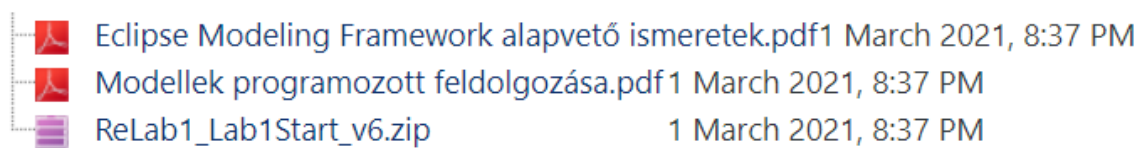


2. Rendszerteverzés laboratórium

Készítette:	Halász Bence Botond(Q7RW4E)
Dátum:	2021.03.15
Github hivatkozás:	https://github.com/SteveP1999/2.ReteLab
Imsc feladat	Nem készült el

Letöltöttem az edu.vik.bme.hu-n található fájlokat.

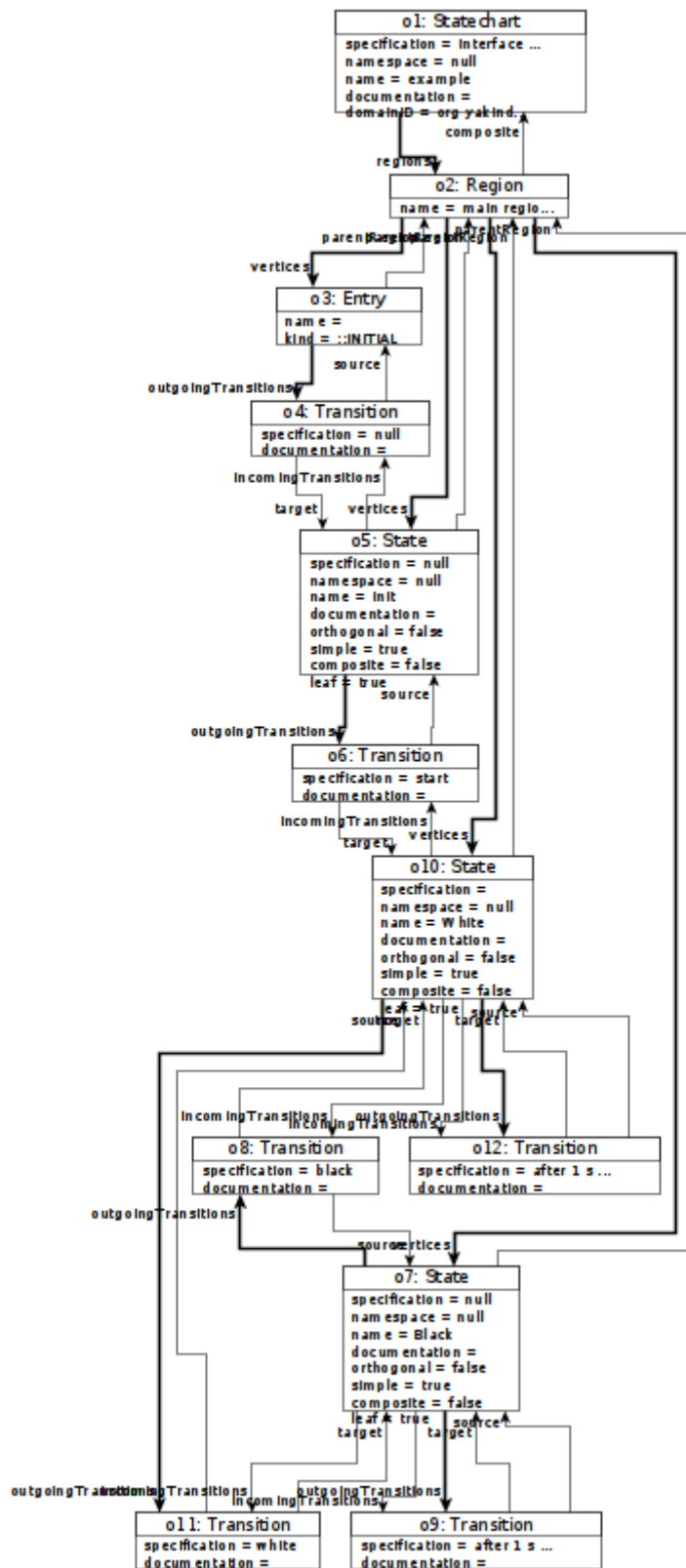


Eolvastam az Eclipse Modeling Framework alapvető ismeretek című pdf-et majd elkezdtem a másik pdf-ben leírt feladatokat, de még előtte letöltöttem az előre összekészített virtuális gépet.

- Vagy letölteni egy összekészített virtuális gépet az alábbi címről: <https://share.mit.bme.hu/index.php/s/WBeGjGHXPdFsW4t>

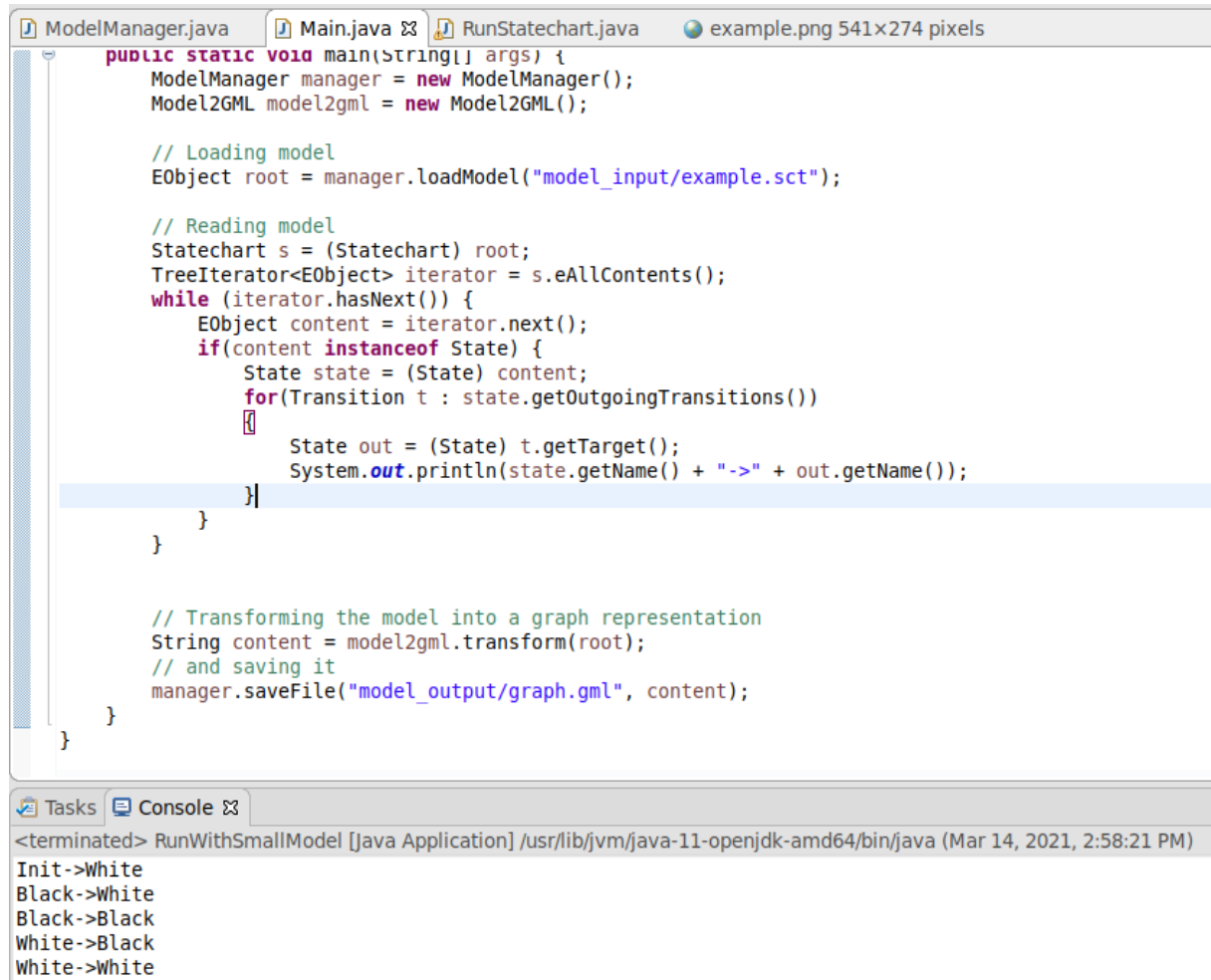
Elindítottam a virtuális gépet majd megnyitottam a Yakindu STC-t, beállítottam amit kértek a feladatkiírásban majd rámásoltam a virtuális gépre a kezdeti projekt .zip kiterjesztésű állományt amit ezután importáltam és átnéztem, hogy mit tartalmaz. Itt kezdődött a második feladat.

Megnyitottam a gráfvizuálást és az layoutban elrendeztem:



2.3 feladat:

Kiegészítettem a kódot, hogy megkeresse a következő állapotot minden állapotnál és kiírja a nevét. Alul látható a kimenet.



The screenshot shows an IDE with three tabs: `ModelManager.java`, `Main.java`, and `RunStatechart.java`. The `Main.java` tab is active, displaying the following Java code:

```
public static void main(String[] args) {
    ModelManager manager = new ModelManager();
    Model2GML model2gml = new Model2GML();

    // Loading model
    EObject root = manager.loadModel("model_input/example.sct");

    // Reading model
    Statechart s = (Statechart) root;
    TreeIterator<EObject> iterator = s.eAllContents();
    while (iterator.hasNext()) {
        EObject content = iterator.next();
        if (content instanceof State) {
            State state = (State) content;
            for (Transition t : state.getOutgoingTransitions())
                State out = (State) t.getTarget();
            System.out.println(state.getName() + "->" + out.getName());
        }
    }

    // Transforming the model into a graph representation
    String content = model2gml.transform(root);
    // and saving it
    manager.saveFile("model_output/graph.gml", content);
}
```

Below the code editor, the `Console` tab is active, showing the output of the program:

```
<terminated> RunWithSmallModel [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 14, 2021, 2:58:21 PM)
Init->White
Black->White
Black->Black
White->Black
White->White
```

2.4 feladat:

Kiegészítettem a fenti kódot egy plusz if-fel ahol ellenőrzöm, hogy üres-e az outgoing transaction lista valamelyik állapotnál, mert amennyiben igen akkor az egy csapda állapot. Ilyenkor kiíratom ennek az állapotnak a nevét.

```
ModelManager manager = new ModelManager();
Model2GML model2gml = new Model2GML();

// Loading model
EObject root = manager.loadModel("model_input/example.sct");

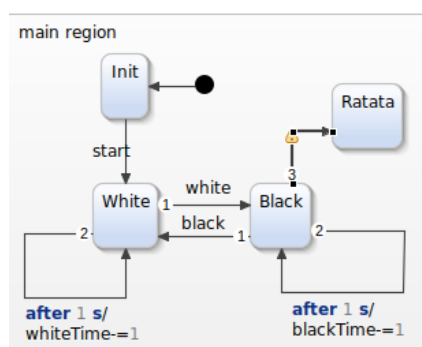
// Reading model
Statechart s = (Statechart) root;
TreeIterator<EObject> iterator = s.eAllContents();
while (iterator.hasNext()) {
    EObject content = iterator.next();
    if (content instanceof State) {
        State state = (State) content;
        for (Transition t : state.getOutgoingTransitions()) {
            State out = (State) t.getTarget();
            System.out.println(state.getName() + "->" + out.getName() + "\t");
            if (out.getOutgoingTransitions().isEmpty() == true) {
                System.out.println("Ez egy csapda állapot: " + out.getName());
            }
        }
    }
}
```

Tasks Console

<terminated> RunWithSmallModel [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 14, 2021, 3:46:38 PM)

Init->White
Black->White
Black->Black
Black->Ratata
Ez egy csapda állapot: Ratata
White->Black
White->White

Továbbá kiegészítettem az example.sct-t, hogy legyen csapda állapot ami a ratata névre hallgat. A fenti képen látszik a kiemenet.



2.5 feladat:

```

// Reading model
Statechart s = (Statechart) root;
TreeIterator<EObject> iterator = s.eAllContents();
while (iterator.hasNext()) {
    EObject content = iterator.next();
    if (content instanceof State) {
        State state = (State) content;
        for (Transition t : state.getOutgoingTransitions())
        {
            int i = 0;
            State out = (State) t.getTarget();
            System.out.println(state.getName() + "->" + out.getName() + "\t");
            if (out.getOutgoingTransitions().isEmpty() == true)
            {
                System.out.println("Ez egy csapda állapot: " + out.getName());
            }
            if (out.getName() == "")
            {
                System.out.println("Legyen jelen állapot neve mondjuk: NewState" + i);
                i++;
            }
        }
    }
}

```

Hozzáírtam még egy plusz if ciklust ami megnézi, hogy az állapot neve üres-e és amennyiben igen akkor javasol neki egy nevet csak nem egészen értettem, hogy mit jelent az, hogy alkalmas név ezért most csak beírtam egy random javasolt nevet.

Kitöröltem a ratata állapot nevét majd futtattam a kódot é sez lett az eredmény:

```

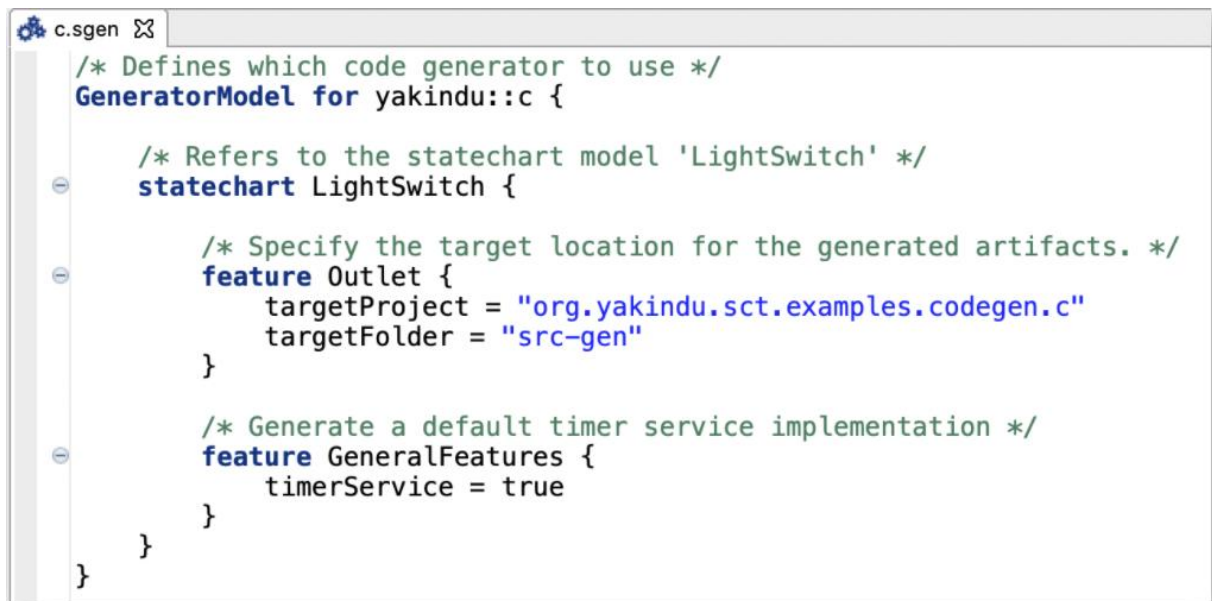
Init->White
Black->White
Black->Black
Black->
Ez egy csapda állapot:
Legyen jelen állapot neve mondjuk: NewState0
White->Black
White->White

```

3.1 feladat:

Kitöröltem a névtelen új állapotot.

3.2 feladat:



```
/* Defines which code generator to use */
GeneratorModel for yakindu::c {

    /* Refers to the statechart model 'LightSwitch' */
    statechart LightSwitch {

        /* Specify the target location for the generated artifacts. */
        feature Outlet {
            targetProject = "org.yakindu.sct.examples.codegen.c"
            targetFolder = "src-gen"
        }

        /* Generate a default timer service implementation */
        feature GeneralFeatures {
            timerService = true
        }
    }
}
```

Ez az oldal alapján azt mondanám, hogy itt különböző általános szolgáltatásokat tudunk hozzáadni továbbá letiltani.

TimeService: Lehetővé teszi, hogy generáljunk egy idő szolgáltatás bármilyen különböző célhoz.

RunTimeService: Ez hasonló mint a TimeService csak kifejezetten arra, hozzá létre az idő számlálót, hogy aktiváljon egy kör ciklust aktiváljon egy state machine alapján.

InterfaceObserverSupport: Letiltja, hogy listener generáljunk egy statemachine diagramhoz.

3.3. feladat:

Futtattam az alábbi kódrészletet és ez lett az eredmény:

```
package hu.bme.mit.yakindu.analysis.workhere;

import java.io.Console;
import java.io.IOException;
import hu.bme.mit.yakindu.analysis.RuntimeService;
import hu.bme.mit.yakindu.analysis.TimerService;
import hu.bme.mit.yakindu.analysis.example.ExampleStatemachine;
import hu.bme.mit.yakindu.analysis.example.IExampleStatemachine;

public class RunStatechart {

    public static void main(String[] args) throws IOException {
        ExampleStatemachine s = new ExampleStatemachine();
        s.setTimer(new TimerService());
        RuntimeService.getInstance().registerStatemachine(s, 200);
        s.init();
        s.enter();
        s.runCycle();
        print(s);
        s.raiseStart();
        s.runCycle();
        System.in.read();
        s.raiseWhite();
        s.runCycle();
        print(s);
        System.exit(0);
    }
}
```

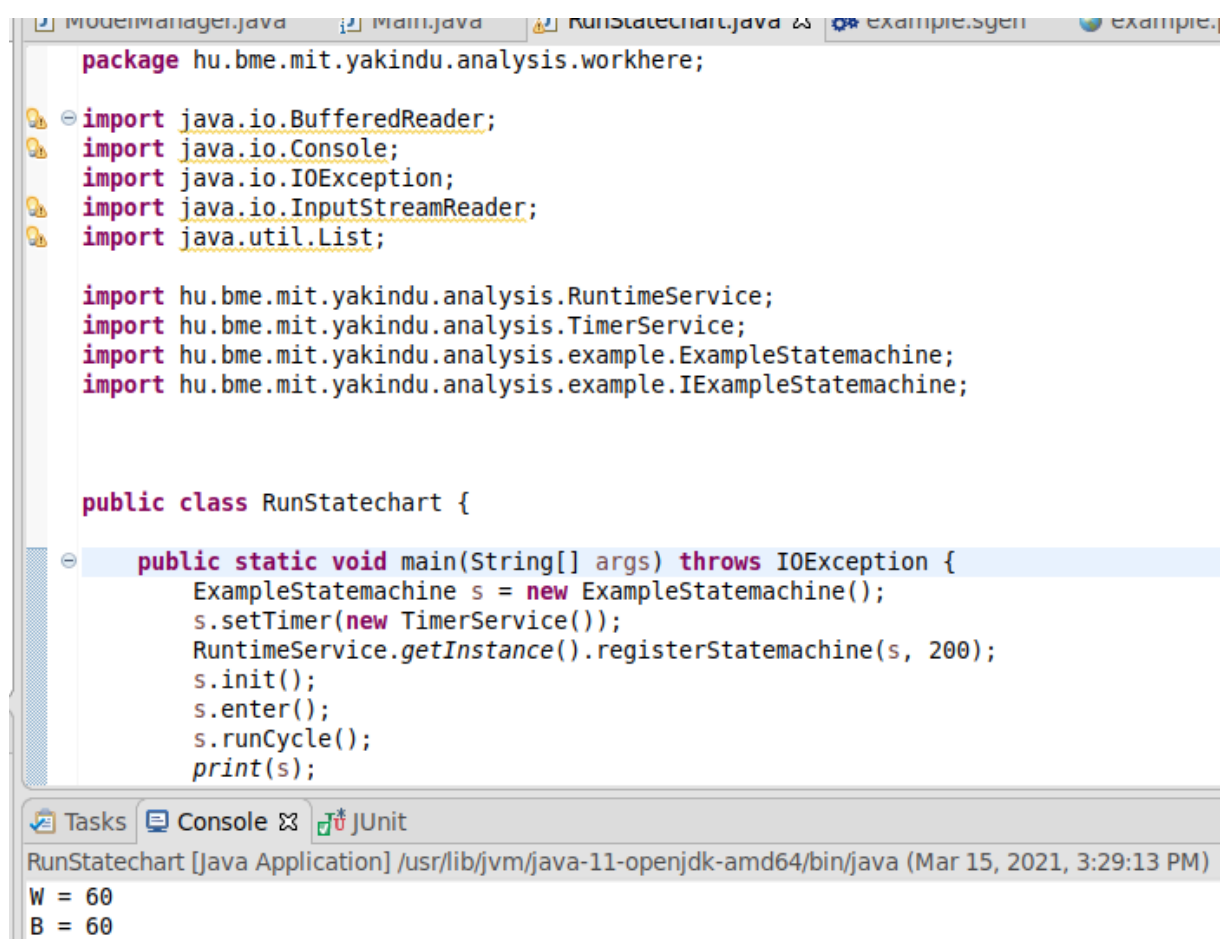
Tasks Console

RunStatechart [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 14, 2021, 4:49:42 PM)

W = 60
B = 60

3.4 feladat:

Futtattam, ez let az eredmény:



```
package hu.bme.mit.yakindu.analysis.workhere;

import java.io.BufferedReader;
import java.io.Console;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;

import hu.bme.mit.yakindu.analysis.RuntimeService;
import hu.bme.mit.yakindu.analysis.TimerService;
import hu.bme.mit.yakindu.analysis.example.ExampleStatemachine;
import hu.bme.mit.yakindu.analysis.example.IExampleStatemachine;

public class RunStatechart {

    public static void main(String[] args) throws IOException {
        ExampleStatemachine s = new ExampleStatemachine();
        s.setTimer(new TimerService());
        RuntimeService.getInstance().registerStatemachine(s, 200);
        s.init();
        s.enter();
        s.runCycle();
        print(s);
    }
}
```

Tasks Console JUnit

RunStatechart [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 15, 2021, 3:29:13 PM)

W = 60
B = 60

3.5 feladat:

```

boolean condition = true;
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
ExampleStatemachine s1 = new ExampleStatemachine();
s1.setTimer(new TimerService());
RuntimeService.getInstance().registerStatemachine(s1, 200);
s1.init();
s1.enter();
s1.runCycle();
while(condition == true)
{
    String str = reader.readLine();
    if(str.equals("start"))
    {
        s1.raiseStart();
        s1.runCycle();
        print(s1);
    }
    if(str.equals("white"))
    {
        s1.raiseWhite();
        s1.runCycle();
        print(s1);
    }
    if(str.equals("black"))
    {
        s1.raiseBlack();
        s1.runCycle();
        print(s1);
    }
    if(str.equals("exit"))
    {
        condition = false;
        print(s1);
        System.exit(0);
    }
}
}

```

Kimenetre példa:

```

start
W = 60
B = 60
white
white
W = 58
B = 60
black
W = 58
B = 57
black
W = 40
B = 57
exit
W = 38
B = 57

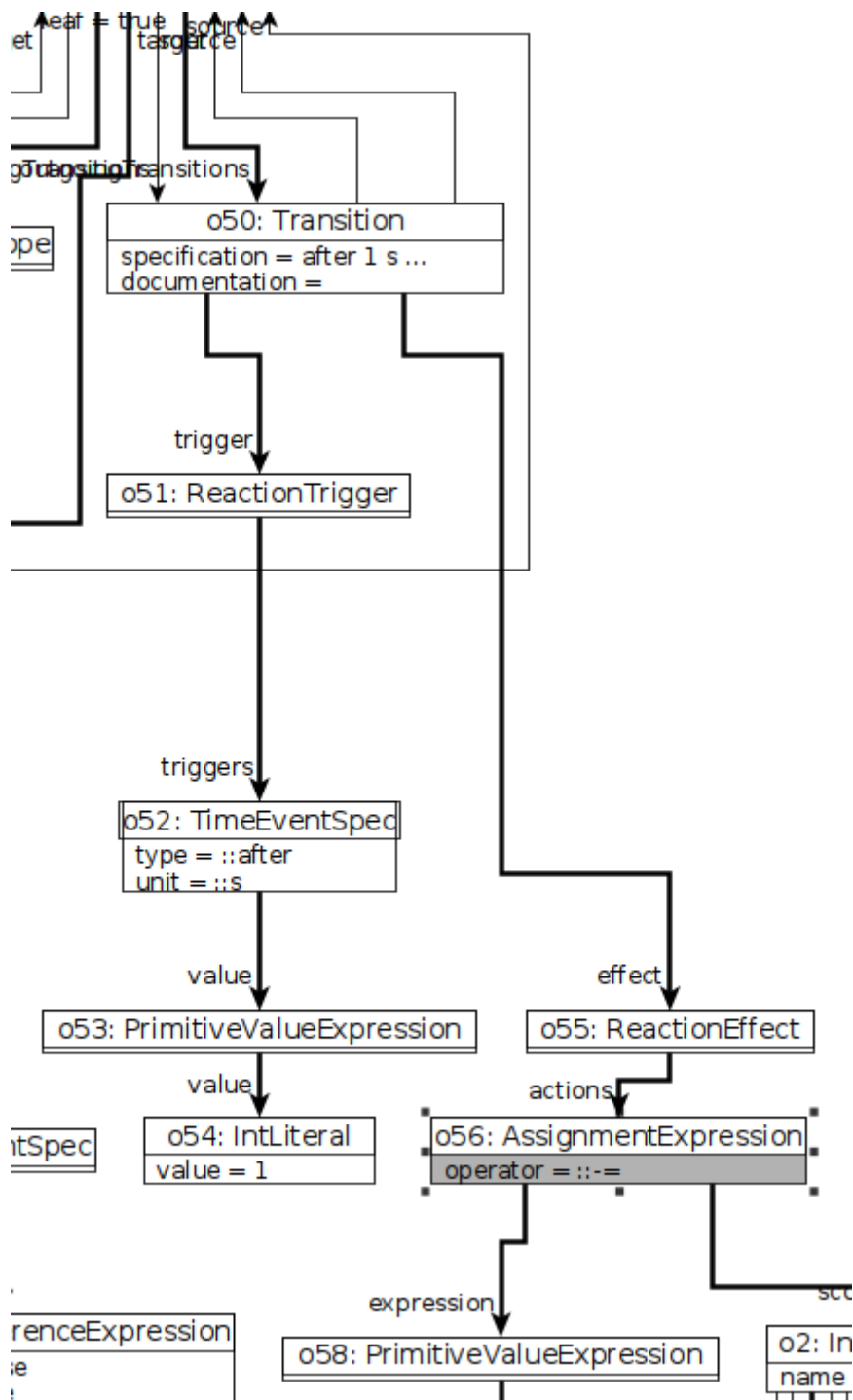
```

4.1 feladat:

A fa kibővült rengeteg új ággal, mint például a whitetime -= 1 és blacktime -= 1 kifejtése és még sok mássa. Ez annak köszönhető, hogy a RunWithBig model lehetővé tette, hogy a teljes example.sct-t lefuttassuk.

4.2 feladat:

Így néz ki ami szerintem hozzá tartozik és még van ami a képen nem látszik egy-kettő ág ami folytatódik csak nem fért rá a képre.



4.3 feladat:

Ez let a plusz kiegészítés:

```
    }  
    else if(content instanceof EventDefinitionImpl)  
    {  
        EventDefinitionImpl edi = (EventDefinitionImpl) content;  
        System.out.println("Bemenő esemény: " + edi.getName());  
    }  
    else if (content instanceof VariableDefinition)  
    {  
        VariableDefinitionImpl vdi = (VariableDefinitionImpl) content;  
        System.out.println("Belső változó" + vdi.getName());  
    }  
}
```

4.4 feladat:

Ezt a kódot írtam hozzá, de valamiért nem működik. Vagyis még a mainben amikor átadok neki egy List<String> listát az üres és nem értem miért.

Oda így veszem fel a dolgokat:

A program elején meg létrehozok kettő arraylistet egy event meg egy var néven.

```
    else if(content instanceof EventDefinitionImpl)  
    {  
        EventDefinitionImpl edi = (EventDefinitionImpl) content;  
        System.out.println("Bemenő esemény: " + edi.getName());  
        event.add(edi.getName());  
    }  
    else if (content instanceof VariableDefinition)  
    {  
        VariableDefinitionImpl vdi = (VariableDefinitionImpl) content;  
        System.out.println("Belső változó" + vdi.getName());  
        var.add(vdi.getName());  
    }  
}
```

Ez a függvény amit írtam:

```
    static String print(List<String> var)  
    {  
        String printedText = null;  
        printedText += "public static void print(IExampleStatemachine s) {";  
        for(int i=0; i<var.size(); i++)  
        {  
            printedText += "System.out.println(\"W = \" + s.getSCInterface().get" + var.get(i) + "().";  
        }  
        printedText += "};";  
        return printedText;  
    }  
}
```

4.5 feladat:

Nem igazán értettem a feladatot és idő szűkében kihagytam.

4.6 feladat:

Ez a feladat sajnos a 4.5-re épült. 😞

4.7 feladat:

Ez a feladat is a 4.5-re épült. 😞