




ЛЕКЦИЯ 6

КОНСТРУКЦИЯ НА МОДЕЛА

-  **Моделът Biotope_1**
-  **Експериментиране с моделът Biotope_1**
-  **Версия и състояние на модела**

ВЪВЕДЕНИЕ

Работата със симулационни модели може в общност да се раздели в следните три основни области:

- Конструирание на модела;**
- Експериментиране с модела;**
- Обработка на резултатите.**

Моделът Biotore_1

Ще разгледаме описанието на модела Biotore_1 в симулационната система Simplex3.

Моделът изучава отношенията между популацията на зайци (жертви) и лисици (хищници). Наблюденията показват, че популациите на зайците и на лисиците варират.

Когато зайците се увеличават бързо, наличната храна за лисиците расте, така че след известно време техният брой също нараства. Обратно, намаляване популацията на зайците води до намаляване на броя на лисиците. Малък брой на лисиците означава добри условия на живеене за зайците.

Моделът Biotope_1

Раждаемостта на зайците се дължи на естествените превишавания на смъртността, тогава, на отсъствието на лисици популацията на зайците ще нарасне, според следното диференциално уравнение:

$$Nare' = a * Nare$$

Това означава, че броят на допълнителните зайци за единица време е пропорционален на текущия брой.

В допълнение на естествената смъртност, размера на смъртността на зайците зависи от срещите между зайци и лисици.

$$c = (\text{Брой срещи}) / (1 \text{ единица време (ЕВ)} * 1 \text{ Заек} * 1 \text{ Лисица})$$

Броят на зайците и лисиците за една единица време (ЕВ) е “с”.

Моделът Biotope_1

Следва, че:

За даден брой зайци и лисици, общия брой срещи за единица време е даден чрез произведението $c * \text{Hare} * \text{Fox}$.

Така наречените брой жертви BFacH и BFacF определя как броя на зайците и лисиците се променя като резултат от срещите. Ако е прието, че един заек е изяден, когато протече среща, тогава задаваме $\text{FacH} = 1$. Ако само един заек е изяден на всеки две срещи, тогава ще зададем $\text{BFacH} = 0.5$. Ако на лисицата е нужно да изяде четири заека, за да оцелее, тогава фактора на жертвата BFacF ще има стойност 0.25.

$\text{BFacH} * c * \text{Hare} * \text{Fox}$ е броя на зайците, които са намалели от популацията за единица време като резултат от изяждането им от лисиците.

Моделът Biotope_1

Подобно, факторът $BFacF * c * Hare * Fox$ дава увеличение на броя лисици като резултат от срещите.

$$Hare' = a * Hare - BFacH * c * Hare * Fox$$

$$Fox' = -b * Fox + BFacF * c * Hare * Fox$$

Ако приемем, че на всяка среща броя на зайците намалява с един, например $BFacH = 1$. Ако десет заека са нужни, за да поддържат една лисица жива, то $BFacF = 0.1$.

Приемайки $BFacH = BFacF$ се получава форма на уравнения, познати като уравнения на Lotka – Volterra:

Моделът Biotope_1

$$\text{Hare}' = a * \text{Hare} - d * \text{Hare} * \text{Fox}$$

$$\text{Fox}' = -b * \text{Fox} + d * \text{Hare} * \text{Fox}$$

Мерните единици са много важни при
конструкцията на модела.

Моделът Biotope_1

```
1  BASIC COMPONENT Biotope_1

2  USE OF UNITS
3    UNIT[NumH] = BASIS
4    UNIT[NumF] = BASIS
5    TIMEUNIT = [a]

6  DECLARATION OF ELEMENTS

7    CONSTANTS
8    a (REAL[1/a]) := 1.75 [1/a],
9    b (REAL[1/a]) := 1.25 [1/a],
10   c (REAL[1/(NumH*NumF*a)])
      := 0.0375 [1/(NumH*NumF*a)],
11   BFacH (REAL[NumH]) := 1.0 [NumH],
12   BFacF (REAL[NumF]) := 0.1 [NumF]

13   STATE VARIABLES
14   CONTINUOUS
15   Hare (REAL[NumH]) := 400 [NumH],
16   Fox  (REAL[NumF]) := 37 [NumF]

17  DYNAMIC BEHAVIOUR
18  DIFFERENTIAL EQUATIONS
19  Hare' := a*Hare - BFacH*c*Hare*Fox;
20  Fox'  := -b*Fox + BFacF*c*Hare*Fox;
21  END

22 END OF Biotope_1
```

Моделът Biotope_1 в средата Simplex3

Моделът Biotope_1

В моделът Biotope_1, са представени две нови основни единици за броя на зайци и лисици. В описанието на този модел, времето T е прието да бъде в единици години.

Константи са:

$$a=1.75 \quad [1/a]$$

$$b=1.25 \quad [1/a]$$

$$c=0.0375 \quad [1(\text{NumH} * \text{NumF} * a)]$$

$$\text{BFacH} = 1.0 \quad [\text{NumH}]$$

$$\text{BFacF} = 0.1 \quad [\text{NumF}]$$

Начални условия са:

$$\text{Hare}(0) = 400 \quad [\text{NumH}]$$

$$\text{Fox}(0) = 37 \quad [\text{NumF}]$$

Експериментиране с моделът Biotore_1

Моделът Biotore_1 показва спецификацията на модела, който е конструиран.

Моделът е много прост – състои се само от една компонента.

Експериментиране с моделът Viotope_1

За да е възможно да се работи с компонента в нейната текуща версия, компонентата трябва да е присъединена към модел.

Експериментиране с моделът Biotope_1

Създаване на експерименти и изпълнения.

Сега могат да се извършат различни експерименти с моделите в каталога Models.

Избира се каталога Experiments и се избира командата New experiment... Ако е необходимо експериментът, може да се преименува.

Един експеримент обикновено се състои от различни симулационни изпълнения. Всяко изпълнение може да се свърже със свой собствен модел от каталога Models. По този начин е възможна работата с различни модели по време на един експеримент и директно да се сравнят резултатите от всеки.

Експериментиране с моделът Biotope_1

Може да се види, че изпълнението Run1 съдържа прекъсвания, контролни параметри, наблюдения, симулационни резултати и протоколи.

Симулационните резултати към тази точка на прекъсване може да се анализират и покажат. Могат да се направят промени на променливите на модела.

Показва вътрешната структура на изпълнение с прекъсвания.

Симулационното изпълнение започва при $T = 0$ с Break0. Симулацията в случая продължава до $T = 10$. Всякакви промени на един или повече моделни променливи се запазват във файла за промени Chg. Основавайки се на състоянието описано чрез End и на промените, съхранявани в Chg, е изчислено новото начално състояние, с което започва следващия сегмент на изпълнение.

Експериментиране с моделът Viotope_1

Контролните параметри контролират как протича изпълнението на симулацията. Те могат да се видят и модифицират в каталога Control parameters.

Наблюдателите предлагат възможността за записване стойностите на избраните променливи на модел по време на изпълнение на симулация. Те могат да бъдат анализирани и показани по-късно.

Експериментиране с моделът Biotope_1

Забележки:

Контролните параметри и наблюдатели са валидни за цялата продължителност на изпълнение и те могат да бъдат създадени или модифицирани само в началото на изпълнението – Break0.

Създадени са два различни наблюдателя, наречени Obs1 и Obs2. Тези наблюдатели следят променливите на състояния Hare, Fox, и Hare', Fox'.

Видът на наблюдателя може да се дефинира, например Complete timeseries за записване на пълни динамични редове.

Експериментиране с моделът Biotope_1

Специфицират се следните атрибути:

- *Име на наблюдателя*

Името на наблюдателя може да се избере свободно.

- *Край*

Посочват се времената за започване и край на записвания интервал.

- *Стъпка*

При малки размери на стъпката се забавя симулацията. При големи размери на стъпката се достига до грубо и неточно представяне на резултатите. В примера размера на стъпката е 0.1.

Експериментиране с моделът Biotope_1

Наблюдателят Obs2 се създава по подобен на Obs1 начин със същите атрибути за Hare' и Fox'.

Когато в каталога Observer е избран прозорецът на съдържанието се показва всичката информация съответна на двата наблюдателя.

Експериментиране с моделът Biotope_1

Забележка:

Ако не се създадат никакви наблюдатели не се записват данни и накрая е налично само крайното състояние End. В този случай поведението на променливите, зависещо от времето, не може да се визуализира.

Експериментиране с моделът Biotope_1

Каталога Simulation results показва
динамични редове записани от
наблюдателя.

Експериментиране с моделът Biotope_1

Стандартно графично на симулационните резултати може да се получи чрез избирането на желаните динамични редове и натискане на бутона за показване в лентата с инструменти.

Получаваме първите графични резултати чрез избиране на динамичните редове Obs1 # Biotope_1 / Hare и Obs1 # Biotope_1 / Fox.

Експериментиране с моделът Biotope_1

Забележка:

Всеки симулационен резултат може да се избере и могат да се правят сравнения на изпълненията.

Експериментиране с моделът Biotope_1

В примера динамичните редове се представят чрез линейна диаграма и таблица.

Тя показва стойностите на променливите Hare и Fox на всички времеви точки (времевата стъпка на наблюдателя е 0.1).

Експериментиране с моделът Biotope_1

Забележка:

Можете да са отворени едновременно, много резултатни прозорци.

Ако е избран само един симулационен резултат, могат да се употребят разнообразни математически методи за анализ. В примерът се разглеждат динамичните редове Obs1 # Biotope_1 / Fox.

Когато е избран метода за анализ Fast Fourier Analysis, се появяват входни полета, с които методите могат да се параметризират.

Експериментиране с моделът Biotope_1

Параметризирани модели.

Симулационните резултати могат да се покажат и анализират на всяка точка на прекъсване. Допълнително стойностите на моделните променливи могат да бъдат модифицирани.

В експериментът разгледан тук има прекъсване Break1 и време $T = 10$. Директорията Break1 начално съдържа само крайното състояние End.

След Break1 е избрана и командата Model parameters е била осъществена се появява прозорец. Лявата част на прозореца показва пълен списък на променливите, използвани в този модел.

Експериментиране с моделът Biotope_1

Стойностите на променливите могат да се променят. Резултатния прозорец, съдържа входни полета за желаната стойност на променливата.

Експериментиране с моделът Biotope_1

Например, промяна стойността на променливата Nare от текущото 353.311048 на 444. След потвърждаването на измененията всички промени са съхранени във файл на промените Chg на прекъсването Break1. Резултатът може да бъде прегледан чрез избиране Chg в прозореца на съдържанията.

Изпълнението на симулацията може да бъде продължено.

Когато продължението на изпълнението на симулацията е завършено, Run1 съдържа ново прекъсване с ново крайно състояние End, наречено Break2.

Експериментиране с моделът Biotore_1

Определението на езика на модела SIMPLEX MDL поддържа четири типа компоненти:

- Основни компоненти.

Тези компоненти могат да се изпълняват независимо или да бъдат част от йерархичния модел. Моделът Biotore_1 показва цялостна основна компонента.

- Компоненти от високо равнище.

Simplex3 осигурява йерархично моделиране. Компонентите от високо равнище съдържат структурно описание, когато се описват връзките между второстепенните компоненти.

- Подвижни компоненти.

Подвижните компоненти представят обекти, които могат да се местят от едно на друго място.

- Функционални компоненти.

За да улесни яснотата и структурата на спецификацията на модела, функциите и подфункциите могат да се дефинират.

Експериментиране с моделът Biotope_1

Прекъсвания.

Изпълненията на програмата могат да бъдат прекъснати произволен брой пъти. В началото на изпълнението съществува само Break0. Това начало се състои само от крайното състояние, което има своя стойност по подразбиране.

Възможно е да изменим състоянието End.

Експериментиране с моделът Biotope_1

Командите Show and state, Show change и Show initial state могат да се използват за получаване на кореспондираща информация.

Сега може да се стартира изпълнението на симулацията. Когато изпълнението на симулацията завърши в подходящата точка от време, е достигнато прекъсване Break1. Това първоначално съдържа състоянието за край.

По-нататъшни промени на Break1 генерират нов файл на промените. Състоянието за край End и файла на промените Chg са използвани за генериране за новото начално състояние, с което може да започне нов сегмент на изпълнение.

Експериментиране с моделът Biotope_1

Ако не са направени промени на Break, оригиналното състояние за край ще се използва като начално състояние за следващ сегмент на изпълнение.

Забележка:

Break се състои от крайно състояние на предишно изпълнение на сегмент и промените и новите начални състояния за следващия изпълним сегмент.

Основавайки се на състоянието за край и промените се генерира ново състояние на следващия сегмент на изпълнение.

Всяко прекъсване може да се направи стартираща точка на ново изпълнение.

Експериментиране с моделът Biotope_1

Break1 може да се избере от няколко възможни прекъсвания и да се използва като Break0 в ново изпълнение Run2.

Експериментиране с моделът Viotore_1

Тази функционалност е особено важна, когато се изучават няколко различни набора от параметри.

Версия и състояние на модела

Описанието на всяка версия на компонента на модела се извършва от SIMPLEX MDL.

Версия на компонента се компилира на две фази в обектния код. Получава се изпълним симулационен модел.

Версия и състояние на модела

Процеса на компилиране.

Описанието на модела в SIMPLEX MDL се превежда в С-код от SIMPLEX MDL компилатор, след което се превежда в обектен код от С-компилатор.

Версия и състояние на модела

Една версия може да е в разнообразни състояния, което зависи от позицията ѝ при компилационния процес.

Възможни са преминавания между състоянията на версията. Те или се стартират от потребителя или се изпълняват автоматично чрез симулационната система.

- Непроверена.

Версията е в това състояние веднага след създаването.

Допълнително това състояние е присъединено от симулационната система към версиите, чийто MDL-код токущо е бил изменен. Версията се представя само в SIMPLEX MDL форма.

- Проверена.

На тази стъпка, компилацията чрез MDL-компиляторът е завършила успешно. Моделът се представя като C-код.

- Подготвена.

На тази стъпка, C-кода е бил преведен чрез C-компилятор в обектен код.

Версия и състояние на модела

Забележка:

Състоянието SyntaxOK е налице само при йерархичните модели. То е необходимо, защото една компонента може да съдържа подкомпоненти или самата тя да е подкомпонента на компонента от по-високо равнище.

Версия и състояние на модела

Компилацията се стартира от потребителя чрез избиране на желаната версия на компонентата и избиране на подходяща команда.

Командата `Check version` превежда MDL описанието в С-код, чрез това преминаването на компонентата от състояние Непроверен в състояние Проверено. Ако има грешки в MDL-описанието, преводът се спира и се извежда съобщение за грешка.

Командата `Prepare version` води версията от състояние Проверена в състояние Подготвена. Използва С-компилятор, за да преведе С-кода в обектен код.

Една компонента може да се представи в различни версии. Текущата версия е една от тези версии.

Ако се избере самата версия, текущата версия може да се провери и подготви. Командите `Check current version` и `Prepare current version` са достъпни за компоненти.

Версия и състояние на модела

Състояния на модели.

За да е възможна работата с компонента или в случай на йерархичен модел – йерархия от компоненти трябва да се създаде модел.

Версия и състояние на модела

Модели могат да попаднат в две различни състояния – конфигурирана или инсталирана.

Забележка:

За йерархични модели, които се състоят от повече от една компонента е нужно да се избере само компонентата от най-високо равнище.

В състоянието Конфигурирана, моделът се състои само от име на модел и каталог с версиите, или в случай на йерархичен модел версии, които ще принадлежат на модела. Състоянието се въвежда чрез избиране на компонента и използване на командата `Install model`.

Версия и състояние на модела

Един модел се премества от състояние Конфигуриран в състояние Инсталиран, използващ командата `Install model`.

Тази команда се достига чрез десния клавиш на мишката, след като се избере моделът да бъде инсталиран. Компилираните версии на компонентите се свързват със системата за изпълнение, за да се създаде изпълнима програма.

Командата `Reset` изтрива тази изпълнима програма. Остава само каталога, съдържащ компонентата версия. Това състояние на преход се извършва автоматично, когато потребителят промени версията на компонентата, принадлежаща на модела, докато тези модификации изискват изпълняваната програма да се прекомпилира.

Версия и състояние на модела

Замисъл на версията.

Всеки компонент може да има различни версии. Версиите са независими една от друга.

Едно сходство илюстрира ситуацията:

Всеки член на семейство си има своя идентичност и име. Те отговарят на версия Членове на едно и също семейство, имат обща фамилия, която ги присъединява в единица от по-високо равнище. Семейството (фамилията) отговаря на компонентата (име).

За да се идентифицира уникално член на едно семейство са необходими и фамилията и индивидуалното първо име. Подобно обект на модел се описва от неговите компонента и имена на версията.

В Simplex3 една версия може да се декларира като текуща версия, което я различава от другите. Текущите версии са достъпни чрез техните компонентни имена.

Версия и състояние на модела

Когато компонента се разшири, принадлежащата ѝ версия става видима. Текущата версия може да се открие чрез избиране на компонентната директория.

Прозорецът на съдържанията показва всички компоненти, включително и версиите им.

Пример:

Моделният обект Biotope_1 / Version0 ще се направи текуща версия. Тогава е достатъчно просто да се даде компонентното име Biotope_1, за да се получи достъп и да се редактира точно тази версия.

Версия и състояние на модела

Решаващото преимущество на концепцията на версията се изяснява, когато се използват йерархичните компоненти. Не са необходими никакви промени на описанието на модела, когато се промени една версия.

Забележка:

Ако един компонент се състои само от една версия, това автоматично е текущата версия. Всички команди, които са валидни за версиите са валидни също и за компонентите.

Версия и състояние на модела

Първите и последните редове трябва да съдържат новото име на компонентата „Biotope_Test”, вместо „Biotope_1”.

Засегнати са следните два реда:

BASIC COMPONENT Biotope_Test

END OF Biotope_Test

- Проверка и подготовка на компонентата:

Подготовката включва проверката. В този случай може да се използва направо командата Prepare version.

Версия и състояние на модела

Упражнение1:

Създайте нова компонента, наречена Biotope_2Priv. Това се състои от допълнително събитие, отговарящо на компонентата Biotope_1.

Когато броя на зайците става под 300, 10% от лисиците се премахват.

Това събитие има следната форма в SIMPLEX MDL:

```
ON ^Hare < 300[NumH]^
```

```
DO
```

```
    Fox^ := Fox – Fox/10;
```

```
    DISPLAY (“ Vreme T %f \ Broi na premahnatite lisici: %f \n\n”, т,  
Fox/10);
```

```
END
```

Събитието трябва да се вмъкне след ключовата дума DYNAMIC
BEHAVIOUR.

Упражнение2:

Вместо развиването на новата компонента като независима компонента, именувана Biotope_2 Priv, създайте я като Version3 на компонентата Biotope_1.

Версия и състояние на модела

Команден способ.

Най-лесно е да се използват менютата. Това е по-бавния способ. По-бързия способ е да се използва командния способ.

Въвеждане на команди.

Командите се въвеждат в penultimate line на SIMPLEX прозорец със заглавието Command Interpreter. Тук желаните команди заедно с подходящите им параметри могат да се въведат от клавиатурата.

Версия и състояние на модела

Валидни са следните правила:

1. Разделителят между името на командата и параметрите е символът интервал.

Пример:

SelBank Biotope

2. Разделителят между параметрите е символът интервал.

Пример:

AddVar Window Hare Fox

SetVar Hare 500

Общия команден синтаксис има следния вид:

Cd1 Par1 Sep Par2 Sep... ParN;

където Cd1...CdN са команди в експерименталната среда,
Par1...ParN са съответни параметри и Sep се използва за
означаване на символа интервал.

Версия и състояние на модела

3. Поправки се правят чрез поставяне курсора от дясно на грешния символ, използвайки курсорните клавиши. При натискане на клавиша Delete този символ ще се изтрие. Тогава нов символ може да се въведе пред курсорът.
4. Всички команди, които се изпълняват се запазват по ред. Чрез натискане на стрелките нагоре и надолу списъка команди може да бъде обхождан и предишни команди могат да бъдат пренесени като команди за редактиране и въвеждане. Командите, които се появяват в полето могат да бъдат модифицирани. При натискане на курсорния клавиш върху командния вход се получава ново командно поле.
5. Когато обектите се създадат, преименуват и копират, новото име е чувствително за големи и малки букви. По-късно потребителят може да използва и главни и малки букви.

Пример:

SelBank Biotope
is identical to
selbank biotope

Версия и състояние на модела

6. Имената на командите могат да се съкращават докато са уникални. Съкращенията не могат да се използват за параметри.

Пример:

SelBank Biotope

is identical to

selb biotope

7. Цялото съдържание на командното поле се въвежда, когато се натисне клавишът Return. Позицията на курсора в командния ред се пренебрегва.

Версия и състояние на модела

Библиотеки на модела.

Всички компоненти, които потребителя разглежда за принадлежащи, заедно могат да се обединят от библиотека на модела. Отделните потребители могат да имат няколко моделни библиотеки, които всички принадлежат на неговия личен `private` каталог от библиотеки. Не е възможен достъп до `private` каталог от библиотеки и затова също и до моделните библиотеки на други потребители.

В допълнение на `private` каталозите от библиотеки, принадлежащи на индивидуален потребител, е достъпен `public` каталог от библиотеки. Всички потребители имат достъп до този каталог от библиотеки. `Public` каталога от библиотеки се използва основно за размяна на компоненти между потребители.