A Student Project: Petri Nets according to the book of Prof. Wolfgang Reisig

OPEN | MODEL
Initiative

www.openmodels.at

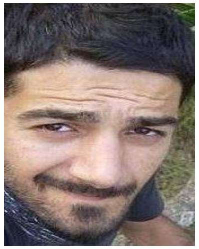
o.Univ.-Prof. Dr. Dimitris Karagiannis

A Student Project

- **Curriculum:** Master of Business Informatics at the University of Vienna, elective module ‘Large Scale Information System: Hybrid Method Engineering‘
- **Prerequisite:** Attended the course ‘Metamodelling‘
- **Task:** Implementation of a Petrinets modelling method on ADO*xx*® focusing on specific analysis mechanisms according to the book ‘Petrinetze‘ of Prof. Wolfgang Reisig
- **Project Team:**

Nesat Efendioglu (presenting today)

Christos Lekaditis (presenting today)



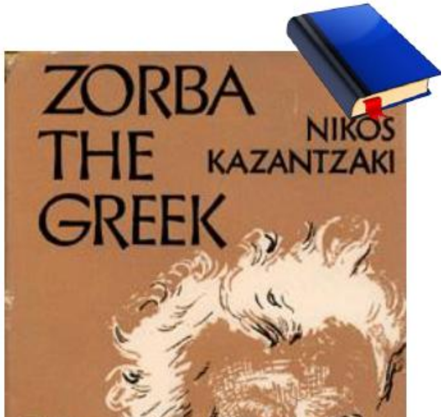
Cihan Celik

Zhan Chen

Supervised by: Alexander Bergmayr, Sandra Hintringer & Niksa Visic

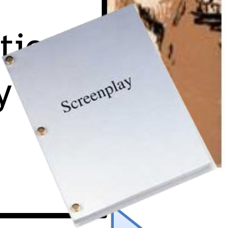
A Metaphor for Conceptualization
“From Book to Movie”

Acting

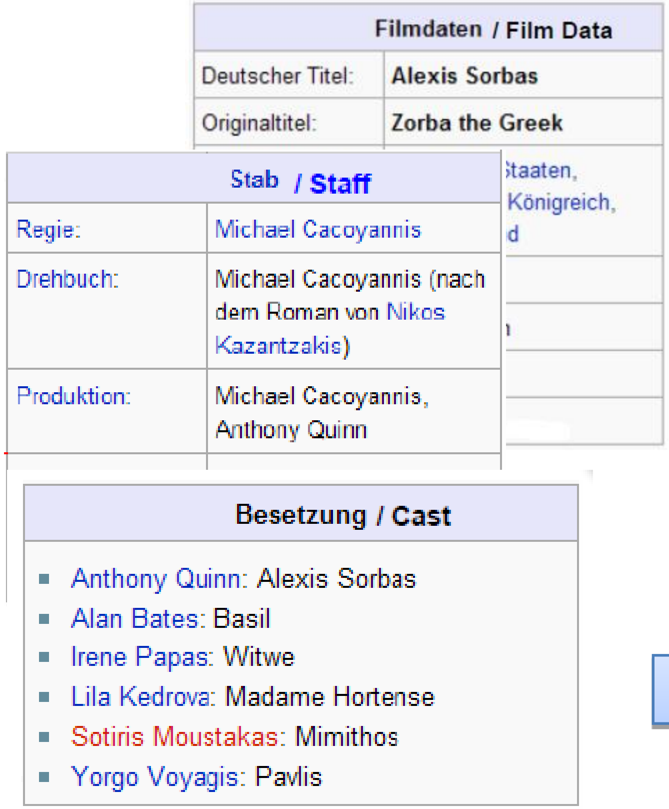


Screen Play

- Dialogs
- Interacti
- Scenery
- Effects
- …



Writing



Filmdaten / Film Data	
Deutscher Titel:	Alexis Sorbas
Originaltitel:	Zorba the Greek
	taaten, Königreich, d
	ı

Stab / Staff	
Regie:	Michael Cacoyannis
Drehbuch:	Michael Cacoyannis (nach dem Roman von Nikos Kazantzakis)
Produktion:	Michael Cacoyannis, Anthony Quinn

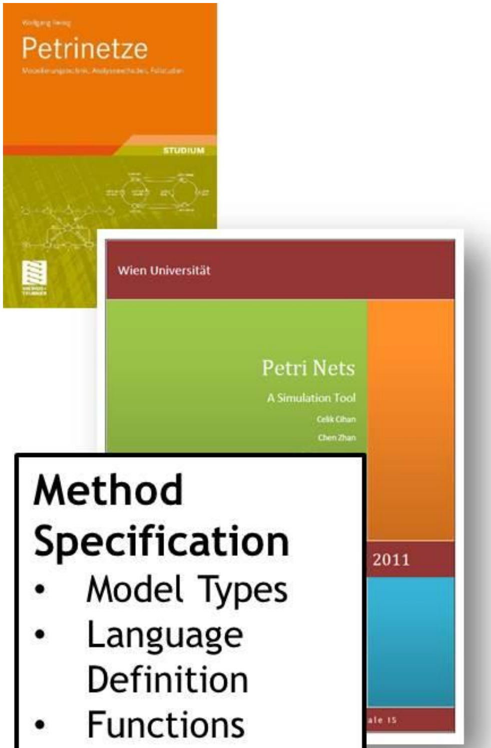
Besetzung / Cast

- Anthony Quinn: Alexis Sorbas
- Alan Bates: Basil
- Irene Papas: Witwe
- Lila Kedrova: Madame Hortense
- Sotiris Moustakas: Mimithos
- Yorgo Voyagis: Pavlis

Cutting

A Metaphor for Conceptualization of a Modelling Method *"From Book to Tool"*

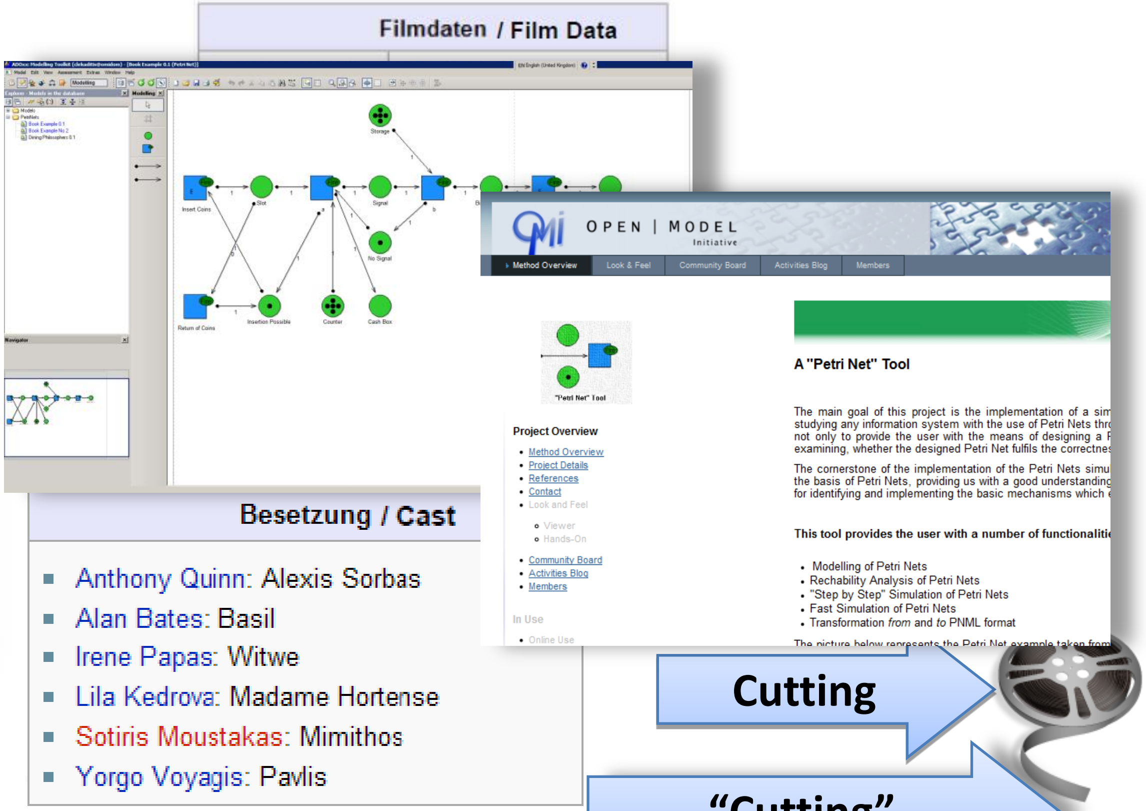
"Acting"



Method Specification

- Model Types
- Language Definition
- Functions
- User Behaviour
- …

"Writing"



Filmdaten / Film Data

Besetzung / Cast

- Anthony Quinn: Alexis Sorbas
- Alan Bates: Basil
- Irene Papas: Witwe
- Lila Kedrova: Madame Hortense
- Sotiris Moustakas: Mimithos
- Yorgo Voyagis: Pavlis

Project Overview

- Method Overview
- Project Details
- References
- Contact
- Look and Feel
  - Viewer
  - Hands-On
- Community Board
- Activities Blog
- Members

In Use

- Online Use

A "Petri Net" Tool

The main goal of this project is the implementation of a sim studying any information system with the use of Petri Nets thr not only to provide the user with the means of designing a P examining, whether the designed Petri Net fulfils the correctnes

The cornerstone of the implementation of the Petri Nets simu the basis of Petri Nets, providing us with a good understanding for identifying and implementing the basic mechanisms which

This tool provides the user with a number of functionaliti

- Modelling of Petri Nets
- Rechability Analysis of Petri Nets
- "Step by Step" Simulation of Petri Nets
- Fast Simulation of Petri Nets
- Transformation *from* and *to* PNML format

The picture below represents the Petri Net example taken from

Cutting

"Cutting"

universität wien

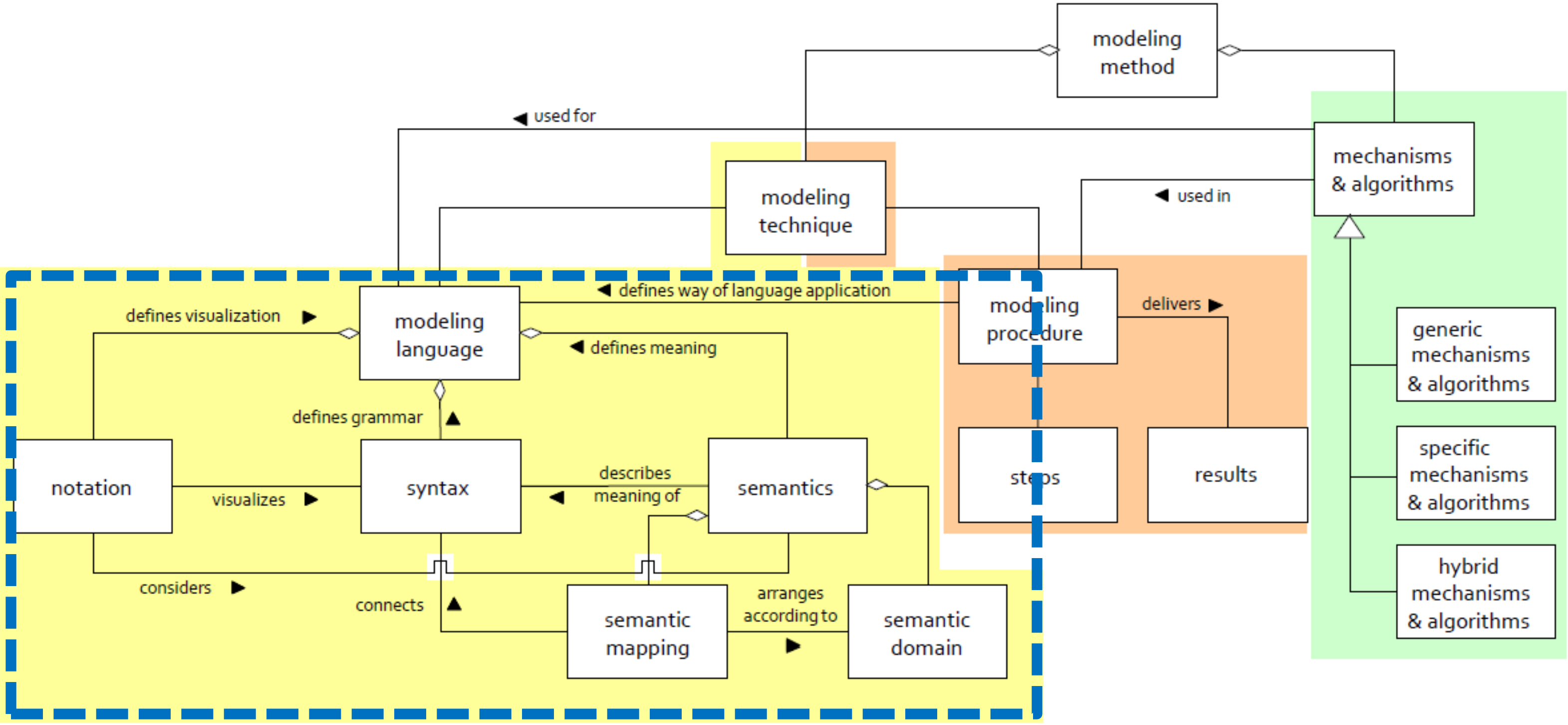
DKE

o.Univ.-Prof. Dr. Dimitris Karagiannis

OPEN | MODEL Initiative

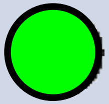
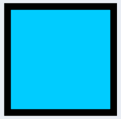
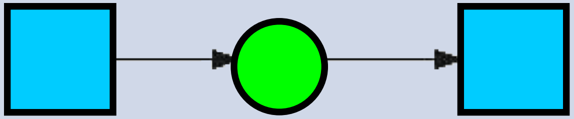
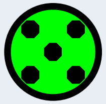
www.openmodels.at

“Writing”- Metamodel Framework - Modelling Language

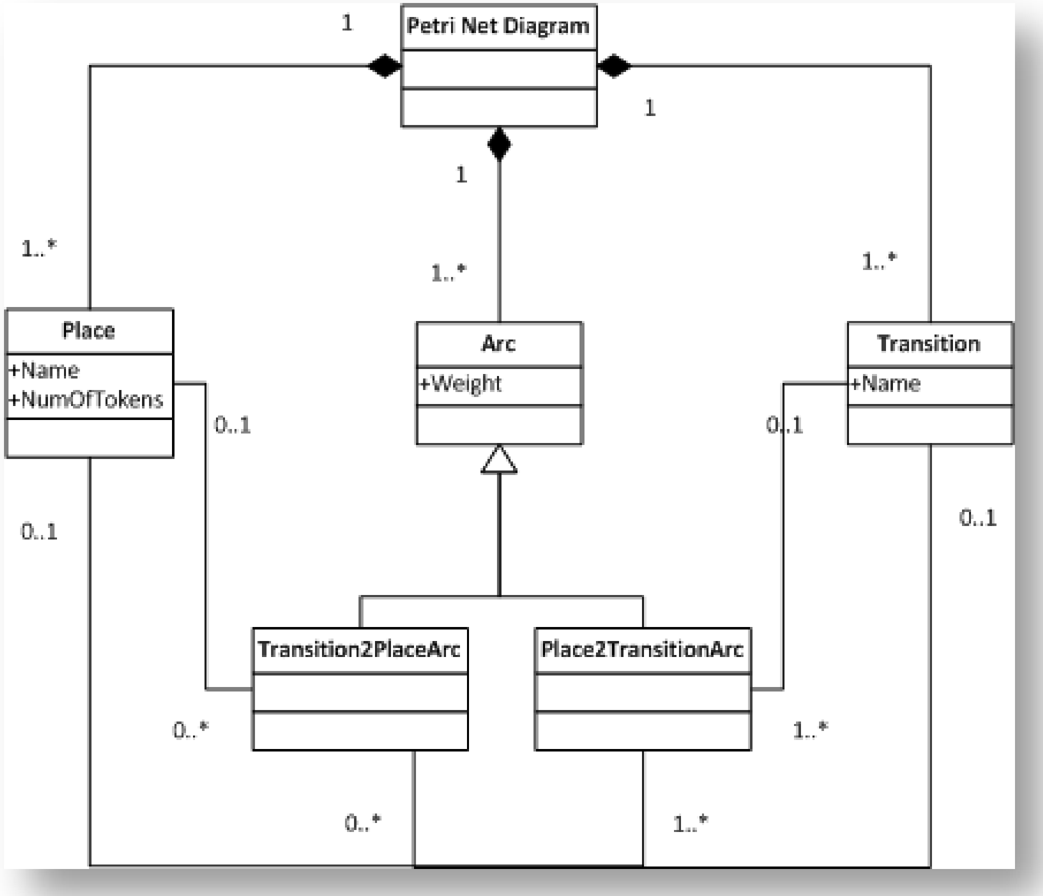


cf. Karagiannis & Kühn (2002)

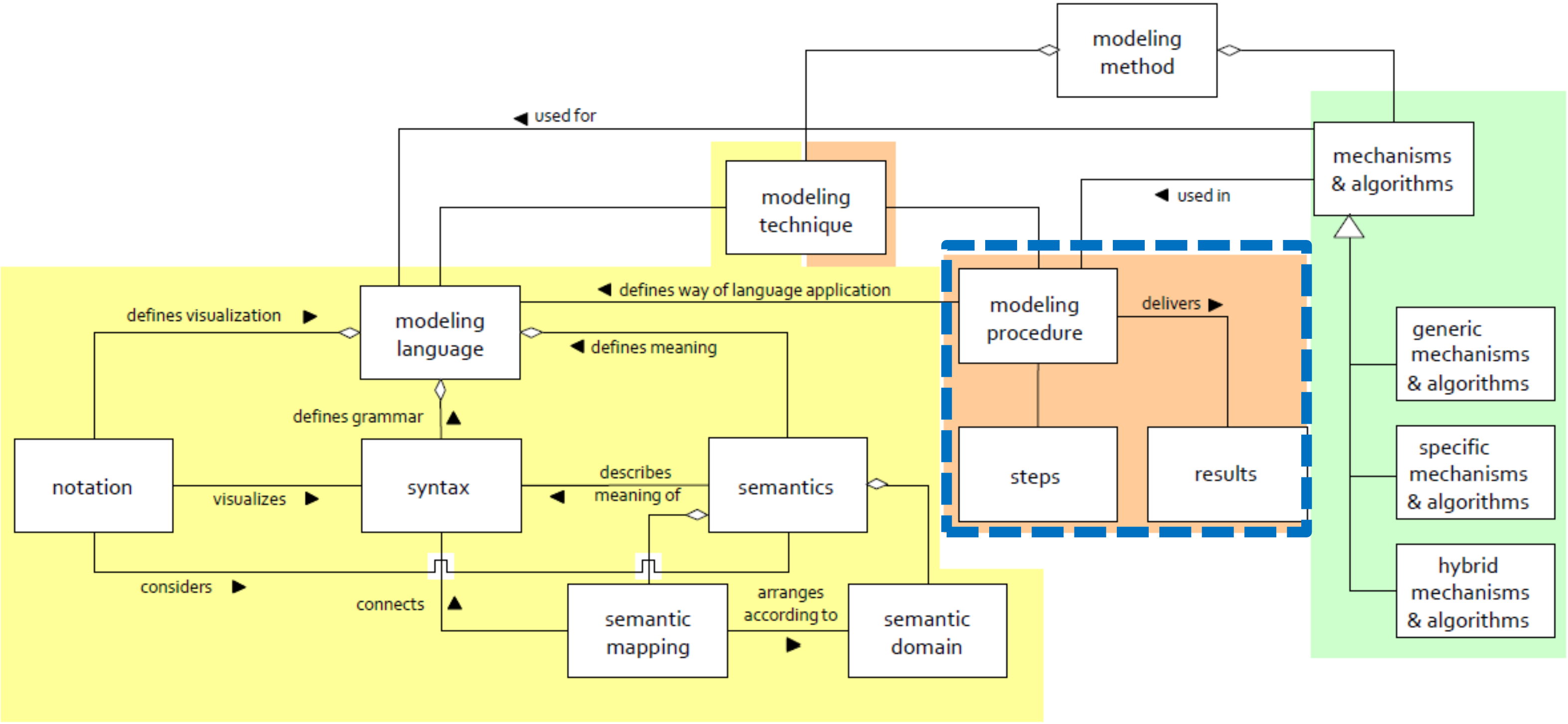
“Writing”- Metamodel Framework Modelling Language - Semantics / Notation

Petri Net: Main Elements	Semantics	Notation
Place	Represents passive components	
Transition	Represents active components	
Arc	Represents relations among components	
Marking	Refers to the distribution of tokens among the places of the Petri Net	

“Writing”- Metamodel Framework
Modelling Language - Syntax



“Writing”- Metamodel Framework - Modelling Procedure



cf. Karagiannis & Kühn (2002)

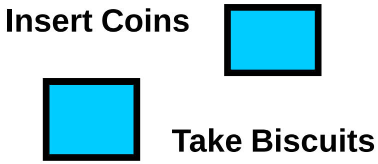
“Writing” - Metamodel Framework - Modelling Procedure

Define the System

- Automatic Biscuit Seller

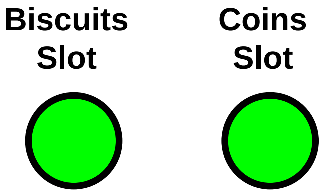
Define Active States

- Insert Coins
- Take Biscuits
- …



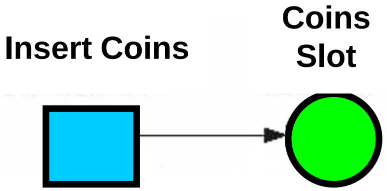
Define Passive States

- Coins Slot
- Biscuits Storage
- Biscuits Slot
- …



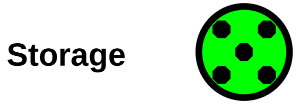
Define Relations

- Insert coins → Coins Slot
- Biscuits Slot → Take Biscuits
- …



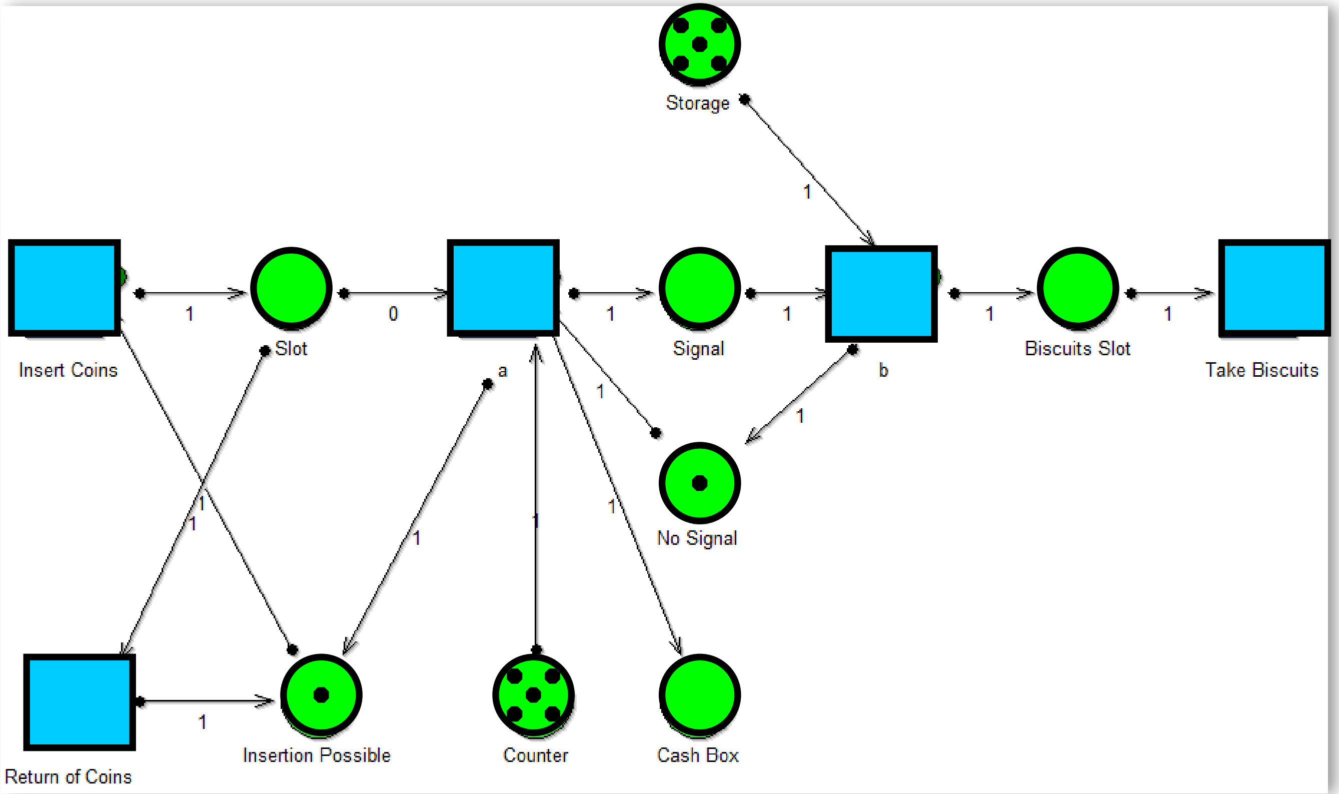
Define Initial Resource Distribution

- Storage: 5 Biscuits
- …



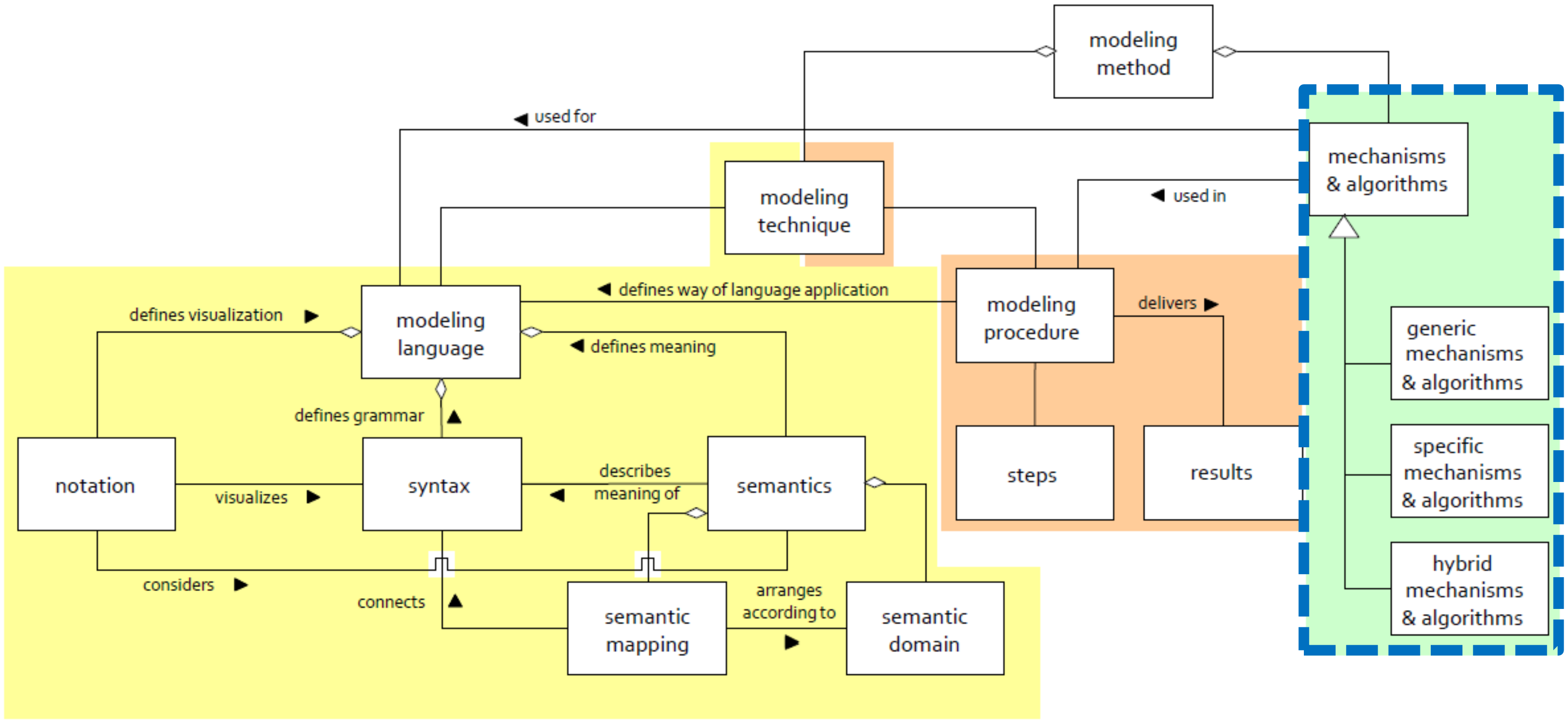
“Writing” - Metamodel Framework - Modelling Procedure

Modelling Procedure Result



Petri Net of Automatic Biscuit Seller from Prof. Reisig’s Book

“Writing” - Metamodel Framework - Mechanisms & Algorithms



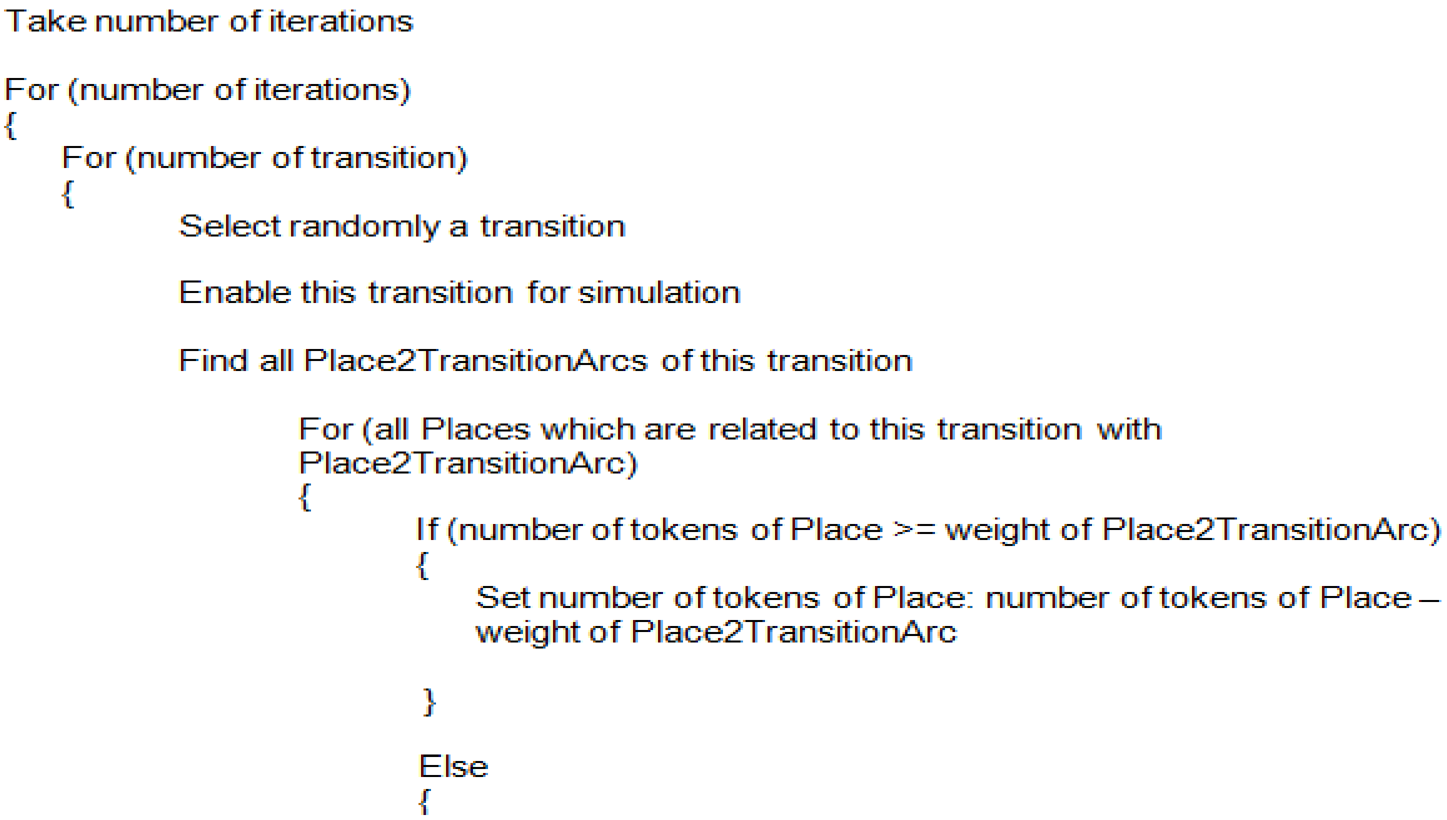
cf. Karagiannis & Kühn (2002)

“Writing” - Metamodel Framework - Mechanisms & Algorithms

- **Simulation:** execution of a Petri Nets (consumption /production of Tokens)
  - **Fast Simulation**
  - **Step by Step Simulation**
- **Net Statistics:** calculation of number of places, transitions and arcs that the Petri Net consists of
- **Transformation/Import/Export:** transformation of the ADO XML format file to PNML format and vice versa
- **Reachability Analysis(in progress)**: checks whether requested Marking is reachable from the initial Marking

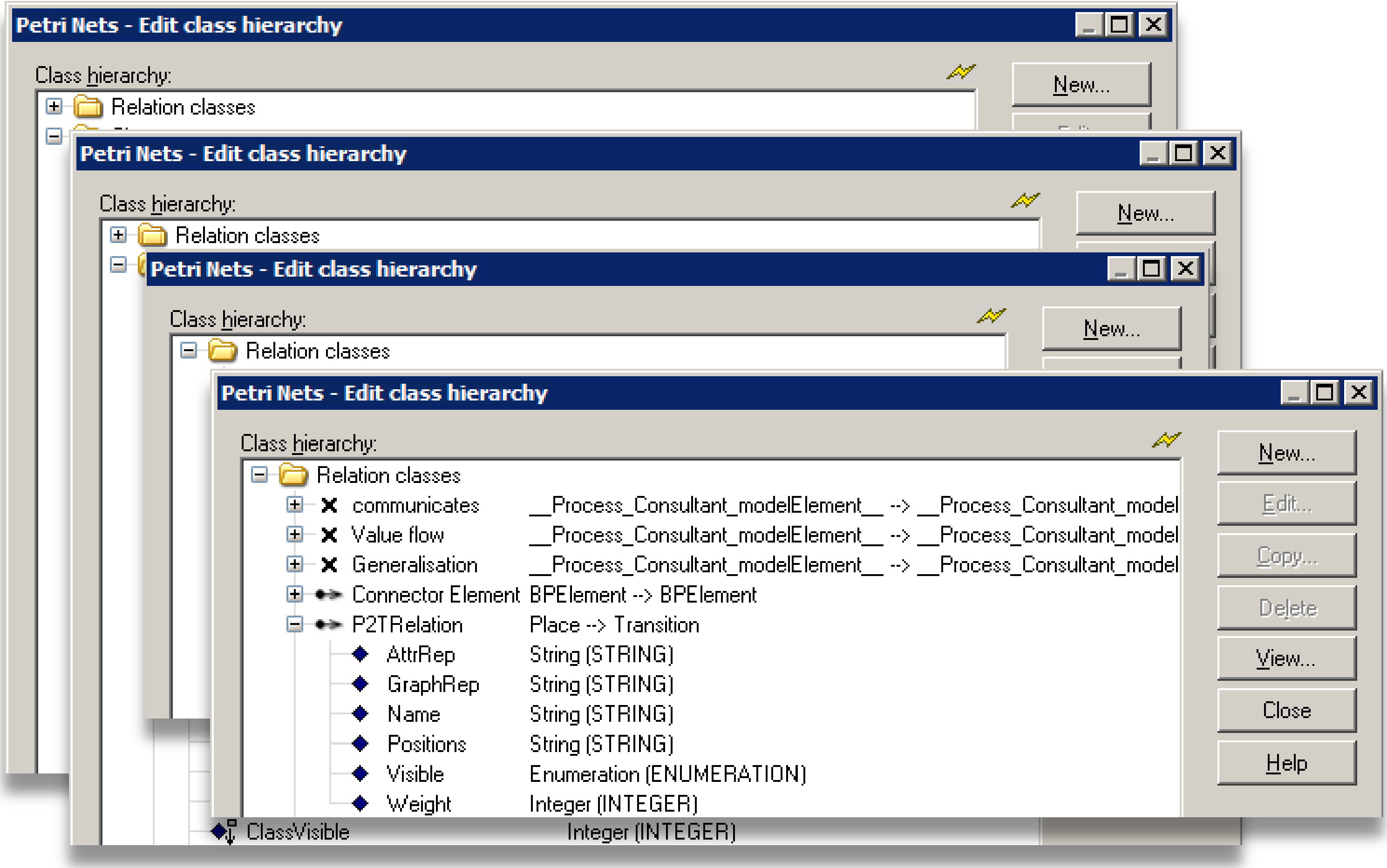
“Writing” - Metamodel Framework - Mechanisms & Algorithms

Fast simulation pseudo code:



```
Take number of iterations

For (number of iterations)
{
    For (number of transition)
    {
            Select randomly a transition

            Enable this transition for simulation

            Find all Place2TransitionArcs of this transition

                    For (all Places which are related to this transition with
                    Place2TransitionArc)
                    {
                            If (number of tokens of Place >= weight of Place2TransitionArc)
                            {
                                Set number of tokens of Place: number of tokens of Place –
                                weight of Place2TransitionArc

                             }

                            Else
                            {
```

“Acting” - ADOxx Specific Implementation - Semantic



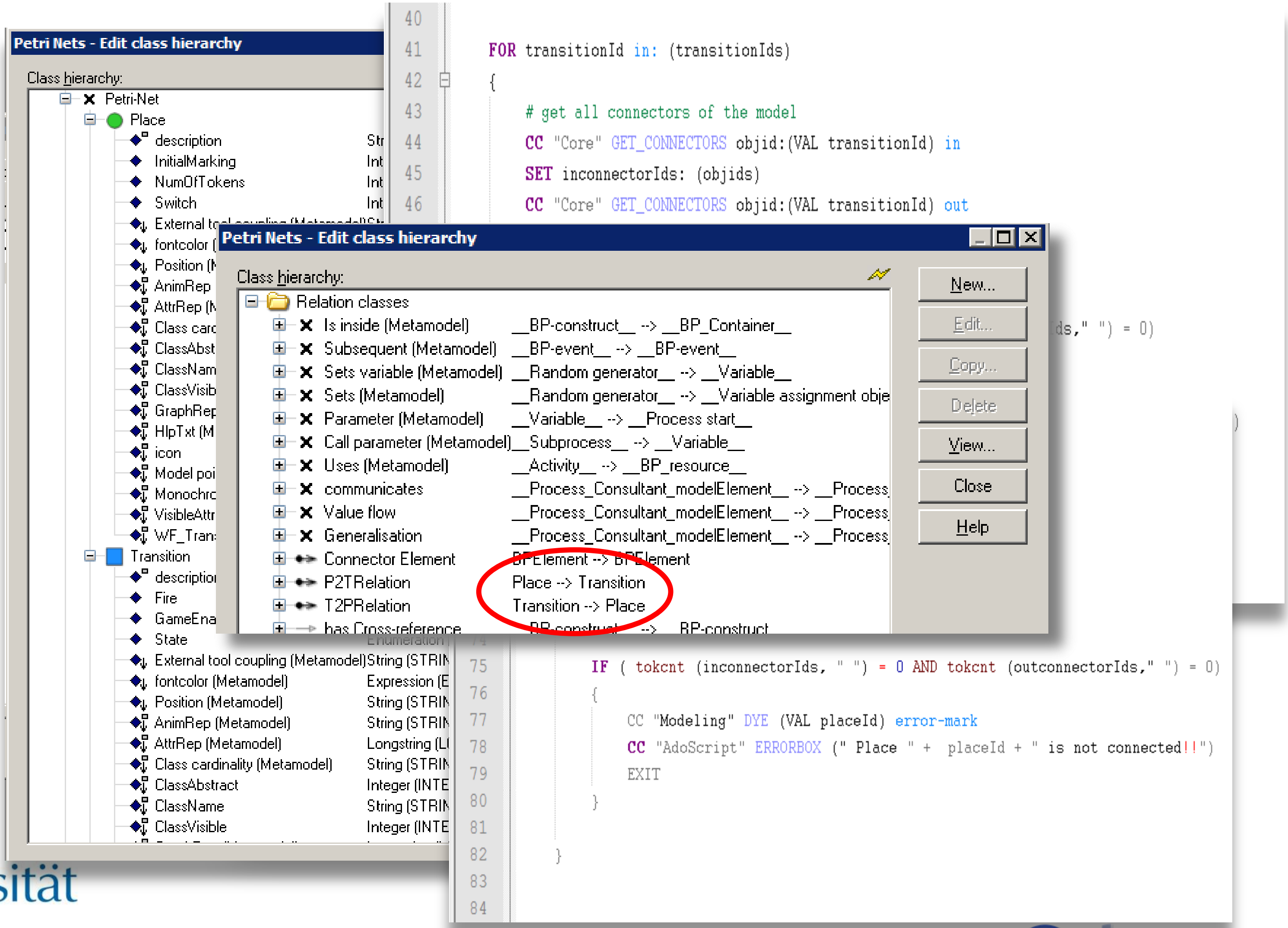
Petri Nets - Edit class hierarchy

Class hierarchy:

- Relation classes
  - communicates __Process_Consultant_modelElement__ --> __Process_Consultant_model
  - Value flow __Process_Consultant_modelElement__ --> __Process_Consultant_model
  - Generalisation __Process_Consultant_modelElement__ --> __Process_Consultant_model
  - Connector Element BPElement --> BPElement
  - P2TRelation Place --> Transition
    - AttrRep String (STRING)
    - GraphRep String (STRING)
    - Name String (STRING)
    - Positions String (STRING)
    - Visible Enumeration (ENUMERATION)
    - Weight Integer (INTEGER)

ClassVisible Integer (INTEGER)

“Acting” - ADOxx Specific Implementation-Syntax



Petri Nets - Edit class hierarchy

Class hierarchy:

- Petri-Net
  - Place
    - description Str
    - InitialMarking Int
    - NumOfTokens Int
    - Switch Int
    - External tool coupling (Metamodel)Str
    - fontcolor (
    - Position (M
    - AnimRep
    - AttrRep (M
    - Class card
    - ClassAbst
    - ClassNam
    - ClassVisib
    - GraphRep
    - HlpTxt (M
    - icon
    - Model poi
    - Monochro
    - VisibleAttr
    - WF_Tran
  - Transition
    - description
    - Fire
    - GameEna
    - State Enumeration
    - External tool coupling (Metamodel) String (STRIN
    - fontcolor (Metamodel) Expression (E
    - Position (Metamodel) String (STRIN
    - AnimRep (Metamodel) String (STRIN
    - AttrRep (Metamodel) Longstring (LI
    - Class cardinality (Metamodel) String (STRIN
    - ClassAbstract Integer (INTE
    - ClassName String (STRIN
    - ClassVisible Integer (INTE

```
FOR transitionId in: (transitionIds)
{
    # get all connectors of the model
    CC "Core" GET_CONNECTORS objid:(VAL transitionId) in
    SET inconnectorIds: (objids)
    CC "Core" GET_CONNECTORS objid:(VAL transitionId) out
```

```
ds," ") = 0)
)
```

Petri Nets - Edit class hierarchy

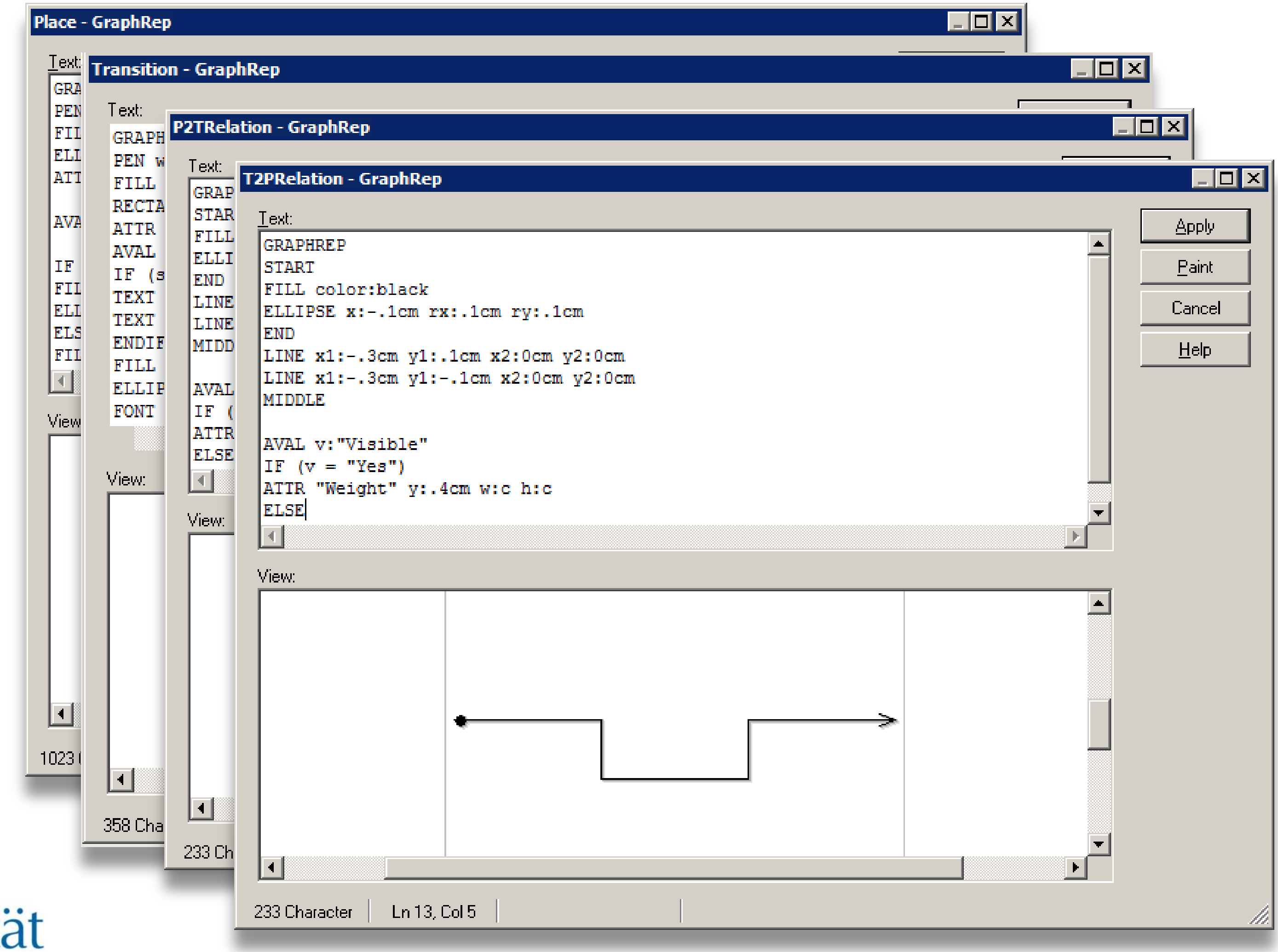
Class hierarchy:

- Relation classes

Is inside (Metamodel)	__BP-construct__ --> __BP_Container__
Subsequent (Metamodel)	__BP-event__ --> __BP-event__
Sets variable (Metamodel)	__Random generator__ --> __Variable__
Sets (Metamodel)	__Random generator__ --> __Variable assignment obje
Parameter (Metamodel)	__Variable__ --> __Process start__
Call parameter (Metamodel)	__Subprocess__ --> __Variable__
Uses (Metamodel)	__Activity__ --> __BP_resource__
communicates	__Process_Consultant_modelElement__ --> __Process
Value flow	__Process_Consultant_modelElement__ --> __Process
Generalisation	__Process_Consultant_modelElement__ --> __Process
Connector Element	BPElement --> BPElement
P2TRelation	Place --> Transition
T2PRelation	Transition --> Place
has Cross-reference	__BP-construct__ --> __BP-construct

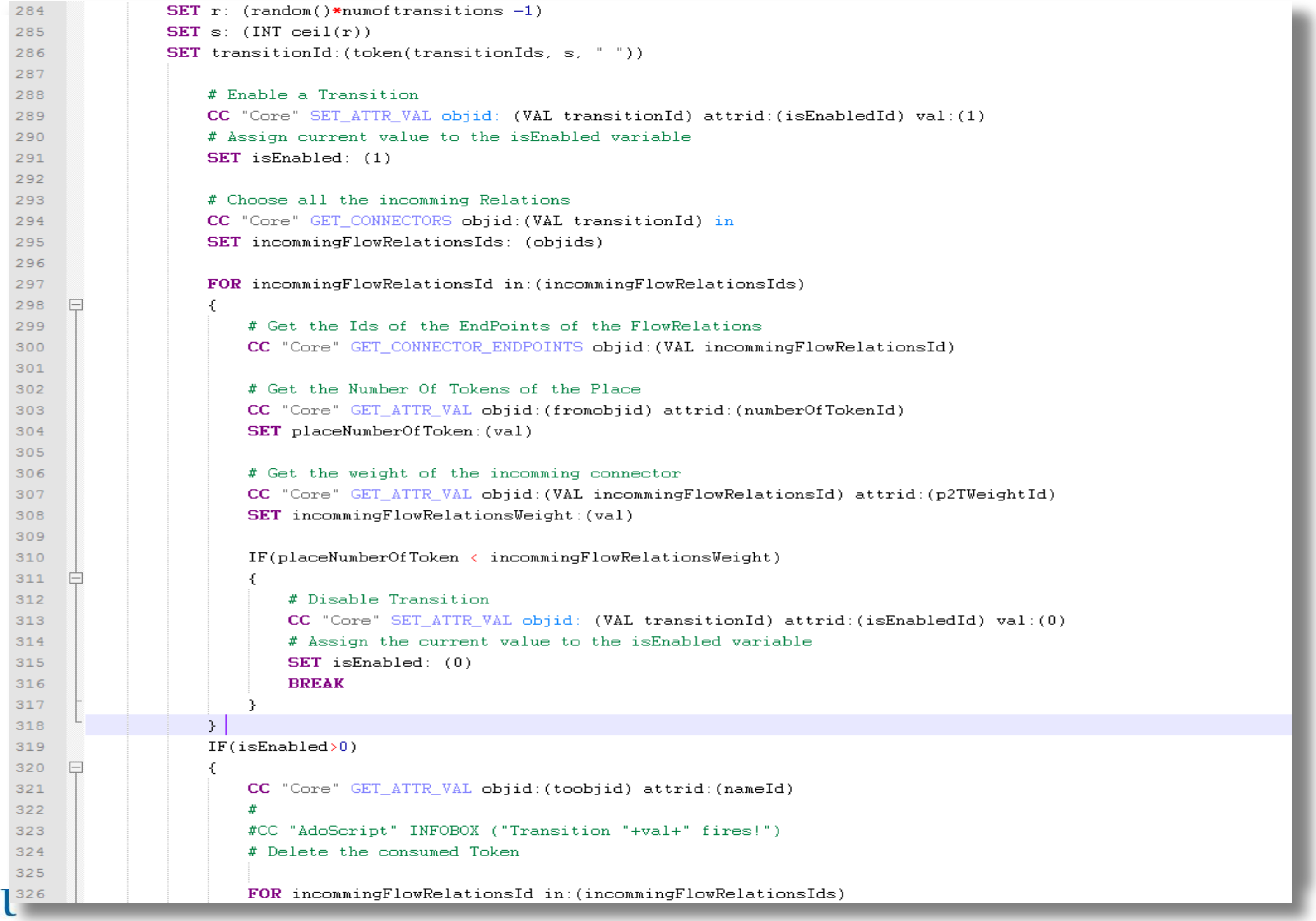
```
        IF ( tokcnt (inconnectorIds, " ") = 0 AND tokcnt (outconnectorIds," ") = 0)
        {
            CC "Modeling" DYE (VAL placeId) error-mark
            CC "AdoScript" ERRORBOX (" Place " + placeId + " is not connected!!")
            EXIT
        }

    }
```

“Acting” - ADOxx Specific Implementation-Notation



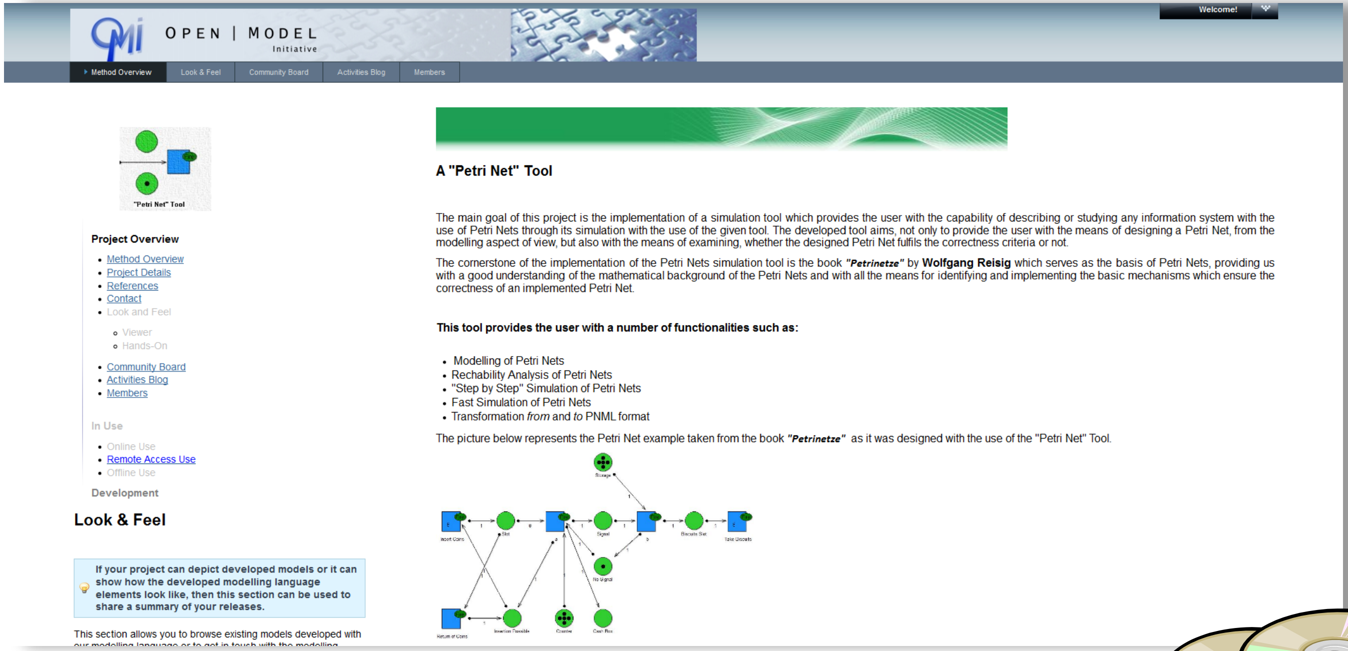
```
GRAPHREP
START
FILL color:black
ELLIPSE x:-.1cm rx:.1cm ry:.1cm
END
LINE x1:-.3cm y1:.1cm x2:0cm y2:0cm
LINE x1:-.3cm y1:-.1cm x2:0cm y2:0cm
MIDDLE

AVAL v:"Visible"
IF (v = "Yes")
ATTR "Weight" y:.4cm w:c h:c
ELSE
```

“Acting” - ADOxx Specific Implementation-Mechanisms & Algorithms



```
SET r: (random()*numoftransitions -1)
SET s: (INT ceil(r))
SET transitionId:(token(transitionIds, s, " "))

    # Enable a Transition
    CC "Core" SET_ATTR_VAL objid: (VAL transitionId) attrid:(isEnabledId) val:(1)
    # Assign current value to the isEnabled variable
    SET isEnabled: (1)

    # Choose all the incomming Relations
    CC "Core" GET_CONNECTORS objid:(VAL transitionId) in
    SET incommingFlowRelationsIds: (objids)

    FOR incommingFlowRelationsId in:(incommingFlowRelationsIds)
    {
        # Get the Ids of the EndPoints of the FlowRelations
        CC "Core" GET_CONNECTOR_ENDPOINTS objid:(VAL incommingFlowRelationsId)

        # Get the Number Of Tokens of the Place
        CC "Core" GET_ATTR_VAL objid:(fromobjid) attrid:(numberOfTokenId)
        SET placeNumberOfToken:(val)

        # Get the weight of the incomming connector
        CC "Core" GET_ATTR_VAL objid:(VAL incommingFlowRelationsId) attrid:(p2TWeightId)
        SET incommingFlowRelationsWeight:(val)

        IF(placeNumberOfToken < incommingFlowRelationsWeight)
        {
            # Disable Transition
            CC "Core" SET_ATTR_VAL objid: (VAL transitionId) attrid:(isEnabledId) val:(0)
            # Assign the current value to the isEnabled variable
            SET isEnabled: (0)
            BREAK
        }
    }
    IF(isEnabled>0)
    {
        CC "Core" GET_ATTR_VAL objid:(toobjid) attrid:(nameId)
        #
        #CC "AdoScript" INFOBOX ("Transition "+val+" fires!")
        # Delete the consumed Token

        FOR incommingFlowRelationsId in:(incommingFlowRelationsIds)
```

“Cutting”



Project Overview

- Method Overview
- Project Details
- References
- Contact
- Look and Feel
  - Viewer
  - Hands-On
- Community Board
- Activities Blog
- Members

In Use

- Online Use
- Remote Access Use
- Offline Use

Development

Look & Feel

If your project can depict developed models or it can show how the developed modelling language elements look like, then this section can be used to share a summary of your releases.

This section allows you to browse existing models developed with our modelling language or to get in touch with the modelling

A "Petri Net" Tool

The main goal of this project is the implementation of a simulation tool which provides the user with the capability of describing or studying any information system with the use of Petri Nets through its simulation with the use of the given tool. The developed tool aims, not only to provide the user with the means of designing a Petri Net, from the modelling aspect of view, but also with the means of examining, whether the designed Petri Net fulfils the correctness criteria or not.

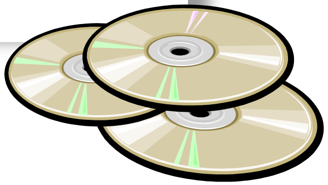
The cornerstone of the implementation of the Petri Nets simulation tool is the book ***"Petrinetze"*** by **Wolfgang Reisig** which serves as the basis of Petri Nets, providing us with a good understanding of the mathematical background of the Petri Nets and with all the means for identifying and implementing the basic mechanisms which ensure the correctness of an implemented Petri Net.

This tool provides the user with a number of functionalities such as:

- Modelling of Petri Nets
- Rechability Analysis of Petri Nets
- "Step by Step" Simulation of Petri Nets
- Fast Simulation of Petri Nets
- Transformation *from* and *to* PNML format

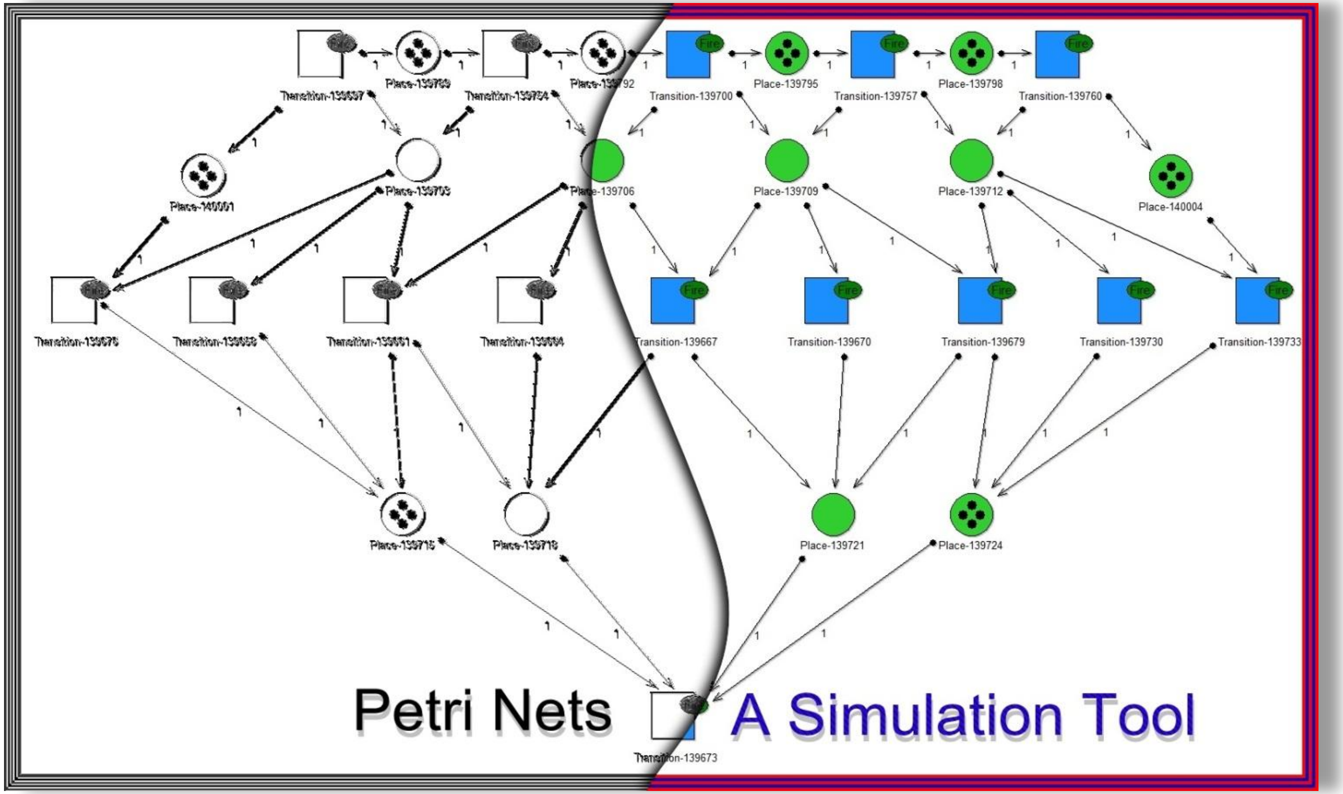
The picture below represents the Petri Net example taken from the book ***"Petrinetze"*** as it was designed with the use of the "Petri Net" Tool.

OMI Website



Petrinets on CD, as download & project website on OMI

o.Univ.-Prof. Dr. Dimitris Karagiannis


Petri Nets
A Simulation Tool
Fire
Transition-139700
Place-139795
Transition-139757
Place-139798
Transition-139760
Place-139706
Place-139709
Place-139712
Place-140004
Transition-139667
Transition-139670
Transition-139679
Transition-139730
Transition-139733
Place-139721
Place-139724
Transition-139673
1

Live Demo

Thank you for your attention!

For any further information please contact

nesatefendioglu@gmail.com
chris.lekaditis@gmail.com