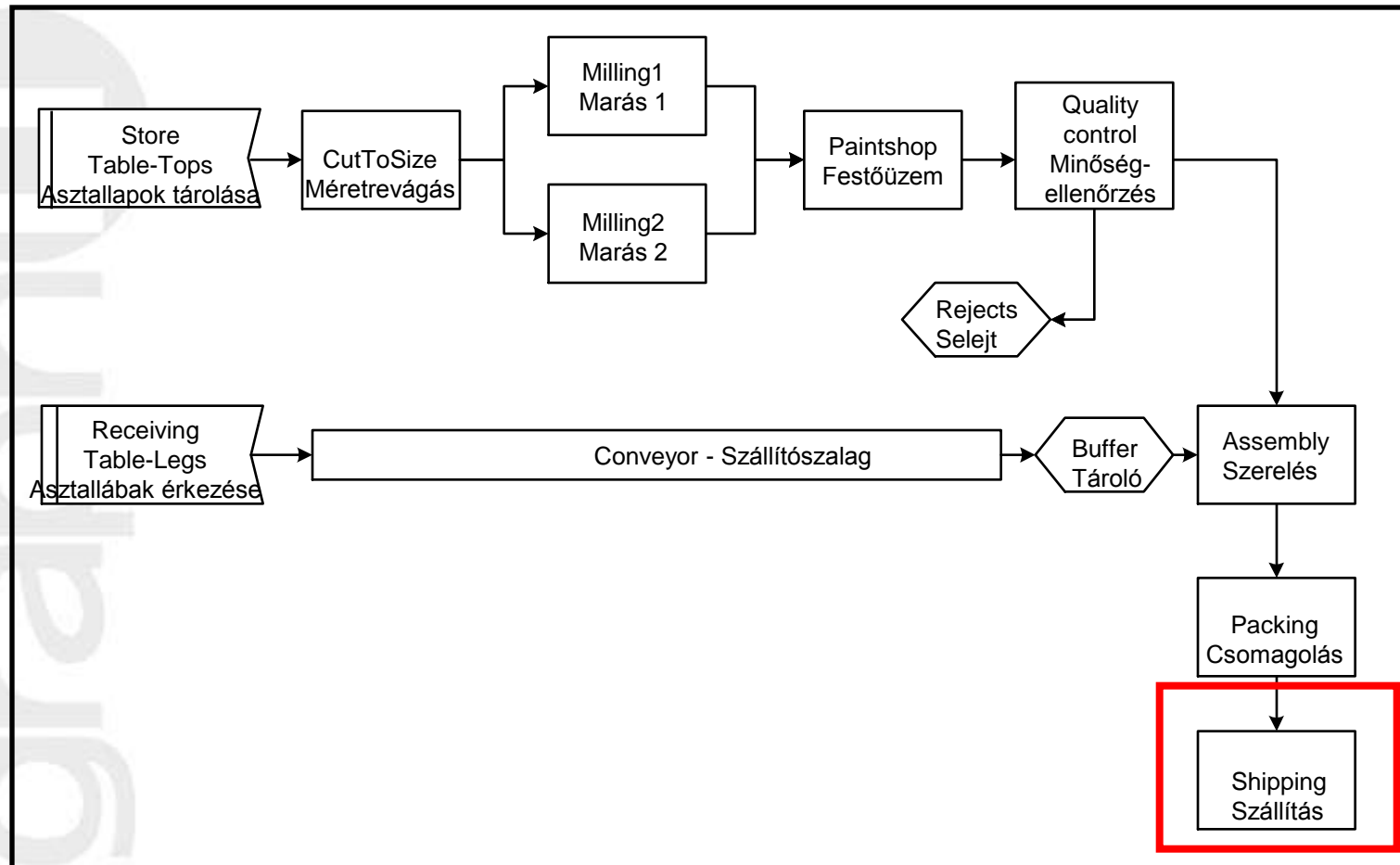
Illesszen be egy *Chart* objektumot a modellbe, és használja szintén az MU-k számának megjelenítésére **Plot** módban (**Y - Channels**).





2. fejezet

Metódusok statisztikai értékek
összegyűjtésére



<code><TableFile>.setCursor(1,1);</code>	a megadott cellába állítja be a táblázat kurzorát
<code><...>.find(`[1,1]..`[*,*],<value>);</code>	érték keresése a megadott tartományban
<code><...>.CursorX, <...>.CursorY;</code>	a kurzor beállítása az oszlophoz (<i>CursorX</i>) vagy a sorhoz (<i>CursorY</i>)
<code><...>.xDim; <...>.yDim;</code>	visszatér az utolsó oszloppal/sorral, amely adatot tartalmaz
<code><...>.sort(3,"down");</code>	egy vagy több oszlop sorba rendezése a táblázatban növekvő, vagy csökkenő sorrendben
<code><...>.meanValue(`[1,*]);</code>	a megadott tartományban lévő értékek átlagával tér vissza
<code><...>.delete(`[1,1]..`[*,*]);</code>	adott tartomány, vagy a teljes tartalom törlése
<code><local_variable>.create;</code>	tábla lokális változó készítése tartalom nélkül

<code><TableFile>.setDataType(1,"real")</code>	oszlop adattípusának beállítása
<code>< >.cutRow(1);</code>	sor kivágása
<code>< >.insertColumn(1);</code>	a megadott oszlop jobb oldalához új oszlop beszúrása
<code>< >.readFile("C:\\... .txt");</code>	fájladat betöltése egy Plant Simulation táblázatba
<code>< >.openfile("C:\\... .txt");</code>	a megadott fájl megnyitása
<code>< >.readFileRow;</code>	annak a fájl sornak a kiolvasása, ahol a kurzor található, és beillesztése a listába vagy táblázatba
<code>< >.FileCursor := <5>;</code>	a belső fájl kurzor beállítása a megadott sorba
<code>< >.closeFile;</code>	megnyitott fájl bezárása
<code>< >.writeFile("C:\\... .txt");</code>	Plant Simulation táblázat adatainak fájlba írása

Összefoglalás: Teljes statisztika - Complete Statistics Table

A teljes statisztika táblázat (complete statistics table) az összes objektum, anyagáram objektum és mozgó objektum statisztikai értékét gyűjti össze.

A `statistics` metódus egy objektumra visszatér az adott objektum statisztikai értékeivel: `<object>.statistics;`

Resource Statistics - Movable Resources

Material Flow Properties

Object	Number of Entries	Number of Exits	Minimum Contents	Maximum Contents	Empty	Relative Occupation without Interruptions	Relative Occupation with Interruptions
.MUs.TableTop:4	0	0	0	0	100.00%	0.00%	0.00%

A táblázatban 111 oszlopban láthatók a statisztikai értékek, amelyek oszlopindexekkel elérhetők.

A statisztikai adatok jobban áttekinthetők HTML jelentésként `view ->`

Statistics Report (F6)

A *statistics* metódus megnyitja a teljes statisztika táblázatot. Az alapértelmezett paraméter egy objektum, például az aktuális MU.

A teljes statisztika tábla megjeleníthető a képernyőn, táblázatba vagy fájlba írható:

A teljes statisztika táblázat különálló értékeihez azt először táblázatba kell írni, például egy lokális változóba.

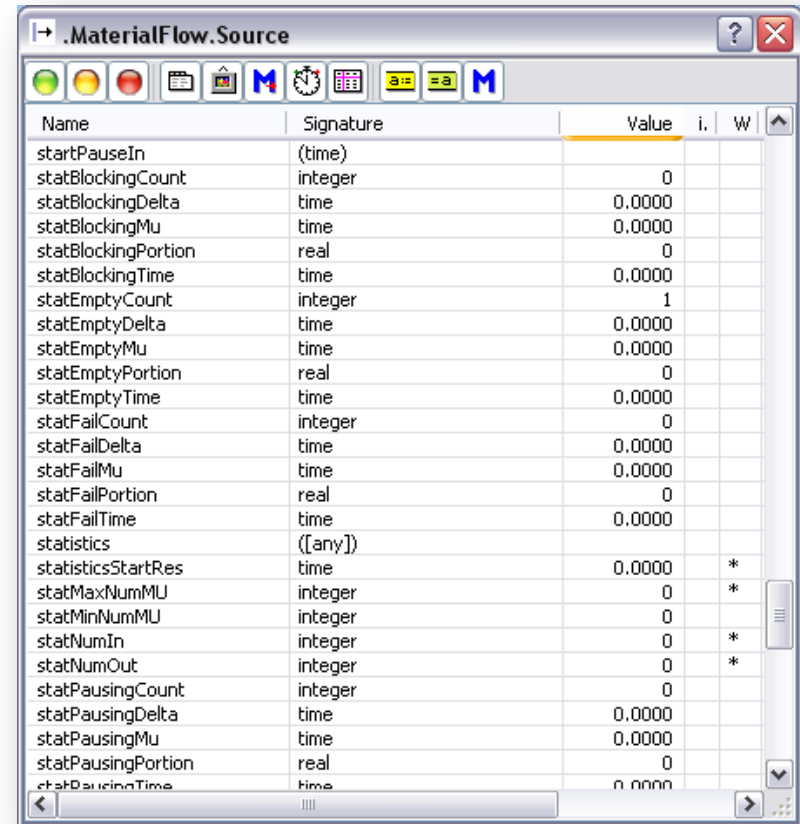
```
<objektum_neve>.statistics(<táblázat>);
```

- **Ha nincs paraméter:** Táblázat megjelenítése a képernyőn.
- **Táblázat argumentumként:** Az értékek kiírása a táblázatba, ami lehet egy *TableFile* objektum vagy egy lokális változó.
- **Szöveg argumentum:** Az értékek fájlba írása ezzel a névvel.

Számos módszer alkalmas anyagáram objektumokra összegyűjtött értékek lekérdezésére, mint például: *statNumIn*, *statNumOut*, stb.

<Változó> := <objektum>.statNumIn;

Az objektum összes attribútuma és módszere megjeleníthető az objektum kiválasztása után az F3 billentyű lenyomásával.



Name	Signature	Value	i.	W
startPauseIn	(time)			
statBlockingCount	integer	0		
statBlockingDelta	time	0.0000		
statBlockingMu	time	0.0000		
statBlockingPortion	real	0		
statBlockingTime	time	0.0000		
statEmptyCount	integer	1		
statEmptyDelta	time	0.0000		
statEmptyMu	time	0.0000		
statEmptyPortion	real	0		
statEmptyTime	time	0.0000		
statFailCount	integer	0		
statFailDelta	time	0.0000		
statFailMu	time	0.0000		
statFailPortion	real	0		
statFailTime	time	0.0000		
statistics	([any])			
statisticsStartRes	time	0.0000		*
statMaxNumMU	integer	0		*
statMinNumMU	integer	0		
statNumIn	integer	0		*
statNumOut	integer	0		*
statPausingCount	integer	0		
statPausingDelta	time	0.0000		
statPausingMu	time	0.0000		
statPausingPortion	real	0		
statPausingTime	time	0.0000		

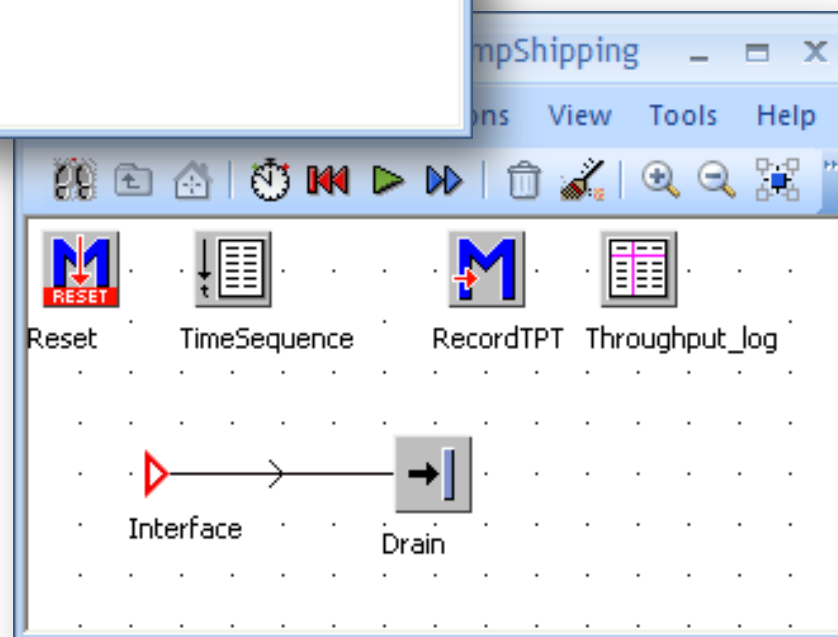
```

M .FacilityComponents.CompShipping.RecordTPT
File Edit Navigate Run Template View Tools Help
1: is
2:   NextRow : integer;
3: do
4:   NextRow := Throughput_log.ydim + 1;
5:   Throughput_log[1,NextRow] := @.id;
6:   Throughput_log[2,NextRow] := @.statAvgLifespan;
7: end;
    
```

Beleírja az MU azonosítóját az első oszlopba, és az első nem „foglalt” sorába a táblázatnak.

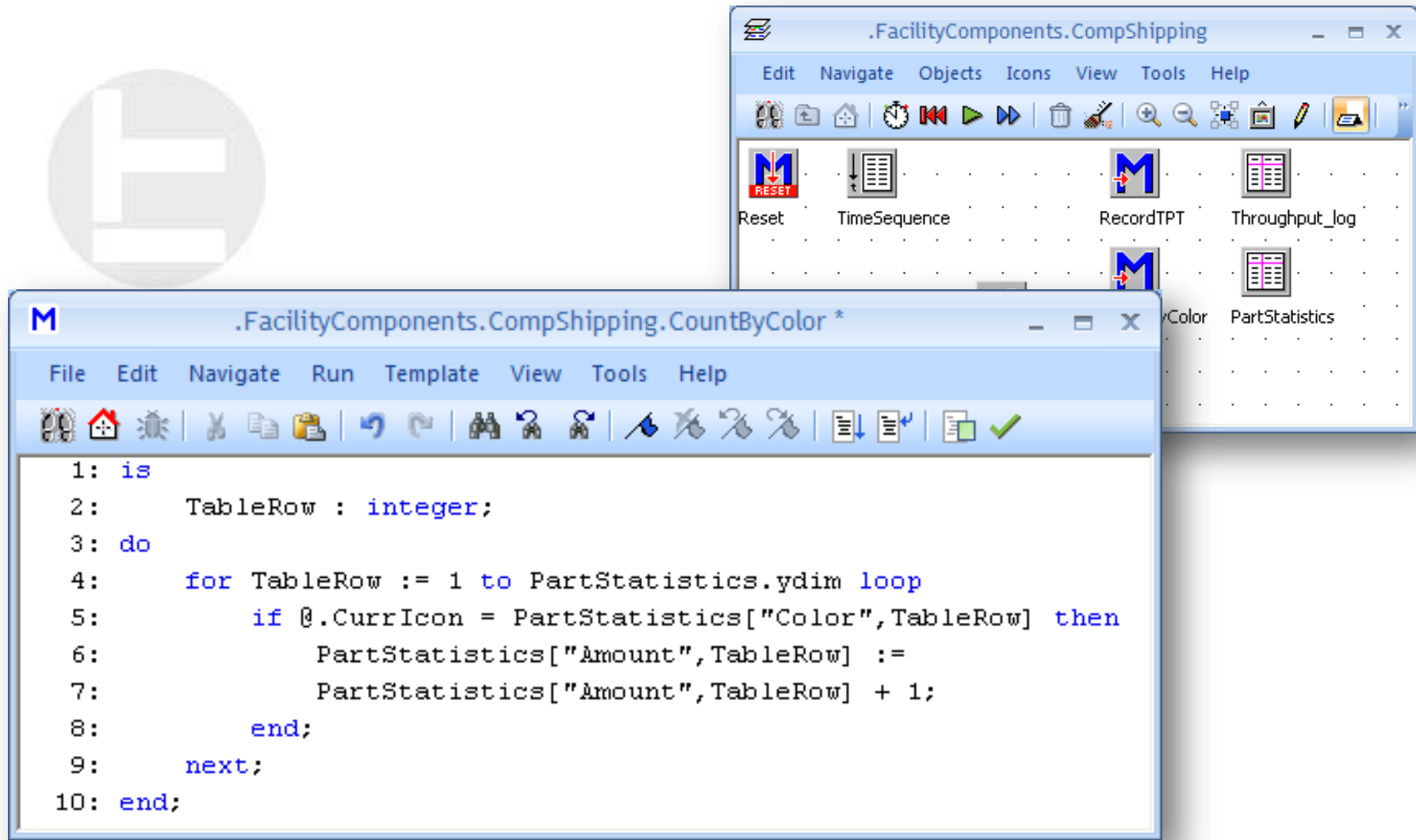
A névleges végighaladási időt beleírja a második oszlopba.

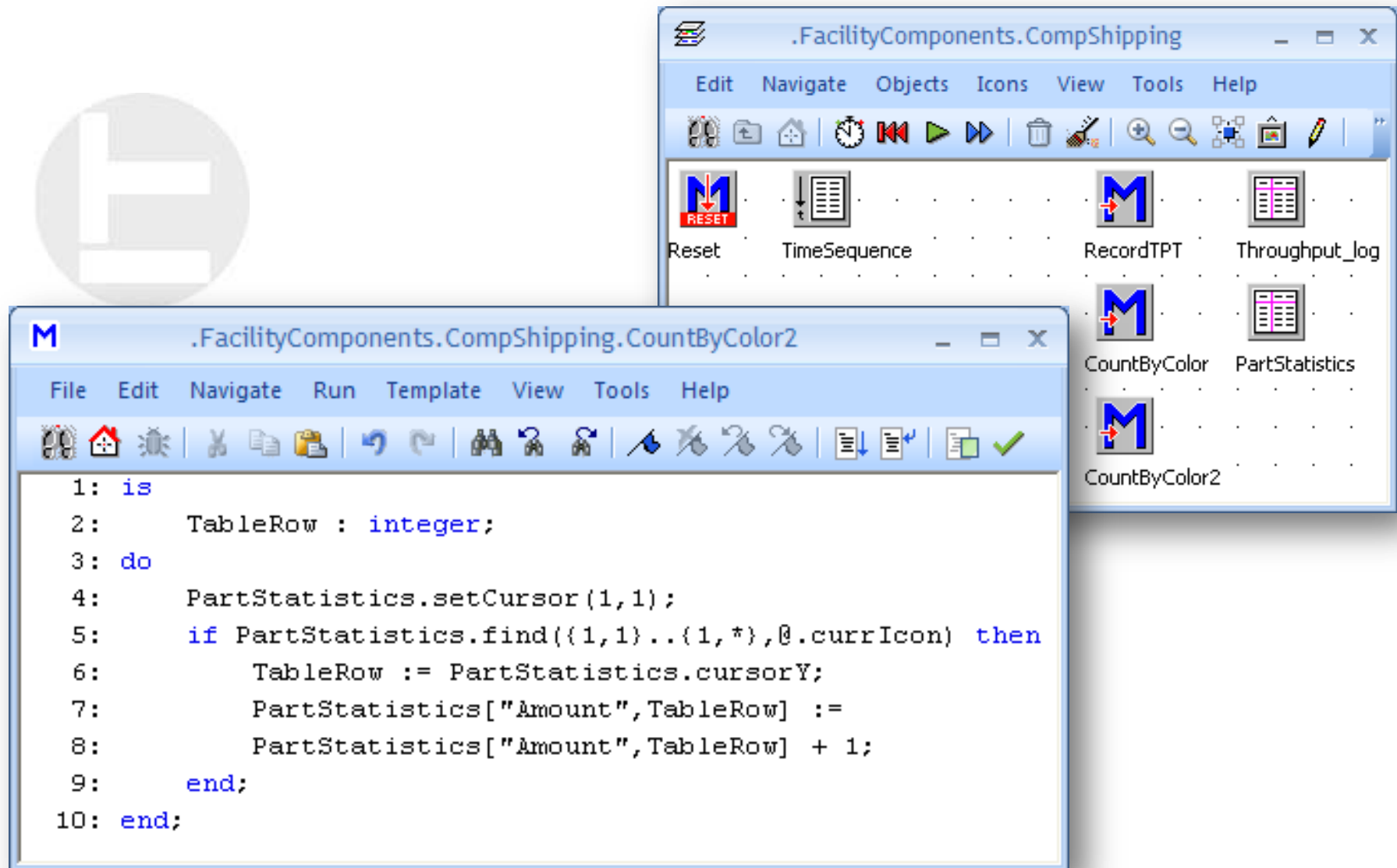
Az RecordTPT metódus a CompShipping Drain belépési kontrolljaként használható.



Számos esetben egy bizonyos termékről van szükség információra, ami egy attribútummal különböztethető meg a többitől.

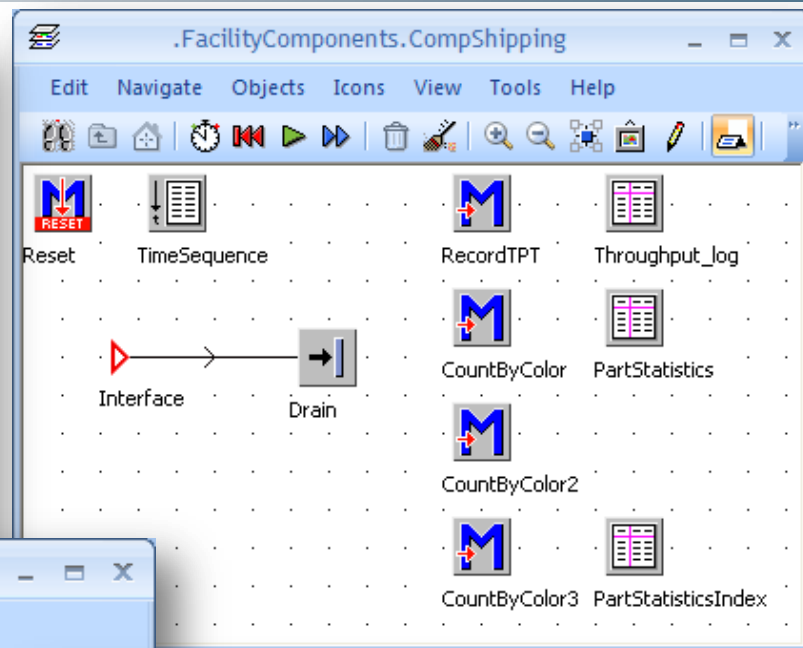
Ez a lecke megmutatja, hogyan lehet attribútum alapján adatokat összegyűjteni, esetünkben az asztal színe alapján.





.FacilityComponents.CompShipping.PartStatist...

	string 1	integer 2
string	Color	Amount
1	blue	
2	red	
3	yellow	
4	brown	
5	green	



.FacilityComponents.CompShipping.CountByColor3

```

1: is
2:   PartColor : string;
3: do
4:   PartColor := @.currIcon;
5:   PartStatisticsIndex["Amount",PartColor] :=
6:   PartStatisticsIndex["Amount",PartColor] + 1;
7: end;
    
```

.FacilityComponents.CompShipping.Reset

```

1: is
2: do
3:   timesequence.delete;
4:   Throughput_log.delete;
5:   PartStatistics.delete;
6:   PartStatisticsIndex.delete;
7: end;
    
```



3. fejezet

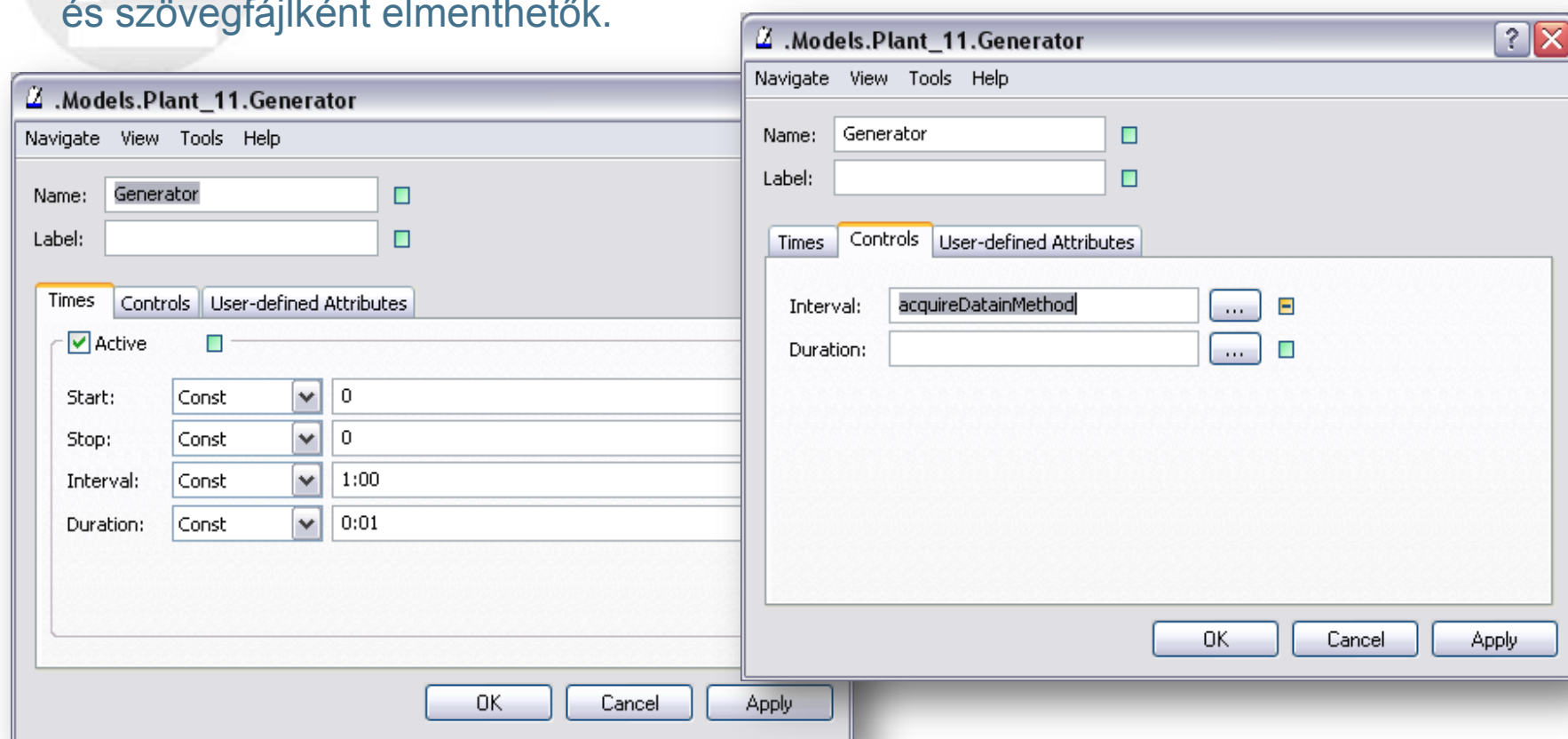
Plant Simulation táblázat mentése
szövegfájlként

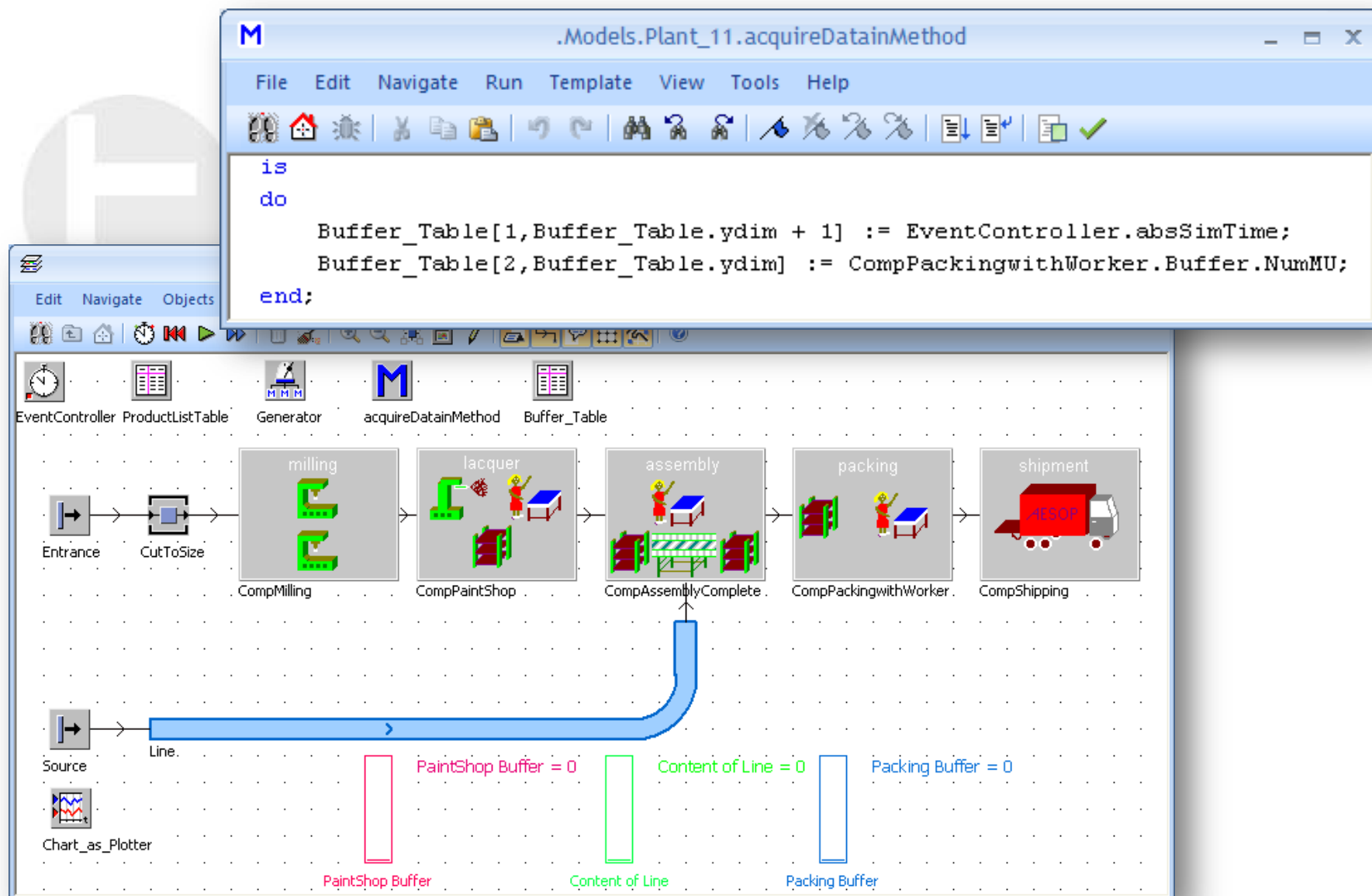
Egy objektum, például egy Plant Simulation *TableFile* adatai elmenthetők szövegfájlba. Így az összegyűjtött adatok más alkalmazásban is felhasználhatók.

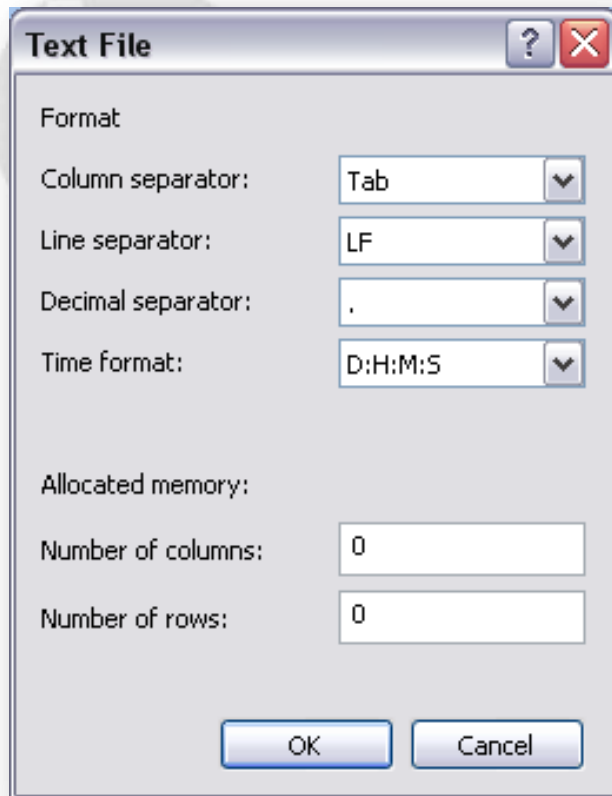
Példa: Plotter

Ha egy *Plotter* által összegyűjtött értékeket szeretnénk használni egy MS Excel táblázatban, illesszünk be egy metódust a *Plotter* helyett a *Frame*-be, és írjuk az értékeket a táblázatba. Ez után ez a Plant Simulation táblázat elmenthető szövegfájlként, ami az MS Excelben megnyitható és tovább kezelhető.

Ezt a metódust **Interval** kontrolljaként készítettük el egy *Generator* objektumnak. Minden percben rögzíti a *Buffer* - ben az MU-k számát, és az *EventController*-ből az aktuális időpillanatot . Ez után az értékek egy Plant Simulation táblázatba írhatók, és szövegfájlként elmenthetők.







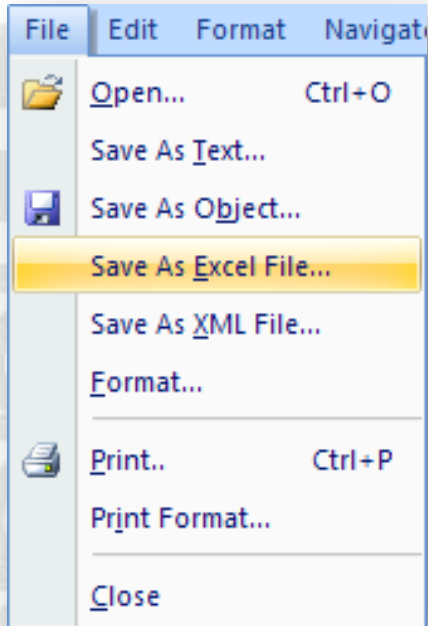
Mielőtt egy Plant Simulation táblázatot szövegfájlba mentene:

Válassza ki a File > Format menüpontot.

Győződjön meg arról, hogy a **Decimal separator** (decimális elválasztó) **period (pont)**.

Ha lehetséges, akkor legyen másodperc az időformátum.

Ezzel elkerülhető, hogy az MS Excel hibásan olvassa be az adatokat.



A Plant Simulation táblázat szövegfájlként mentéséhez:

1. Válassza ki a File > Save As Text menüpontot

Adja meg a fájl nevét és válassza ki a mappát.

2. Használja a `writeFile` metódust.

Ez felülírja az esetleg meglévő fájlt.

Példa:

```
TableFile.writeFile("C:\\<folder>\\<filename>.txt");
```



4. fejezet

Szövegfájl betöltése Plant Simulation
táblázatba

Szövegfájl betöltése a Plant Simulation rendszerbe:

```
<TableFile>.readFile("D:\\Files\\TableX.txt");  
<TableFile>.openFile("D:\\Files\\TableX.txt");  
<TableFile>.readFileRow;
```

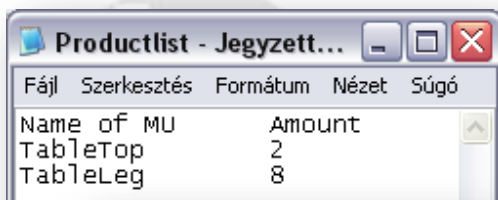
Ha a szövegfájlt sorról sorra kívánja betölteni, először meg kell nyitni a fájlt. A fájlnek van egy belső kurzora.

A táblázatnak, amelybe az adatokat be kívánja tölteni, léteznie kell a Frame - en belül, vagy létre kell azt hozni az alábbi módon:

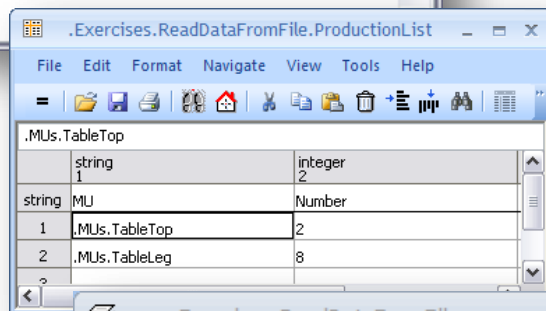
```
.InformationFlow.TableFile.createObject(current,10,30);
```

Példa gyártási adatok a *Source* számára, az alkatrészek neveivel és darabszámukkal.

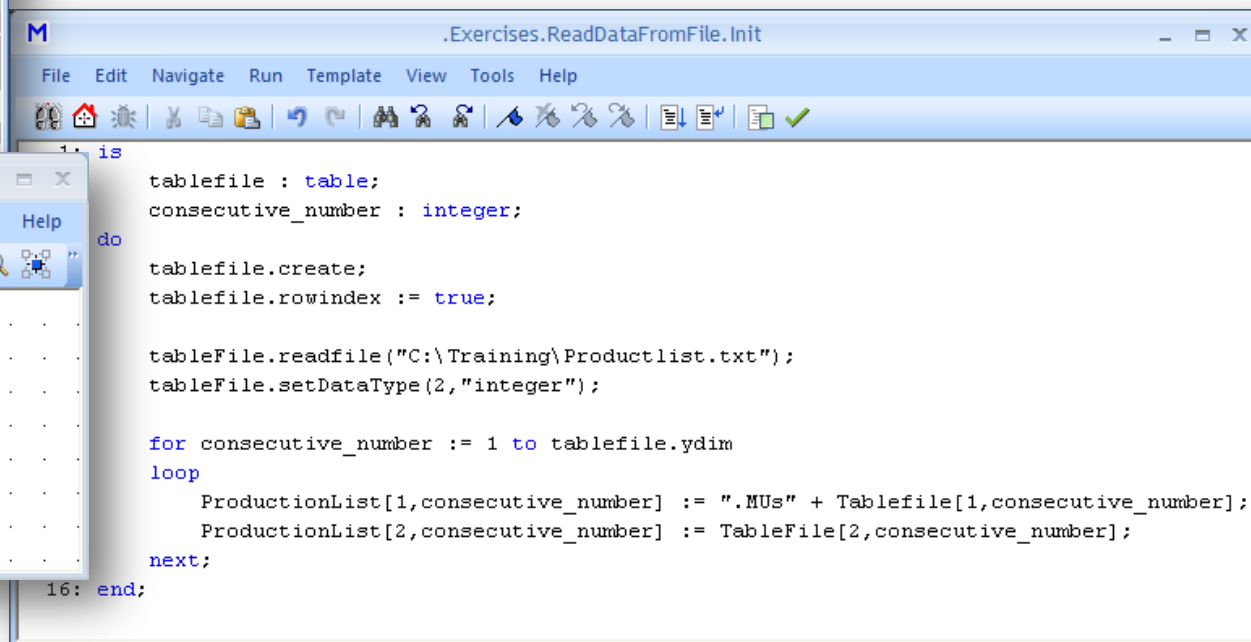
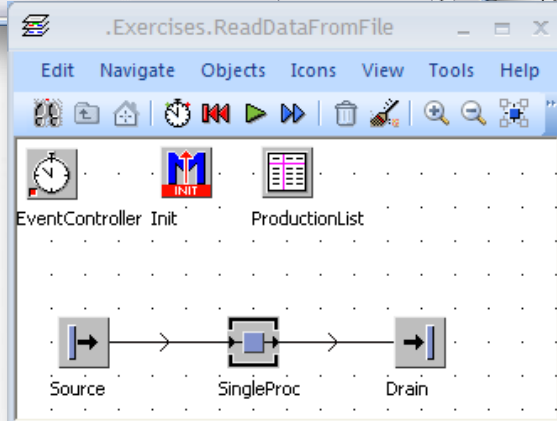
A betöltendő szövegfájlban az oszlopokat tabulátornak vagy pontosvesszőnek kell elválasztani. (összehasonlítható a *File > Format* beállításával a Plant Simulation táblázatban).



Fájl	Szerkesztés	Formátum	Nézet	Súgó
Name of MU	Amount			
TableTop	2			
TableLeg	8			



File	Edit	Format	Navigate	View	Tools	Help
.MUs.TableTop						
string	1	integer	2			
string	MU	Number				
1	.MUs.TableTop	2				
2	.MUs.TableLeg	8				



```
1: is
tablefile : table;
consecutive_number : integer;
do
tablefile.create;
tablefile.rowindex := true;

tableFile.readFile("C:\Training\Productlist.txt");
tableFile.setDataType(2,"integer");

for consecutive_number := 1 to tablefile.ydim
loop
ProductionList[1,consecutive_number] := ".MUs" + Tablefile[1,consecutive_number];
ProductionList[2,consecutive_number] := TableFile[2,consecutive_number];
next;
16: end;
```



5. fejezet

Interfész más alkalmazások felé

A Plant Simulation három típusú szabványos interfészt nyújt más alkalmazások felé:

Fájl interfészek

- DDE (Dynamic Data Exchange)
- ODBC (Open Database Connectivity)
- SQL/Oracle (SQL – Structured Query Language)
- Socket
- XML (Extensible Markup Language)
- SDX (Simulation Data Exchange)

Kommunikációs interfészek (nem része ennek a tanfolyamnak)

- C Interface
- Active/X

Kontrol interfészek (nem része ennek a tanfolyamnak)

- COM RemoteControl Interface
- HTML

A *DDE Interface* a Plant Simulationbe integrálva lehetővé teszi a Windows DDE rendszer használatát. A Plant Simulation lehet DDE szerver vagy kliens.

A Plant Simulation, mint DDE kliens: Lehetővé teszi tetszőleges adat átküldését a szimulációs modellből más alkalmazásokba, mint például: Microsoft Access, vagy MS Excel vagy MS Word.

A Plant Simulation, mint DDE - szerver: Más alkalmazások hozzá tudnak férni a Változó (Variable) típusú Plant Simulation objektumokhoz. Például MS Excelből:

```
"='eM-Plant' | data! ' .Models.Frame.Variable ' "
```

A megcímzett alkalmazásnak és a megfelelő fájlnak meg kell nyitva lenni.

A *DDE* interfész nem igényel külön licencet.

Lépések:

1. Először létre kell hozni a kapcsolatot egy alkalmazáshoz – ez egy csatornát nyit meg. Visszaadja a csatorna számát, például egy globális változóval, itt most ennek a neve `Channel`. A csatornaszám mindig nagyobb, mint nulla.

- `Channel := ddeConnect("Excel", "Table1");`

2. Ez után át kell küldeni az adatot a **ddePoke** metódussal - paraméterei a megnyitott csatorna száma, a mező neve, és a küldendő adat, valamint egy időtúllépés érték milliszekundumokban:

- `ddePoke(Channel, <field>, to_str(Drain.numIn), 1000);`

3. A szimuláció vagy az adatküldés befejeztével be kell zárni a megnyitott csatornát:

```
ddeDisconnect(Channel);
```

```
Channel := 0;
```



Az MS Excel táblázatnak nyitva kell lennie!

```
is
    field1, field2 : string;
do
    tab.create;
    field1 := "R" + num_to_str(row_counter) + "C1";
    field2 := "R" + num_to_str(row_counter) + "C1";
    ddePoke(channel, field1, to_str (@.ID), 1000);
    ddePoke(channel, field1, to_str (@.statAvrgLifeSpan), 1000);
    row_counter := row_counter+1;
end;
```

-- **field**: a következő MS Excel sor megcímzése "R1C1" Row1Column1.


-- **row_counter**: globális változó.

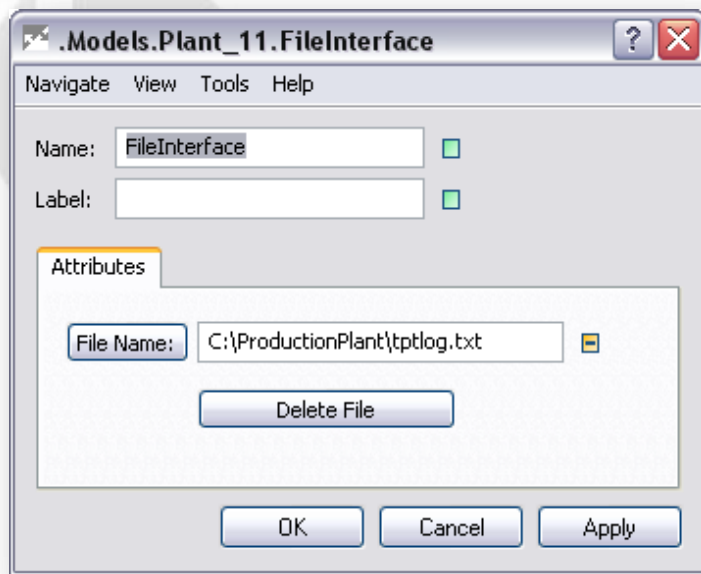
ODBC: Külön licencet igénylő kiegészítés tetszőleges ODBC adatbázishoz történő hozzáférésre, pl: *MS Excel, dBase, szövegfájlok, Oracle, Informix*, stb.

SQL/Oracle: Külön licencet igénylő kiegészítés. Lehetőséget nyújt Oracle 8.0/8.1 adatbázisokhoz történő hozzáférésre az SQL*Net interfészen keresztül.

Socket: Külön licencet igénylő kiegészítés. Két folyamat TCP/IP protokoll alapú kommunikációját biztosítja kétirányú adatcserével.

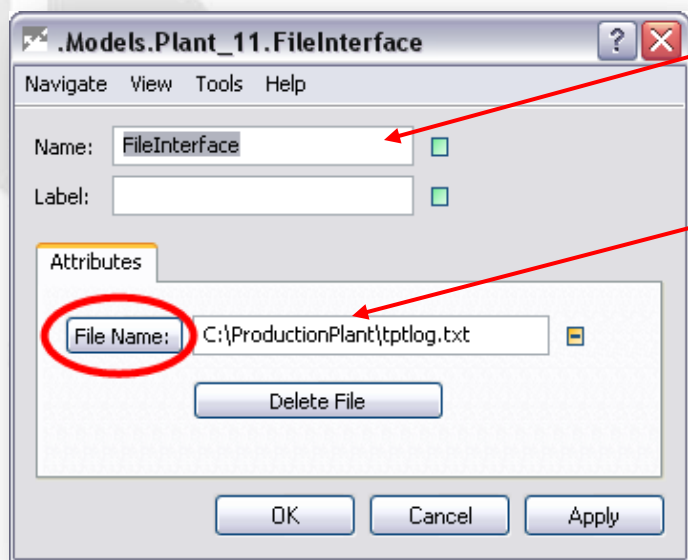
Jellemzők:

- Ikon: 
- Kapacitás: 0
- Aktív információáram objektum



A *FileInterface* más programok által nyújtott adatok kezelésére biztosít egyszerű felületet.

Használható Plant Simulation statisztikai, szimulációs adatok mentésére külső fájlba, illetve szövegfájlok Plant Simulationbe történő beolvasására. Nem szükséges a használata előtt táblázatba írni az adatokat.



Adja meg a nevet, amely az objektumra jellemző (**Name**).

Adja meg a fájlnevet (**File name**), melyet létre kíván hozni.

Vagy a **File name** gombra kattintva válasszon ki egy mappát és egy fájlt.

Új adat érkezésekor a FileInterface az új adatot a meglévő adat végére írja.

A fájl tartalmának törlése a **Delete File** gombbal vagy a következő metódussal lehetséges:

```
<name_of_FileInterface>.remove;
```

```
<name_of_FileInterface>.writeln(<data>," ; ", "string");
```

Ez a metódus egy sort ír ki a fájlba. Az oszlopok között elválasztót kell használni. Így könnyebbé válik az adat feldolgozása más alkalmazásokban.

```
< name_of_FileInterface>.readLn;
```

Ez a metódus egy sort olvas a fájlból, és megnöveli a belső fájl kurzort eggyel.

```
< name_of_FileInterface>.eof;
```

Ez a metódus a kurzorral a fájl végéhez ugrik.

Példa:

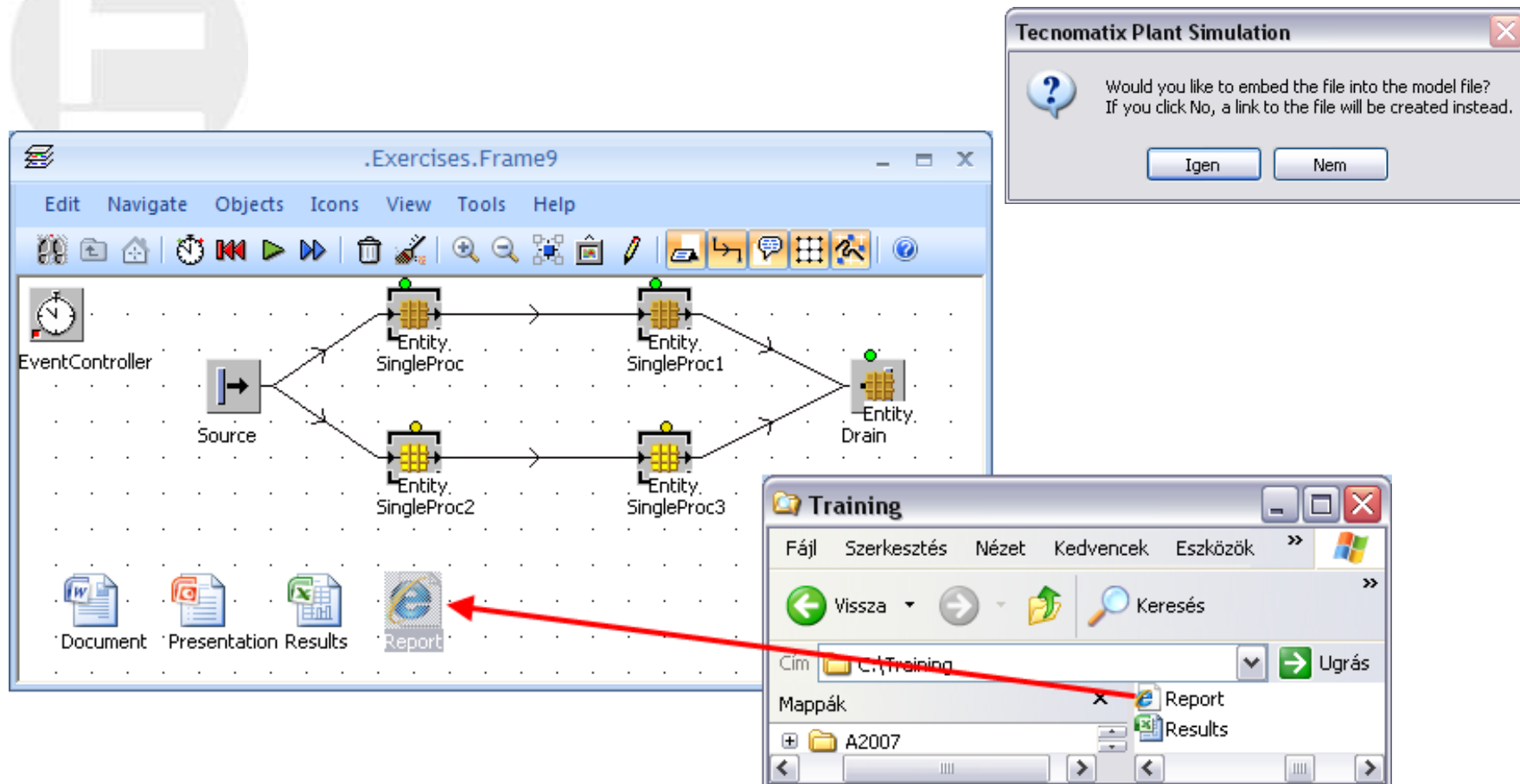
```
while not FileInterface.eof loop str := FileInterface.readLn...
```

→ Ez után az érték táblázatba írható, vagy helyi változóként használható.

```
FileInterface.remove;
```

→ Törli a *FileInterface* által használt fájlt.

Egy fájlt a beágyazáshoz a Windows Intézőből húzd és ejtsd módszerrel rá kell dobni a *Frame* - re, amelybe be kívánjuk ágyazni.

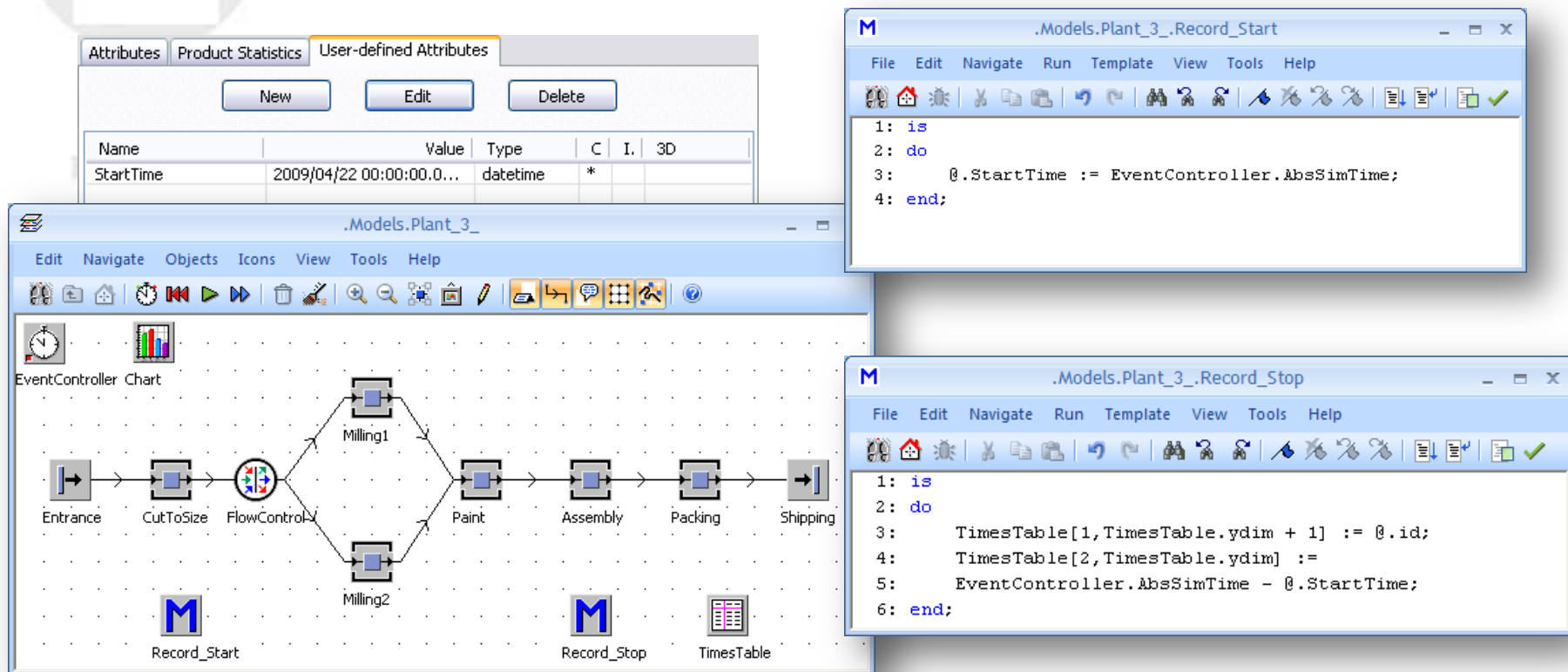


Függelék

Példák

Példa: MU kimenet rögzítése a gyár egy szakaszán

Az `EventController.absSimTime` az `EventController` aktuális idejével tér vissza. Az MU - n belül egy egyedi jellemző definiálható, ami ezt az időértéket használja.



Példa: Az MU-k érkezési ideje, mint az alkatrészek szállítási ideje.

- Adja meg a végidőt (**End**) a szimulációra az *EventController* - ben.
- Írja az MU - k érkezési idejét egy táblába. Mentse el a táblát szövegfájlként a következő metódussal:

```
.writeFile("D:\\...").
```

- Várja meg, amíg a szimuláció befejeződik. Indítsa az **EventController** - t a modellben az **Endsim** metódussal:

```
<Path>.EventController.reset;
```

```
<Path>.EventController.init;
```

```
<Path>.EventController.start;
```

- A meghívott modell **Init** metódusa a szövegfájlt olvassa be:

```
<tablefile>.readFile("D:\\...D).
```



Köszönöm a figyelmet!