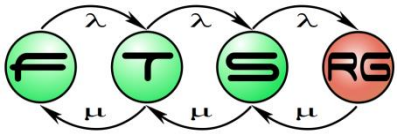
Kódgenerálás

Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport

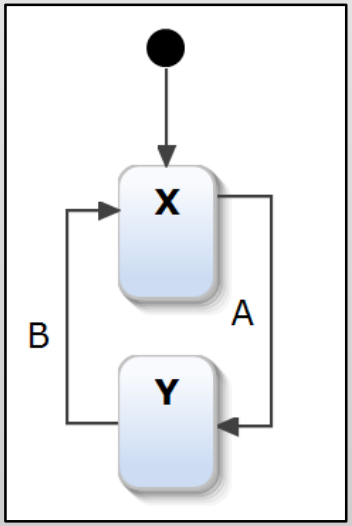


MODELLEZŐ ESZKÖZ FUNKCIÓI

Yakindu modellezési funkciói

Konkrét szintaxis (Szerkesztők)

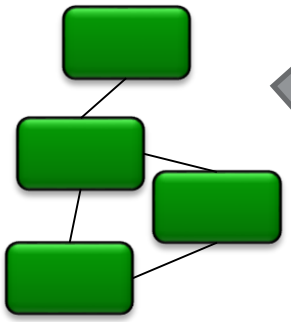
Grafikus



Szöveges

Demo
internal:
event A
event B

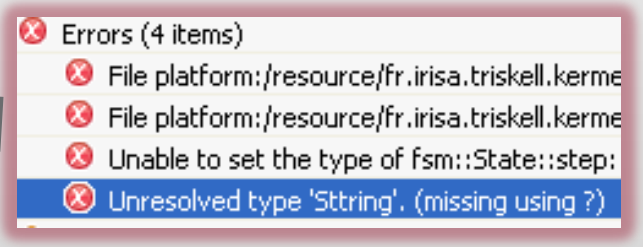
Szintaxis → Szemantika



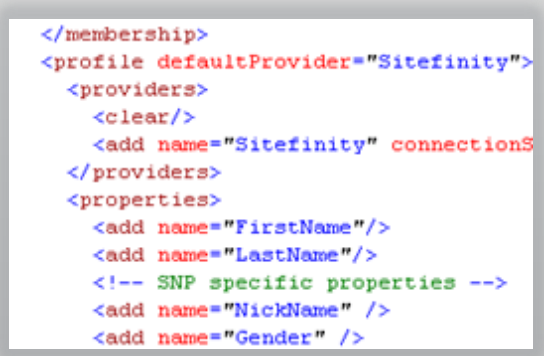
Modell
(Absztrakt szintaxis)

Modellező funkciók

Ellenőrzés



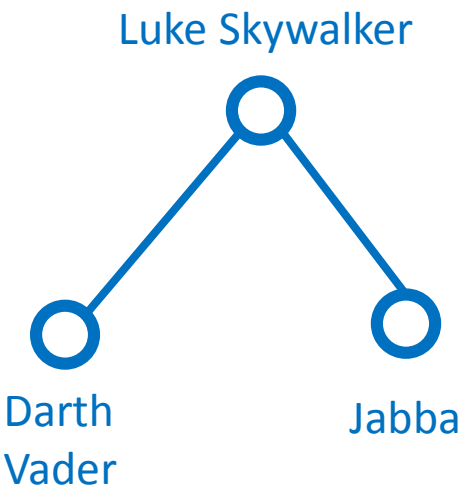
Kódgenerálás

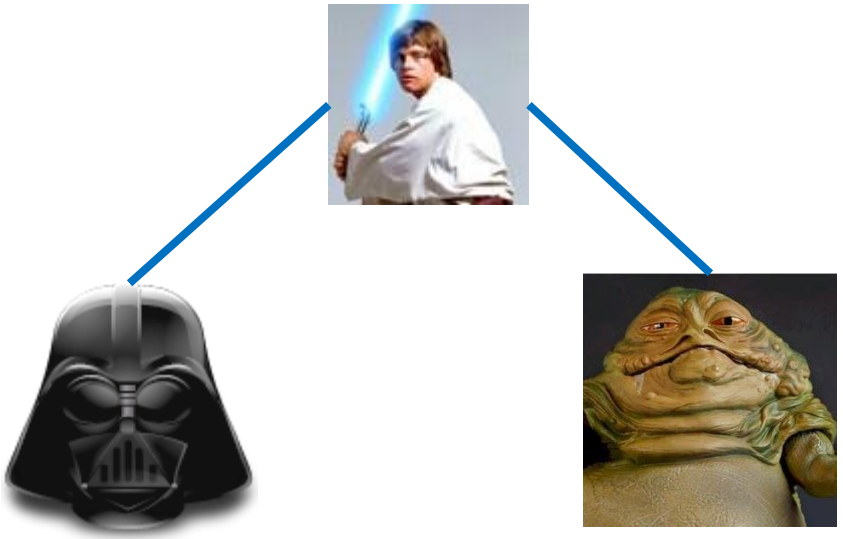


(Forráskód, dokumentáció, konfiguráció)

Absztrakt szintaxis

- **Definíció:** a szerkesztés alatt álló rendszer strukturális modellje.
- Modellező program kezeli
- Emlékeztető: strukturális modell = **gráf**
  - **csomópontok, élek és tulajdonságok gráfja**

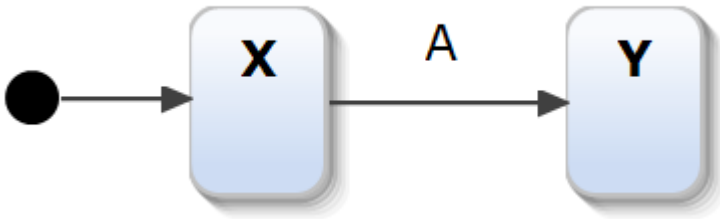




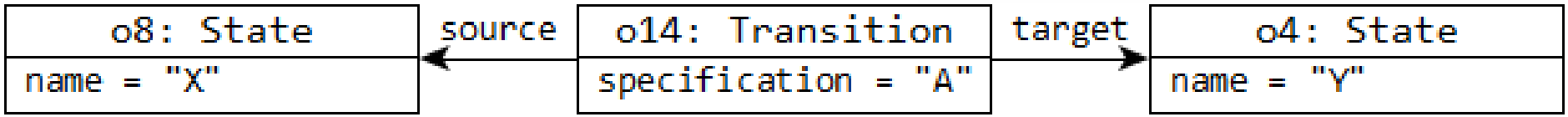
Absztrakt szintaxis példa: Yakindu

Kérdés:
Hogyan készítenénk modellező programot?

Példa: Yakindu modell



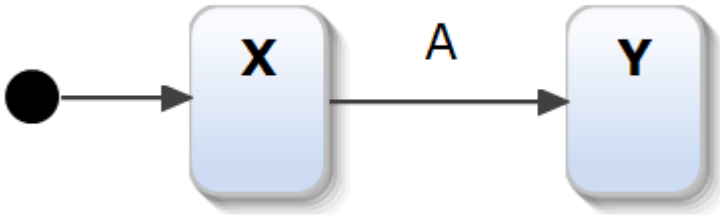
Abstract Syntax



Absztrakt szintaxis példa: Yakindu

Kérdés:
Hogyan készítenénk modellező programot?

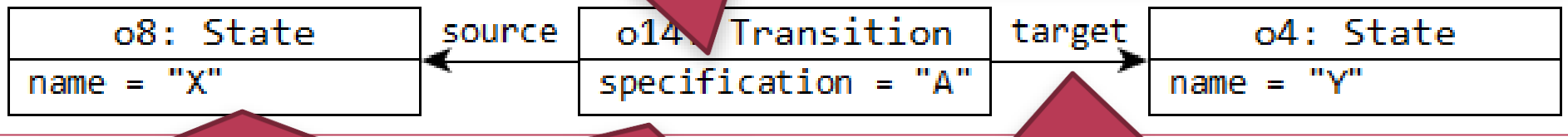
Példa: Yakindu modell



Neveket String-ként tároljuk

```
name = "A"
```

Abstract Syntax



Modellelemeket objektumként

Kapcsolatokat pointerekkel

Válasz: Egyszerű objektum-orientált program

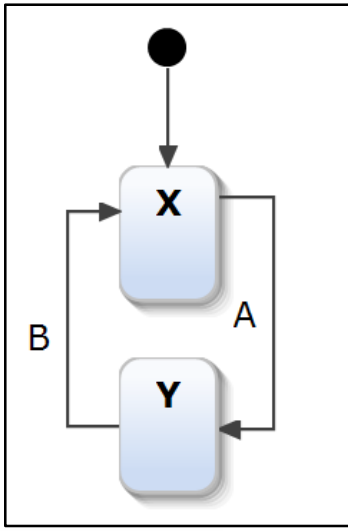
Konkrét szintaxis

- **Cél:** konkrét megjelenítés ⇔ mögöttes modell
- Szöveges szintaxis (programozási nyelv)
  - Feladat: Szöveg → Modell
  - Szabályok alapján (egyébként nehéz!)

    <Digit>::= "**1**" | "**2**" | "**3**" | "**4**" | … | "**9**" | "**0**"

    <Number>::= <Digit> | **<Digit> <Number>**
- Grafikus szintaxis (Diagram)
  - Könnyebben átlátható, nehezebben írható
  - **Itt is:** vannak saját technológiák

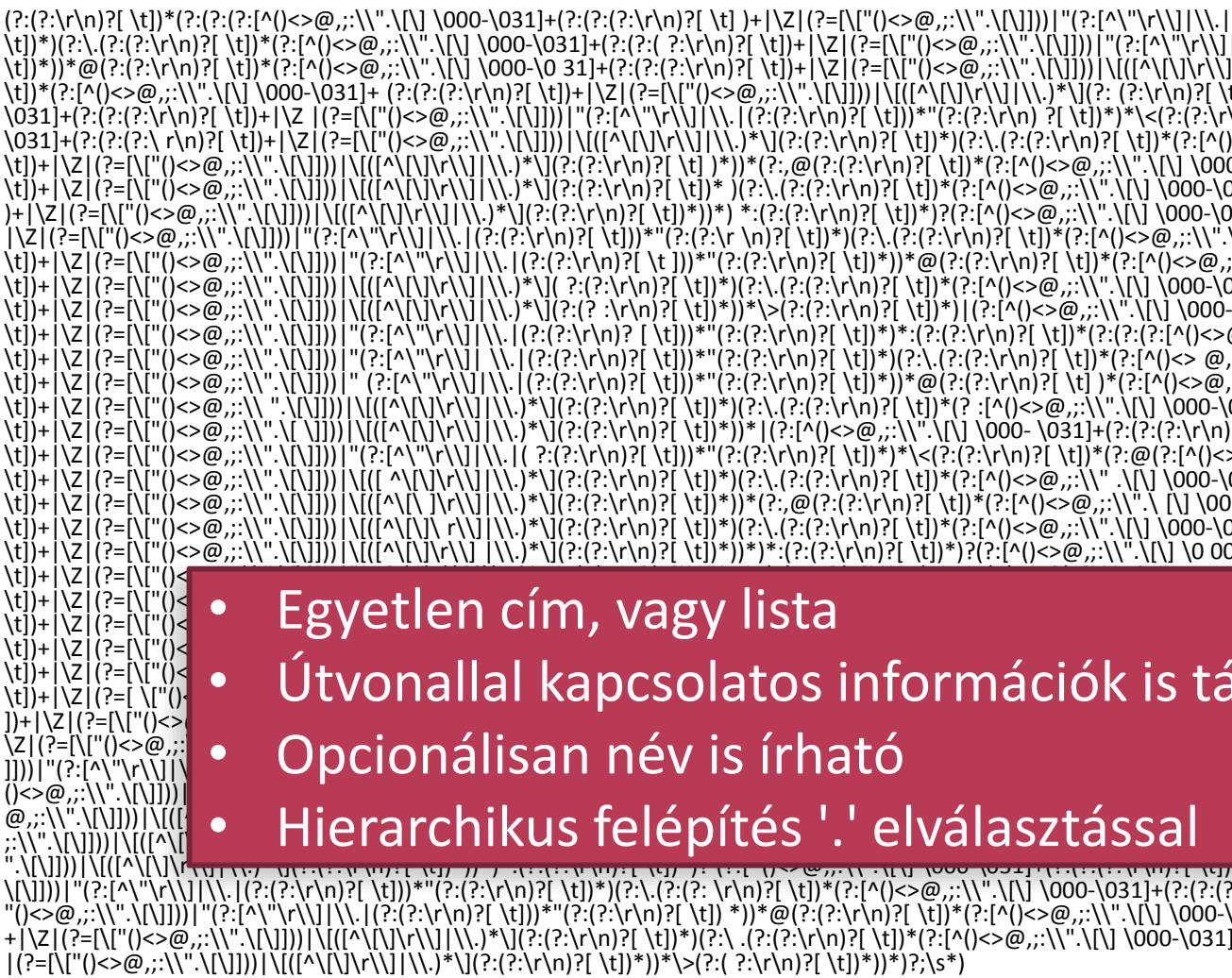
Demo
internal:
event A
event B



Megfelelő módszerekkel **mindenki** tud **saját** programozási nyelvet csinálni!

Ellenpélda megfelelő módszerre

- **Feladat:** reguláris kifejezéssel eldönteni szövegről email cím-e



- Egyetlen cím, vagy lista
- Útvonallal kapcsolatos információk is tárolhatóak
- Opcionálisan név is írható
- Hierarchikus felépítés '.' elválasztással

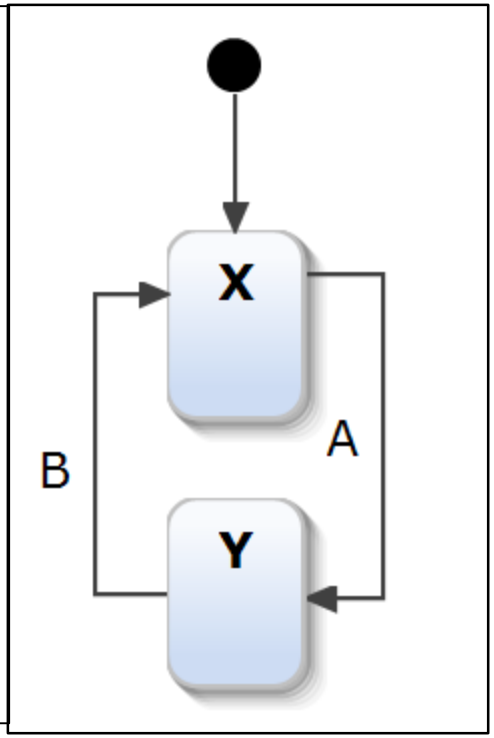
Forrás: http://www.ex-parrot.com/~pdw/Mail-RFC822-Address.html

KÓDGENERÁTOR

Kódgenerátor példán keresztül:

- **Feldadat:**
  Generáljunk C kódot
  Yakindu állapotgépekből

Demo
internal:
event A
event B



- Írjunk olyan függvényt:
  → kap egy Modell objektumot
  ← visszaad egy szöveget
- A szöveg egy „Demo.c” fájlba kerül
- Fordító lefordítja

Sablon alapú kódgenerátor (Xtend)

- Cél: Állapotok → Enum

```
enum states {
   StateX,
   StateY
};
```

Összevágunk egy char*-ba a kimenetet, %s helyére írjuk az X,Y neveket

- Megoldás: C program

```
sprintf(result,
   "enum states {\n\tState%s,\nState%s\n};",
   state1->name,
   state2->name);
```

Ez egyből működik!
☹ nehezen átlátható.

- Sablon (Xtend):

```
'''
enum states {
    State«state1.name»,
    State«state2.name»
}'''
```

Megadjuk a változások helyét

Megírjuk a sablonosan

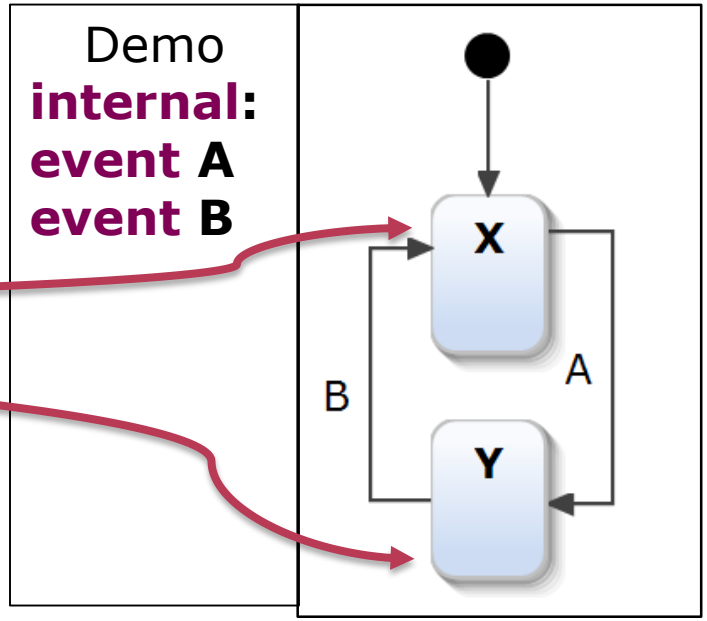
☺ Könnyebb írni
☺ Átláthatóbb
☺ Könnyű módosítani
☹ +1 technológia

Kódgenerátor példa: Állapotok

- Várt C kód:

```
//States of the statemachine
enum states {
   StateX,
   StateY
};
```

Lehetséges állapotok:
Enumként felsorolva



- Sablon:

```
//States of the statemachine
enum states {
«FOR state : states»
   State«state.name»
«ENDFOR»
};
```

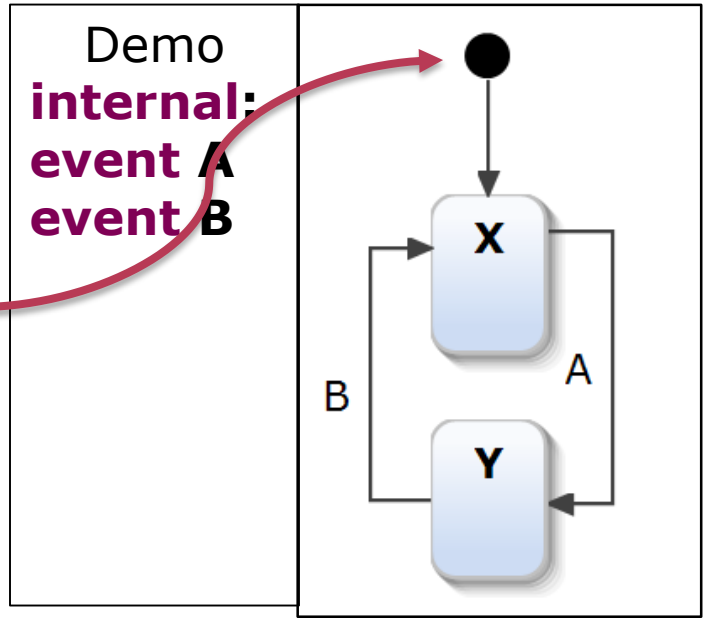
1.Összes állapoton végigmegyünk
2.Vesszővel elválasztva Írjuk ki:
State«név»
Pl: StateX

Kódgenerátor példa: kezdőállapot

- Várt C kód:

```
// The actual state
// First = entry state.
enum states actualState = StateX
```

Aktuális állapot = kezdőállapot



- Sablon:

```
// The actual state
// First = entry state.
enum states actualState = State«findEntry(states).name»
```

1.Megkeressük a kezdőelemet
2.Kiírjuk a nevét

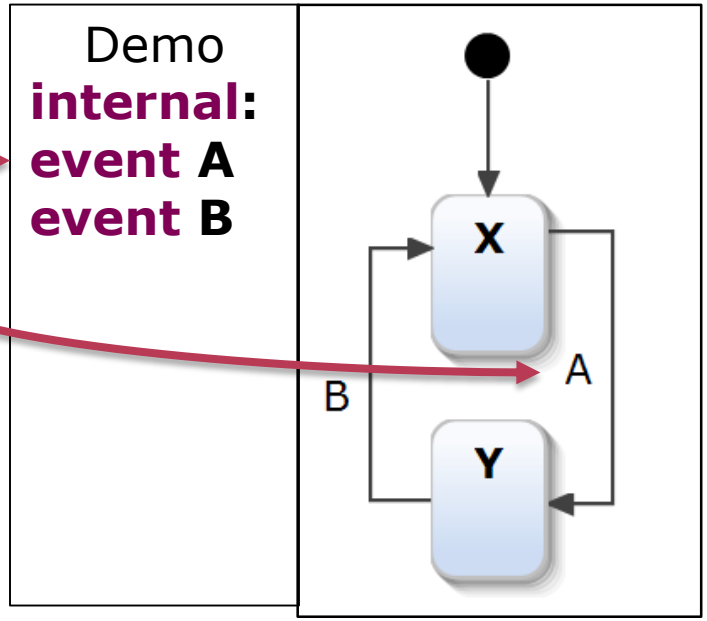
Kódgenerátor példa: Állapotátmenet

- Várt C kód:

```
// Execute "A" event
void doA{
switch(actualState) {
        case StateX:
            actualState = StateY;
            break;
        case StateY:
            break;
    }
}
```

A / X → Y

A / -

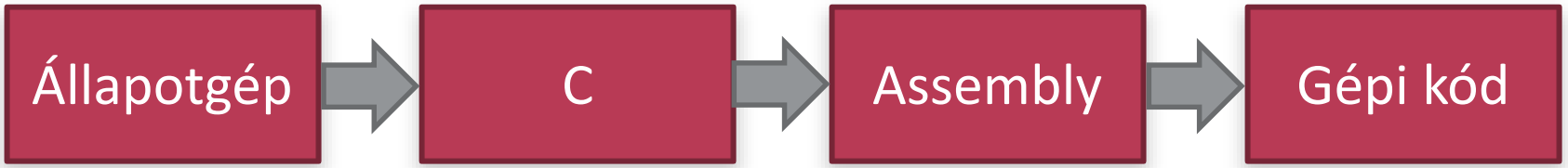


- Sablon (vázlatosan):
  1. Minden eseményhez egy `do«Esemény neve»` függvény
  2. A tranzíciók alapján mást csinál a függvény

Egy (egyszerű) állapotgéphez ennyi a kódgenerátor!

Kódgenerátor Összefoglalás

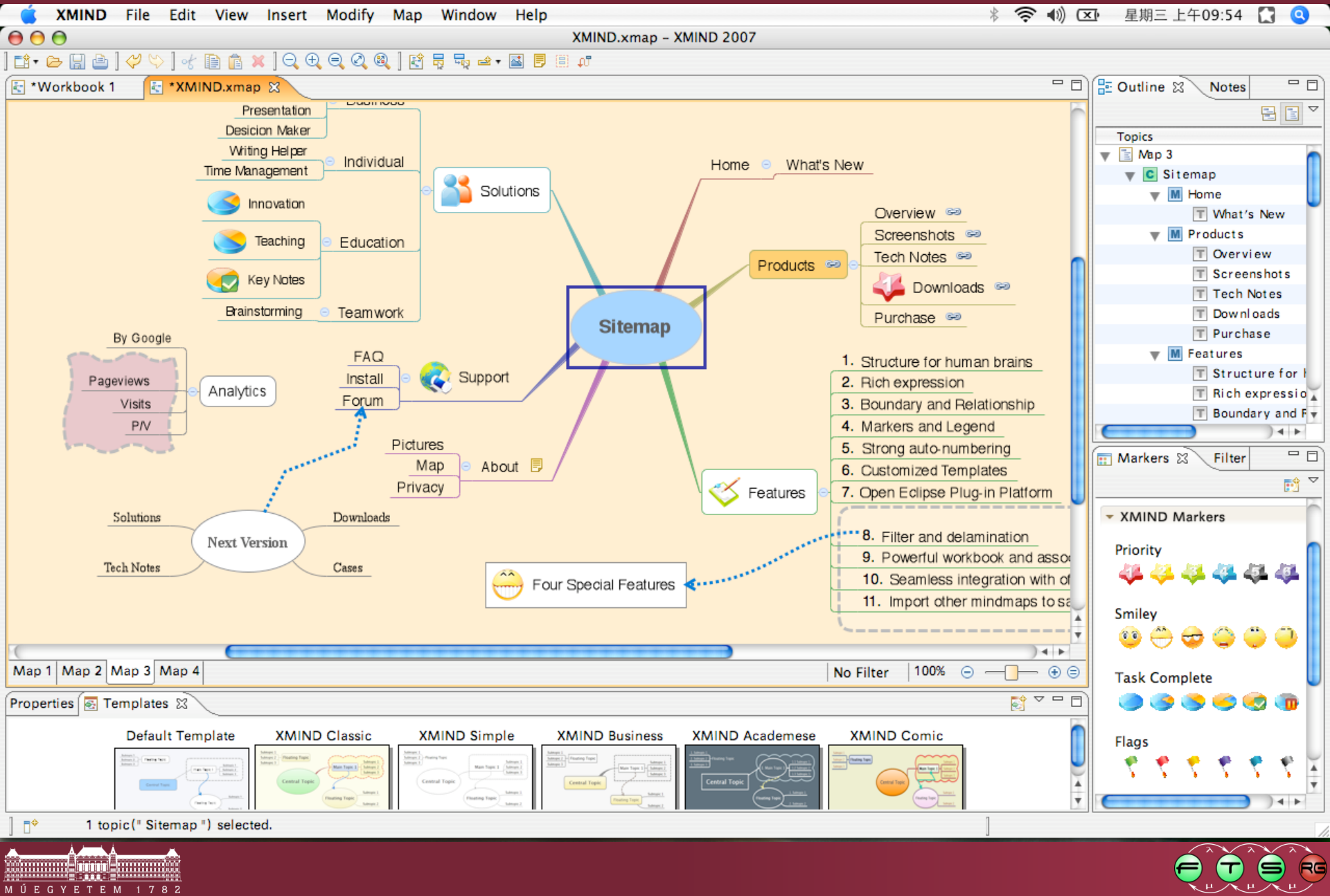
- Kódgenerálás = Fordító
- Ugyanaz a lépés:



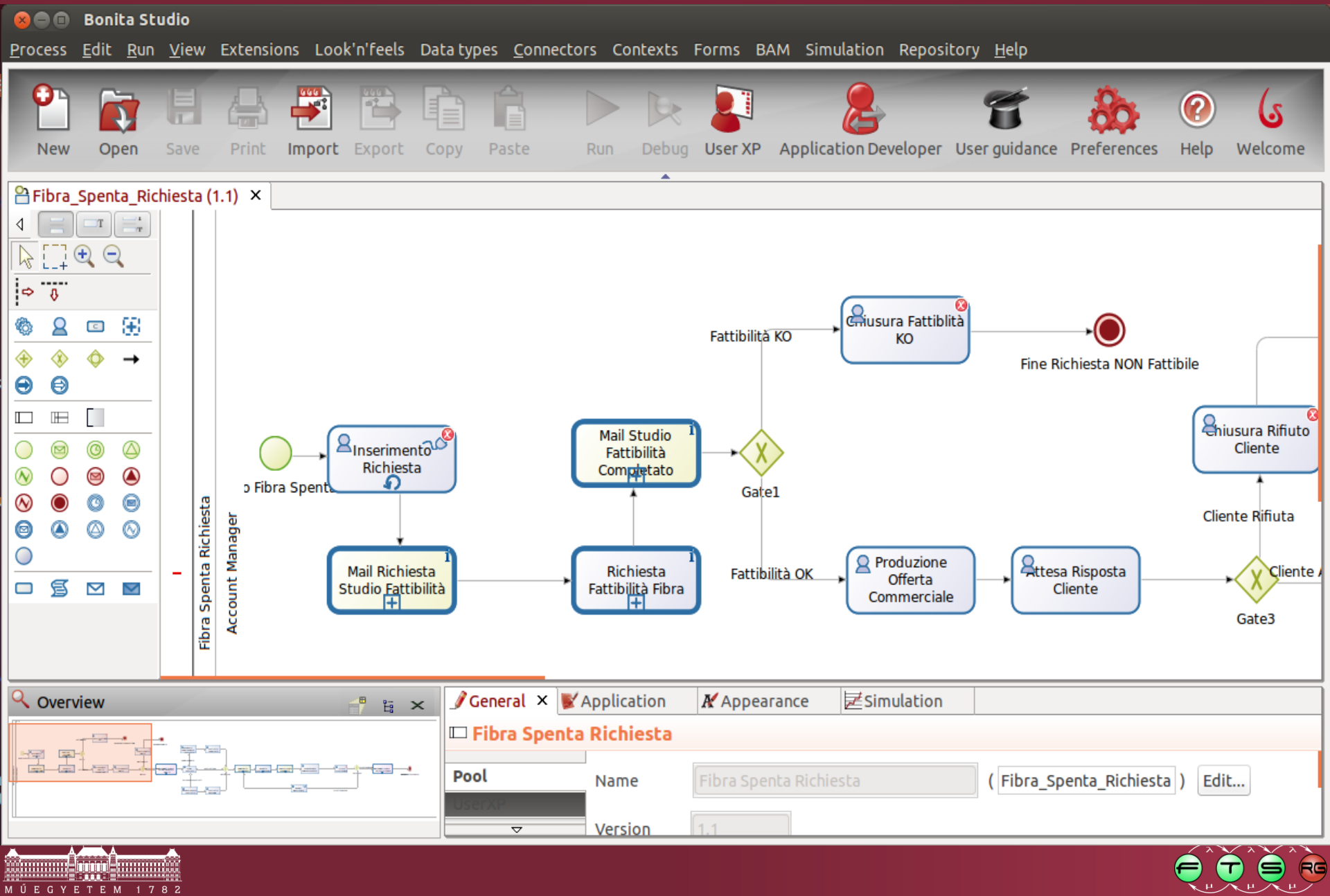
- A problémát a saját nyelvén: Produktivitás ++
- Sok unalmas, bonyolult kód automatikusan Teljesítmény ++
- Ellenőrizzük a saját nyelvén: Megbízhatóság ++
- Tanszékünkön fejlesztett projektek: akár **95%** generált kód

PÉLDA ECLIPSE ALAPÚ MODELLEZŐ ESZKÖZÖK

XMind: Mindmap tervező



Bonita: Üzleti folyamatok modellezése



Kalypso: Vízügyi tervezés + szimuláció