Received: May 13, 2018

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

SBGN2HFPN Transformation of SBGN-PD into Petri Nets Illustrated on the Glycolysis Pathway

Safae Cherdal[1]* **Salma Mouline**[1] **Souad Amghar**[1]

[1]*Research Laboratory in Computer Science and Telecommunications (LRIT)-*
CNRST Associate Research Unit (URAC 29), Rabat IT Center, Faculty of Sciences,
Mohammed V University in Rabat, Morocco
* Corresponding author's Email: safaecherdal@gmail.com

Abstract: Systems Biology Graphical Notation (SBGN) is a standard graphical language for representing biological and biochemical processes and interactions. Even that SBGN Process Description-an SBGN sub-language- facilitates biological systems representation, it allows only qualitative descriptions of metabolic pathways and does not provide quantitative analytic environment. However such descriptions are essential for behavioural analysis. This analysis can be possible using computational formalisms such as Hybrid Functional Petri Net (HFPN) which is a Petri net extension dedicated to study and verify biopathways. However, biologists use generally graphical notations such as SBGN. To address this paradox, we propose, in this paper, an SBGN-PD to HFPN transformation based on Model Driven Engineering (MDE). Furthermore, we illustrate this transformation on the Glycolysis pathway and verify the transformation resulting HFPN model by comparing it with the existing HFPN model through simulation.

Keywords: Model driven engineering (MDE), Model-to-model transformation, Meta-modelling, HFPN, SBGN, SBGN process description diagram, Glycolysis pathway.

1. Introduction

1.1 Context

Biologists and biochemists are constantly in search for methods and strategies that can help them in the understanding of life complexity.

Biochemical networks represent a large complex connectivity between reactions occurring in living cells. They represent the abstract description of one or more complex metabolic pathway. Dynamic properties of these networks determine not only the physiology of cells but also the development and behaviour of the organism. Being dynamic and complex, their behaviour may be hard to analyse and to predict.

A wide variety of strategies has been developed to represent such networks, ranging from textual representations through chemical reactions or through XML-based standards like SBML (Systems Biology Markup Language) [1] to graphical notations such as SBGN (Systems Biology Graphical Notation) [2]. Graphical maps of biochemical reaction networks are proving to be powerful tools for facilitating an overview of the interactions of particular molecules. SBGN is a standard for drawing such reaction networks aiming to provide biologists and biochemists with an easily understandable description of these networks. The SBGN Process Description (SBGN-PD), an SBGN sub-language, is a visual language with a precise grammar that builds on an underlying abstraction as the basis of its semantics [2]. The graphical nature of SBGN-PD allows qualitative descriptions of metabolic pathways. However, such descriptions, need the inclusion of the corresponding mathematical details such as parameters and equations for computing the rates and velocities of the occurring reactions in cells.

From a computational point of view, in order to address this limit, computer modelling approaches, simulations and tools are proposed to model and analyse biochemical networks [3]. Among the proposed formal specifications, there are Petri nets. Petri nets are popular and used to model and simulate this kind of systems since they allow intuitive graphical representation of biochemical systems, help revealing the logic behind these complex systems, offer specific analysis, namely, static, qualitative and quantitative behavioural analysis. Furthermore, they are supported by a wide range of simulation and analysis tools and offer several extensions such as Hybrid Functional Petri Net [4] which is a very useful and promising Petri net extension for biopathways expression [5].

1.2 Problems & issues

In the context of studying biochemical networks, it is interesting to use formal languages for modelling and simulation, nevertheless, biologists do not dispose necessarily knowledge in this field, and generally prefer to use dedicated graphical notations. However, visual languages such as Systems Biology Graphical Notation (SBGN) [2] do not provide any kind of quantitative analytic environment. Although to address these limitations, some SBGN translations have been proposed such as SBGN process description diagram (SBGN-PD) to SBGN Activity flow diagram (SBGN-AF) in [6] and [7], SBGN-PD to a textual representation in [8], also SBGN-PD is translated to BioPEPA through SBGNtext2BioPEPA in [9, 10]. However, no transformation from SBGN-PD to formal language is proposed. We choose to use Petri net as a formal specification language especially HFPN [4] extension. Thus, an SBGN-PD to HFPN transformation might be a solution allowing to analyse SBGN-PD models representing metabolic pathways not only through simulation but also by formal analysis.

1.3 Contribution & contents

We use the Model Driven Engineering (MDE) approach to transform an SBGN-PD model into an HFPN model. MDE [11] is a software development methodology providing tools, concepts and languages to facilitate creation and transformation models based on meta-models and transformation rules.

In this paper, our proposal is therefore divided into two large stages. The first one consists on defining meta-models corresponding to SBGN-PD and HFPN languages including their OCL well-completeness constraints. The second one, consists on defining and implementing transformation rules allowing the mapping of SBGN-PD models to HFPN models.

This paper is organized as follows: Section 2 introduces briefly the graphical notation SBGN in general and SBGN-PD particularly, and the Petri net extension HFPN. Also, we describe their corresponding meta-models in this section. In section 3 the SBGN2HFPN transformation rules and their implementation are given. In Section 4, the SBGN2HFPN transformation is illustrated on the Glycolysis Pathway. Finally, we end by reviewing related works and providing some perspectives for further developments.

2. Definitions and meta-models

In the context of MDE, several model transformation approaches are proposed, such as programming approach, modelling approach and others. Selecting the approach to use, is related to the transformation application domain.

We choose to use the modelling approach, in which the transformation is a process based on transformation rules describing the mapping between metadatas of the source meta-model and those of the target meta-model.

In this section, we explain how to use the semantics of SBGN-PD to map an SBGN-PD model to an HFPN model for quantitative analysis. We introduce briefly the source and target languages, namely, SBGN especially SBGN-PD, and HFPN and we define their corresponding meta-models in order to perform the model-to-model transformation.

2.1 SBGN: systems biology graphical notation

SBGN [12] is a visual language which focuses on the graphical representation of biochemical and cellular processes. It provides comprehensive sets of symbols with precise semantics to represent interactions and regulations between molecular species, such as protein modification. It also defines detailed syntactic rules defining the use of those symbols. Besides, it describes the manner in which graphical informations should be interpreted [2].

It should be noted that semantics of SBGN diagrams does not depend on the relative position of the symbols, or on colours, patterns, shades, shapes and thickness of edges [12].

Actually, SBGN consists of three complementary sublanguages: process description diagram (SBGN-PD), entity relationship diagram (SBGN-ER), and activity flow diagram (SBGN-AF). Together the different notations enable scientists to represent biological networks in a standard and unambiguous way.

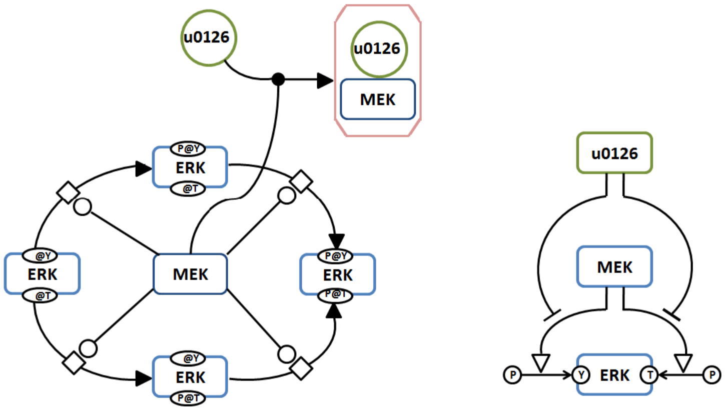


Figure. 1 SBGN models representing an example of protein phosphorylation inspired by [12]. SBGN-PD (Left), SBGN-ER (Center) and SBGN-AF (Right)

Fig. 1 inspired by [12] and [13], represents a simple example of protein phosphorylation (ERK) catalysed by an enzyme (MEK) and modulated by an inhibitor (u0126).

The given example illustrates the SBGN three sublanguages such as:

- Process Description diagram (SBGN-PD), displays explicitly all the four forms of ERK, phosphorylated and non-phosphorylated on the tyrosine (Y) and the threonine (T). It also represents the processes of phosphorylation by MEK and the inhibition of MEK by forming a complex with u0126.
- Entity Relationship diagram (SBGN-ER), shows ERK protein only once, then the different possible states are not depicted. It shows ERK and the assignment of its phosphorylations (at the tyrosine and threonine residues), as well as the relationships between those, MEK and u0126.
- Activity Flow diagram (SBGN-AF) depicts, in an abstract way, the ERK activation by MEK and the inhibition of MEK by u0126.

The three sub-languages correspond to three different levels of abstraction of the same biological system. Generally, we note that SBGN-PD represents transitions of entities from one form or state to another. The SBGN-ER, represents the influences of entities upon the behaviour of others. However, SBGN-AF represents the activity flow from one entity to another within the same entity.

We consider that SBGN-PD is the most explicit among the three SBGN sub-languages, and it is the most suitable for metabolic pathways modelling [6, 14] since it gives a high level of representation details. Furthermore, it is the most used and it is supported by several tools (e.g. COPASI [15], BIOCHAM [16], CellDesigner [17] ....) unlike the other SBGN sublanguages [18].

In the rest of this paper, our work is based on SBGN-PD sub-language as a source model of the proposed transformation.

2.2 SBGN-PD meta-model

SBGN-PD offers several symbols (See Fig. 2), to represent different molecules (protein, gene, receivers ...) and reactions (association, translocation[1] ...).

Process diagram glyphs description given in [2] shown in Fig. 2 are as follows :

- **An Entity Pool** is a set of entities that cannot be distinguished from each other, since it is formed by all the molecular entities fulfilling the same role in a given process. It can represent different granularity levels, such as all the proteins or all the instances of a given protein. Entity pool nodes refer to *unspecified entity, simple chemical, macromolecule, nucleic acid feature, perturbing agent, source sink, multimers and complex.*
- **Process Nodes** include processes allowing transformation of one or several entity pools into one or several entity pools, which can be identical or different. Process nodes refer to *process, omitted process, uncertain process, association, dissociation* and *phenotype.*
- **Connecting Arcs** are lines that connect nodes of entities and process nodes together. The symbols attached at their ends indicate their semantics. Arcs are *consumption arc, production arc, modulation arc, stimulation arc, catalysis arc, inhibition arc, necessary stimulation arc, logic arc* and *equivalence arc.*

[1] Translocation is a genetic mutation characterized by the reciprocal chromosomal material between non-homologous chromosomes.

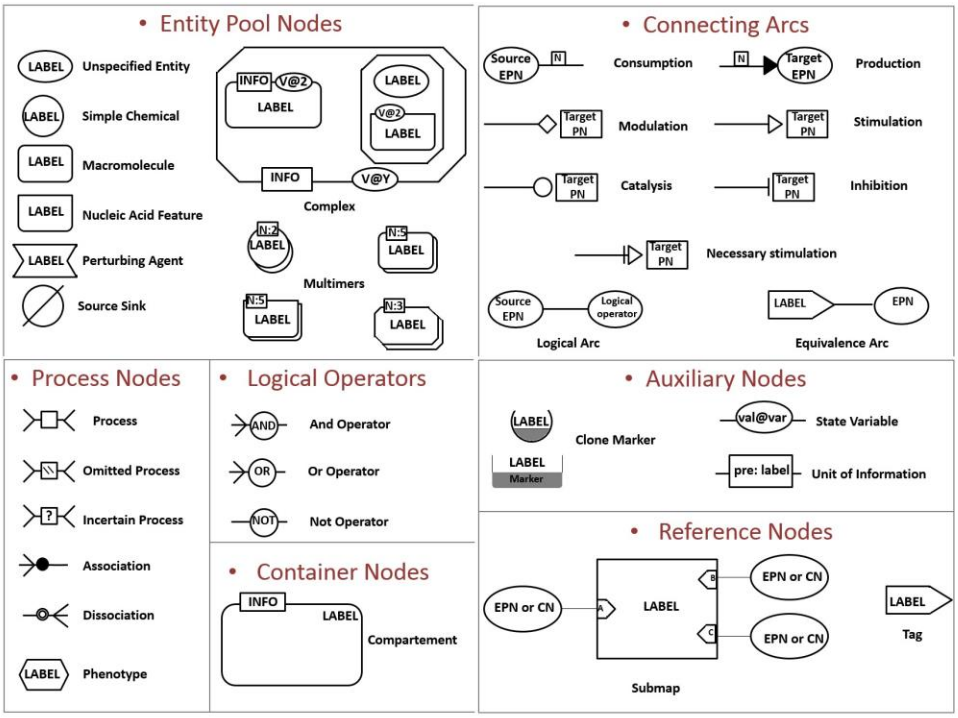


Figure. 2 SBGN process description diagram symbols inspired by [2] and [13]

- **Auxiliary Units** are glyphs used to decorate other glyphs in order to give additional information that may help the reader to well understand diagrams. They refer to *unit of information, state variable and clone marker*.
- **Container Nodes** referring to *compartment* which is a logical or physical structure that contains entity nodes.
- **Reference Nodes** handle links or relationships between elements of a map and sub-map. *Submap* is used to encapsulate processes including all types of nodes and arcs within a glyph. The sub-map hides its contents from the users, and displays only the input terminals linked to the entity node. A *tag* is also a reference node, it is a named handle, or reference, to another entity pool node or compartment. Tags are used to identify those elements in submaps.
- **Logical Operators** Combine one or several inputs into one output. Logical operators are *AND, OR* and *NOT*.

In order to define the meta-model corresponding to SBGN-PD, we have been inspired by the SBGN syntax proposed in [2].

Fig. 3, shows the Class Diagram representing the SBGN-PD meta-model containing all the SBGN-PD concepts shown in Fig. 2.

We define the process concept **“SBGNProcess”** composed of all glyphs of the process diagram **“SBGNGlyph”**, which is a generalization of two categories: arc **“SBGNArc”** and node **“SBGNNode”**. Each category is a generalization of several subcategories. “**SBGNNode**” contains **“Process”**, **“EntityPoolNode”**, **“Compartment”** and **“ReferenceNode”**.

Similarly, the **”SBGNArc”** is divided into **“SBGN_PN”** for arcs connecting processes to nodes, **“SBGN_NP”** for arcs connecting nodes to processes and **“FluxArc”**.

To enhance our SBGN-PD meta-model we add OCL (Object Constraint Language)[19] constraints.

-The first constraint is defined at the “**Association**” concept. An association is a biological process gathering one or more entity nodes, the resulting object is a **“Complex”**. This constraint is added to express that an **“Association”** process must have at least two input arcs and one output arc, this is defined in OCL as follows :

Context Association
invariant inv1: self.origin->size()>1;
invariant inv2: self.target->size()=1 and forAll(x | x.oclIsTypeOf(Complex));

-The second constraint is used to express that an “**Entity Node Pool**” can not have more than one “**Clone Marker**”. A clone marker is a part of the surface of an entity pool node that has been

Received: May 13, 2018

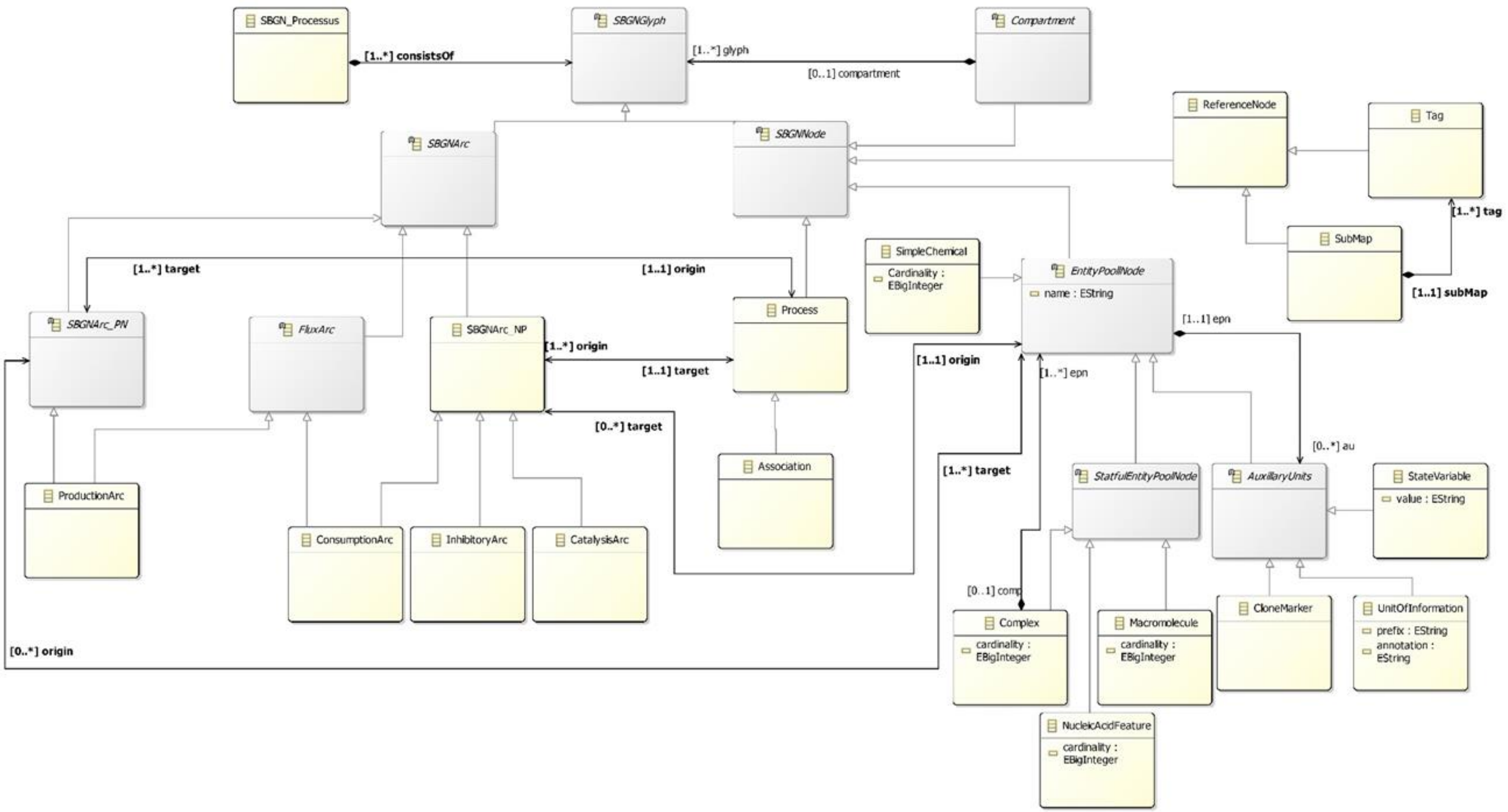


Figure. 3 Class diagram representing the SBGN-PD meta-model

modified graphically through the use of a different shade, texture, or colour. The constraint is as follows :

Context EntityPoolNode
invariant inv: self.au->select(
oclIsTypeOf(CloneMarker))->size()<=1;

-The third constraint makes it possible to express that an entity of “**Simple Chemical**” type - a chemical compound that is not formed by the covalent linking of pseudo-identical residues according to [20] - can have only one type of “**Auxiliary Unit**” which is the ”**Clone Marker**” :

Context SimpleChemical
invariant inv1: self.au->select(
oclIsTypeOf(UnitOfInformation))->isEmpty();
invariant inv2: self.au-> select(
oclIsTypeOf(StateVariable))->isEmpty();

2.3 HFPN: hybrid functional petri nets

Petri net is a formal specification language consisting of place, transition and tokens. A place can hold a number of tokens as its content, a transition has arcs coming from places and arcs going out from the transition to some places. A transition, through their connected arcs, defines a firing rule relative to the contents of the places where the arcs are attached.

HFPN [4] is a Petri net extension which includes in one formalism, HPN (Hybrid Petri Nets), HDN (Hybrid Dynamic Nets), and FPN (Functional Petri Nets).

According to [20], HPN represent an extension of ordinary Petri nets which contains two kinds of places and transitions : Discrete places (resp transitions) and continuous places (resp transitions). Discrete places and transitions correspond to places and transitions in ordinary Petri nets. A continuous place contains a quantity which is a non-negative real. A continuous transition fires continuously at a fixed rate, specified by a parameter assigned to the transition. Besides, HPN offer different types of arcs, namely, normal, test and inhibitor arcs.

The graphical notations of a discrete transition, a discrete place, a continuous transition, and a continuous place, together with the three types of arcs are shown in Fig. 4.

A weight *W* is a specific value and is assigned to each arc.

When a normal arc is attached to a discrete/continuous transition, *W* tokens are transferred through the normal arc.

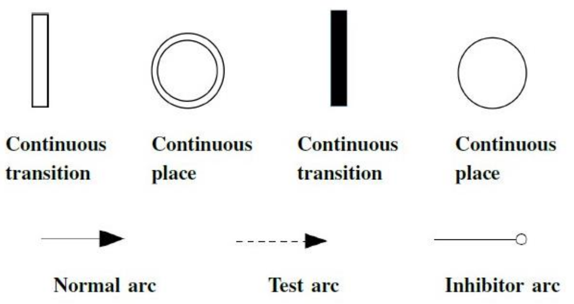


Figure. 4 Graphical notations of Hybrid Functional Petri Net elements inspired by [4, 20, 21]

An inhibitor arc allows firing the transition only when the contents of the source place is less than the weight *W*.

A test arc allows to not consume the contents of the source place when destination transition fires.

Hybrid dynamic net (HDN) [22], has a similar structure to the HPN since it uses the same kinds of places and transitions. They are different in terms of the firing rule of continuous transitions. In HDN, the speed of continuous transition can be given as a function of values in the places.

In fact, both of the features for the firing mechanisms of continuous transition of HPN and HDN, are actually required for modelling common biological reactions. For this reason, based on this motivation, HFPN including both features above, has been proposed.

In addition to HDN and HPN, HFPN has a third feature for arcs. A function of values of the places can be assigned to any arc. This feature has been introduced in FPN (Functional Petri nets) [23] in order to realise the calculation of dynamic biological catalytic process on Petri net based biological pathway modelling.

An HFPN [4], is then a triplet (*P*,*T*,*A*):

- A set of places $P = P_c \cup P_d$, P_c elements (resp. P_d) are called continuous (resp. discrete) places.
- A set of transitions $T = T_c \cup T_d$ such that $T \cap P = \{\}$, the T_c (resp. T_d) are called continuous (resp. discrete) transitions.
- A set of arcs $A \subset (T \times P) \cup (P \times T)$; Each arc is annotated with a label showing if this is :
  - a continuous input (output) arc, it connects a continuous place (transition) to a continuous transition (place).
  - a discrete input (output) arc, it connects a continuous or discrete) place (discrete transition) to a discrete transition (any place).

- a test (inhibitor) input arc, it connects a (continuous or discrete) place to a (continuous or discrete) transition.

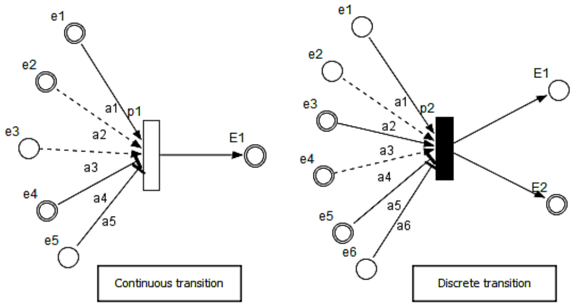


Figure. 5 Continuous (p1) and Discrete (p2) Transitions of hybrid functional Petri net inspired by [24].

Fig. 5, inspired by [24], shows HFPN different places (e_i) and arcs attached to the continuous (discrete) transition p1 (p2). It also shows different arcs headed from transitions p1 and p2. a_i is the weight of arc from place ei for i = 1; 2; 3; 4; 5.

2.4 HFPN meta-model

We define HFPN meta-model representing three general concepts of the language, i.e. ”Place”, ”Transition” and ”Arc”.

Fig. 6, shows the Class diagram representing the HFPN meta-model containing the three HFPN concepts previously presented as follows :

- Place is a generalization of two categories, continuous **“ContinuousPlace”** and discrete **“DiscretPlace”**.
- Transition is a generalization of two categories, continuous **“ContinuousTransition”** and discrete **“DiscretTransition”**.
- Arc is generalization of two arc categories :
  - **“ArcPT”** as arcs connecting places to transitions. They include **“TestArc”**, **“InhibitoryArc”** and **“NormalArc_PT”**.
  - **“ArcTP”** as arcs connecting transitions to places. They include **“NormalArc_TP”**.

3. SBGN2HFPN transformation

We are interested by a formal analysis of SBGN-PD models representing complex biological systems. This analysis takes place through transformation based on a mapping from SBGN-PD models to Petri nets, for which efficient formal tools exist.

HFPN has been successfully used for modelling and simulation of many biological processes [21, 33-35].

We use HFPN models as they are the most suited Petri net extension to model and study this kind of systems [25], and their capabilities for mathematical analysis.

The second large stage of our proposal consists then on defining the transformation rules allowing the mapping of an SBGN model to an HFPN model.

3.1 SBGN2HFPN transformation rules

We define ten rules allowing the transformation of any SBGN-PD model to an HFPN model, since those rules consider all the source and target meta-datas previously presented in Fig. 2.

Table 1 give the mapping rules summary of SBGN and HFPN elements.

- Rules R_1, R_2, R_3 and R_4 consist on transforming respectively four elements of SBGN-PD, namely, “Simple Chemical”, “Macromolecule”, “Complex” and “Nucleic Acid Feature” to an HFPN element which is a “Continuous Place”.
- Rules R_5 and R_6 consist on transforming two elements of SBGN-PD, namely, “Process” and “Association” to an HFPN element which is a “Continuous Transition”.
- The rule R_7 consists on transforming a “Consumption Arc” to a normal arc connecting a place to a transition and called “Normal Arc_TP”.
- The rule R_8 consists on transforming a “Production Arc” to a normal arc connecting a transition to a place and called “Normal Arc_TP”.
- The rule R_9 consist on transforming another SBGN-PD arc type called “Catalysis Arc” to a “Test Arc”.
- Finally, the last rule R_{10} consists on transforming an SBGN-PD “Inhibitor Arc” to an HFPN “Inhibitory Arc”.

The set of rules presented above contains the necessary and sufficient rules relative to the mapping (Cf. Table 1) of all the elements of the source model SBGN-PD (Cf. Fig. 2) to the target model HFPN (Cf. Fig. 4).

3.2 SBGN2HFPN transformation implementation

To complete the model-to-model transformation from SBGN-PD to HFPN we implement the transformation rules previously presented in subsection 3.1. This implementation is achieved by using ATLAS Transformation Language (ATL) [24, 26] (Fig. 7). ATL [27], according to [11] is a sophisticated Model to Model transformation language corresponding to MOF QVT [29]. It has been applied in several domains, many use cases are presented in [30].

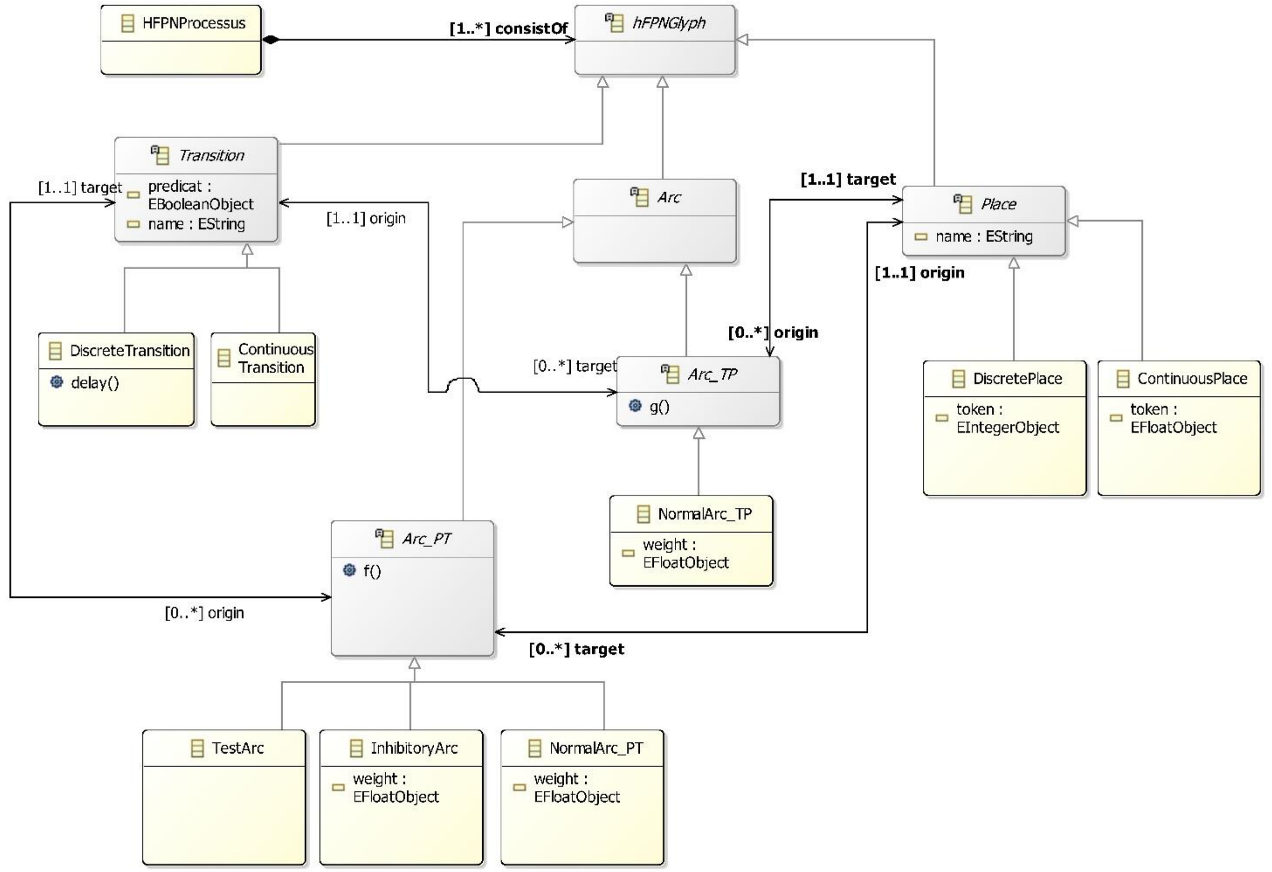
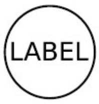
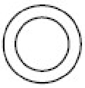
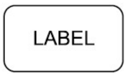
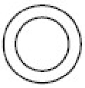
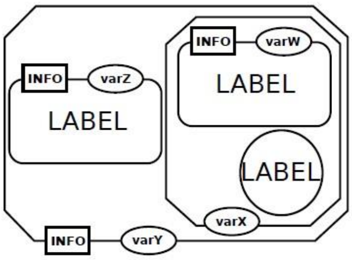
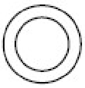
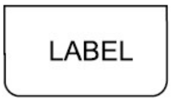
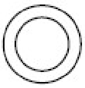
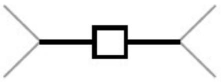
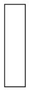
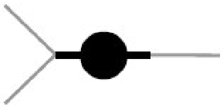
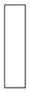
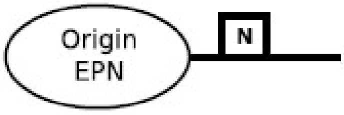
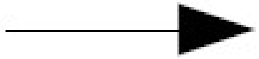
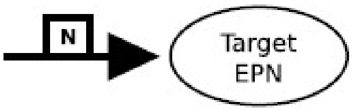
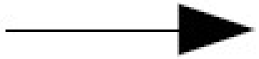
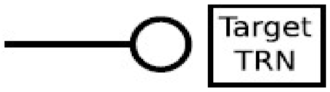
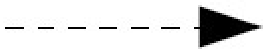
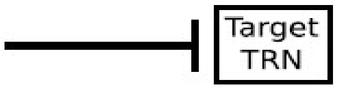
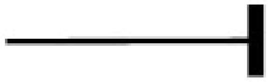


Figure. 6 Class diagram representing the HFPN meta-model

Table 1. SBGN-PD to HFPN Mapping

Rules	SBGN-PD		HFPN	
	Element	Glyph	Element	Glyph
R_1	SimpleChemical		ContinuousPlace	
R_2	Macromolecule		ContinuousPlace	
R_3	Complex		ContinuousPlace	
R_4	NucleicAcidFeature		ContinuousPlace	

R_5	Process		ContinuousTransition	
R_6	Association		ContinuousTransition	
R_7	ConsumptionArc		NormalArc_PT	
R_8	ProductionArc		NormalArc_TP	
R9	CatalysisArc		TestArc	
*R*10	InhibitorArc		InhibitoryArc	

-The **first rule** (R_1), converts a simple molecule **(SBGN-PD : Simple Chemical)** to a biochemical entity **(HFPN : Continuous Place)** as follows :

```
rule SimpleChemicalToContinuousPlace{
from Sc:SBGN!SimpleChemical
to cp:HFPN!ContinuousPlace(
name <- Sc.name,
origin <- Sc.origin,
target <- Sc.target)}
```

From this rule, we have assigned all properties of the source meta-data (name, origin, and target) to the target meta-data.

In the same way, we have defined three other transformation rules (R_2, R_3 and R_4) allowing respectively the conversion of **(SBGN-PD : Macromolecule, Complex and Nucleic Acid Feature)** to **(HFPN : Continuous Place)**.

-The **fifth rule** (R_5), converts **(SBGN-PD: Process)** to **(HFPN : Continuous Transition)** as follows :

```
rule ProcessToContinuousTransition{
from pc: SBGN!Process
to ct: HFPN!ContinuousTransition(
origin <- pc.origin,
target <- pc.target)
do {
thisModule.id <- thisModule.id + 1;
ct.name <-'T'+ thisModule.id;  }}
```

For this rule we have defined a section *do* which allows to define a different name to each transition created, by incrementing the global value *thisModule* every time the rule is executed.

In the same way, we have defined the transformation rule (R_6) which converts an **(SBGN-PD : Association)** to **(HFPN : Continuous Transition)**.

-The **seventh rule** (R_7), converts **(SBGN-PD: Consumption Arc)** to **(HFPN : Normal Arc from a place to a transition)** as follows :

```
rule ConsumptionArc2NoromalArc_PT{
from ca: SBGN!ConsumptionArc
to na_pt: HFPN!NormalArc_PT}
```

-The **eighth rule** (R_8), converts **(SBGN-PD: Production Arc)** to **(HFPN : Normal Arc from a transition to a place)** as follows :

```
rule ProductionArc2NormalArc_TP{
from pa: SBGN!ProductionArc
to na_tp: HFPN!NormalArc_TP }
```

-The **nineth rule** (R_9), converts **(SBGN-PD: Catalysis Arc)** to **(HFPN : Test Arc)** as follows :

```
rule CatalysisArc2TestArc{
from catA:SBGN!CatalysisArc
to ta:HFPN!TestArc}
```

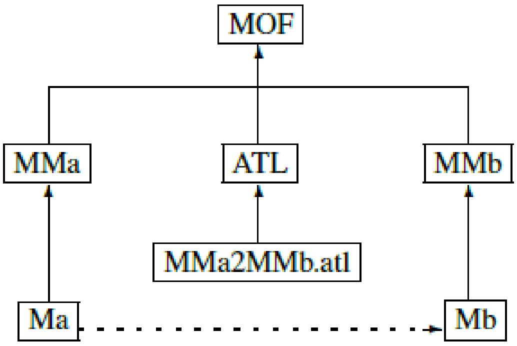


Figure. 7 The full model transformation process inspired from [24]. MMa: the source metamodel; MMb: the target metamodel; Ma: the source Model; Mb: the target model

-The **tenth rule** (R_{10}), converts **(SBGN-PD: Inhibitor Arc)** to **(HFPN : Inhibitory Arc)** as follows :

```
rule InhibitorArc2InhibitorArc{
from ia:SBGN!InhibitorArc
to ta:HFPN!InhibitorArc}
```

In the next section, we illustrate our proposal by applying the SBGN2HFPN transformation on a case of study (i.e. the Glycolysis Pathway).

In the order to validate the proposed transformation rules, we have conducted several tests. As an illustration, we consider in this paper, the Glycolysis Pathway as a case of study.

4. Case of study : glycolysis pathway

In this section, we follow the design steps mentioned previously in order to illustrate the SBGN2HFPN transformation on the Glycolysis Pathway. We are using the Glycolysis Pathway as an example, but our approach is general and can be applied to other metabolic pathways.

4.1 Glycolysis pathway : a brief presentation

Glycolysis, is the major pathway for glucose oxidation, it occurs in the cytosol of all cells. Glucose is converted to pyruvate in 10 steps by glycolysis.

Specific enzymes catalyze reactions through which glucose is converted to glucose 6P, fructose 6P, fructose 1.6 P and other metabolites until reaching pyruvate.

The pathway is modelled using SBGN-PD sub-language in [2] (Cf. Fig. 8), and is also modelled using HFPN in [21] (Cf. Fig. 11 (Right)).

The SBGN-PD model shown in Fig. 8 will play the role of source model of the proposed SBGN2HFPN transformation. The resulted HFPN model will be compared with the HFPN model shown in Fig. 11 (Right).

4.2 Illustration of the SBGN2HFPN on the glycolysis pathway

We create first meta-models corresponding to SBGN-PD and HFPN previously presented in Section 2, using EMF Eclipse Modelling Framework.

Then, we implement an Eclipse ATL file containing the transformation rules, allowing the mapping of SBGN-PD models to HFPN, mentioned in Section 3.

Note that the meta-models creation and the ATL rules implementation in Eclipse, have to be implemented only the first time of using the SBGN2HFPN transformation. Due to the manuscript limited Length, we will not show the corresponding catches.

Based on the SBGN-PD meta-model, we define the SBGN model in Fig. 9 corresponding to the SBGN model shown in Fig. 8.

After executing the ATL file, an XMI file containing elements of the HFPN model is generated by the SBGN2HFPN transformation as shown in Fig. 10.

Using then the HFPN concrete syntax, we interpret the HFPN.xmi file content generated by the SBGN2HFPN transformation.

Fig. 11 (Left) shows the HFPN model created based on the interpretation of the xmi file contents resulting from the SGBN2HFPN transformation.

We can observe that metabolites and enzymes represented by simple chemicals and macromolecules respectively in SBGN-PD model (Fig. 8) are represented by continuous places in the HFPN model. SBGN-PD processes are represented by continuous transitions. Catalysis arcs (i.e. arcs connecting macromolecules and processes) are represented by test arcs. Consumption and production arcs of the SBGN-PD model are represented by HFPN normal arcs. Indeed, all the SBGN-PD elements are converted to the corresponding HFPN elements, and are connected in a similar way.

Comparing HFPN models (Left) and (Right) in Fig. 11, we note that the resulting HFPN model from SBGN2HFPN transformation is almost similar to the existing HFPN model proposed in [21]. All elements and reactions occurring in the glycolysis pathway, are represented in both of models, in a similar way. ATP and ADP redundancy and metabolites degradation represent the whole

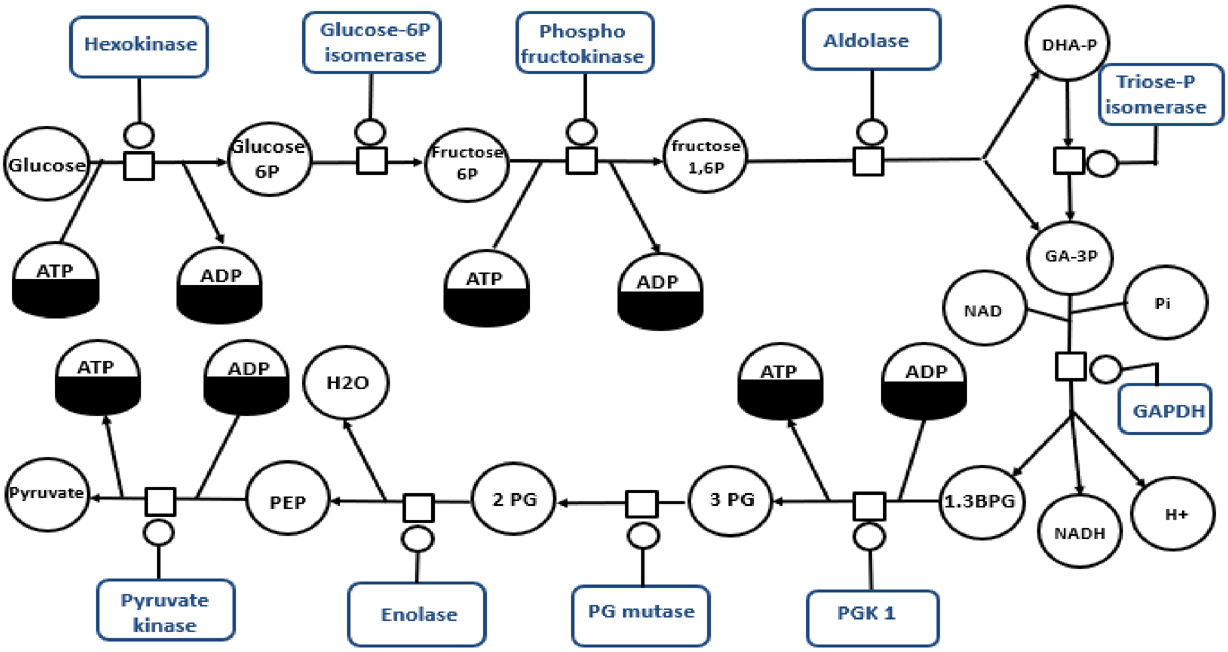
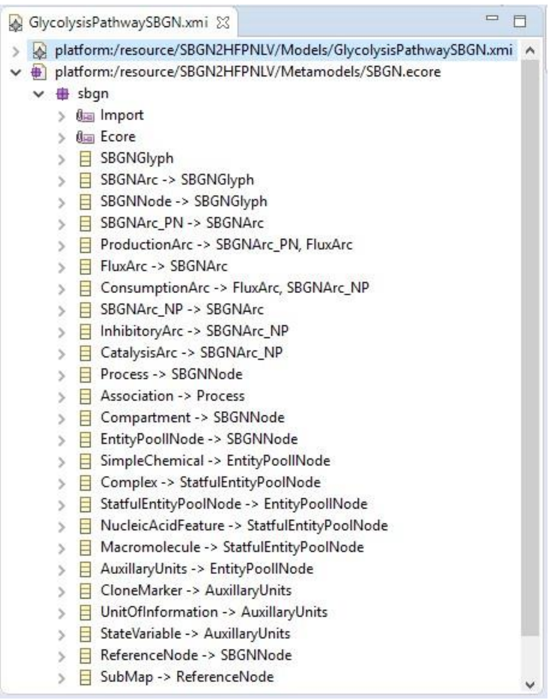
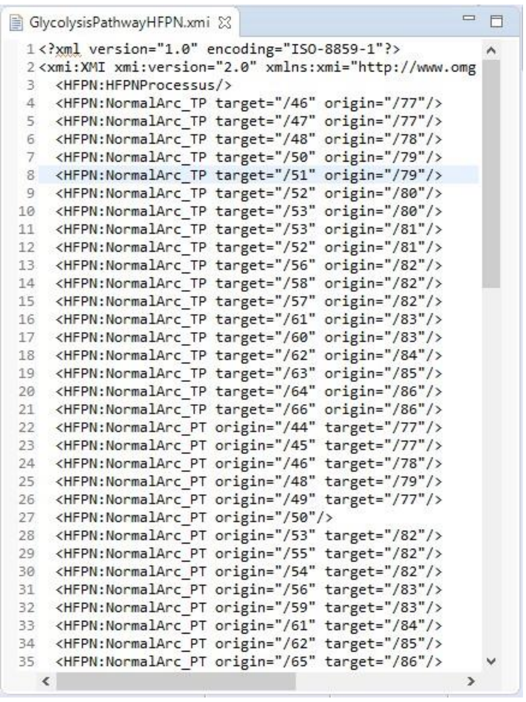


Figure. 8 SBGN-PD model representing the glycolysis pathway inspired by [2]



```
> platform:/resource/SBGN2HFPNLV/Models/GlycolysisPathwaySBGN.xmi
v platform:/resource/SBGN2HFPNLV/Metamodels/SBGN.ecore
  v sbgn
    > Import
    > Ecore
    > SBGNGlyph
    > SBGNArc -> SBGNGlyph
    > SBGNNode -> SBGNGlyph
    > SBGNArc_PN -> SBGNArc
    > ProductionArc -> SBGNArc_PN, FluxArc
    > FluxArc -> SBGNArc
    > ConsumptionArc -> FluxArc, SBGNArc_NP
    > SBGNArc_NP -> SBGNArc
    > InhibitoryArc -> SBGNArc_NP
    > CatalysisArc -> SBGNArc_NP
    > Process -> SBGNNode
    > Association -> Process
    > Compartment -> SBGNNode
    > EntityPoollNode -> SBGNNode
    > SimpleChemical -> EntityPoollNode
    > Complex -> StatfulEntityPoolNode
    > StatfulEntityPoolNode -> EntityPoollNode
    > NucleicAcidFeature -> StatfulEntityPoolNode
    > Macromolecule -> StatfulEntityPoolNode
    > AuxillaryUnits -> EntityPoollNode
    > CloneMarker -> AuxillaryUnits
    > UnitOfInformation -> AuxillaryUnits
    > StateVariable -> AuxillaryUnits
    > ReferenceNode -> SBGNNode
    > SubMap -> ReferenceNode
```

Figure. 9 SBGN-PD model representing the glycolysis pathway created in eclipse



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg
  <HFPN:HFPNProcessus/>
  <HFPN:NormalArc_TP target="/46" origin="/77"/>
  <HFPN:NormalArc_TP target="/47" origin="/77"/>
  <HFPN:NormalArc_TP target="/48" origin="/78"/>
  <HFPN:NormalArc_TP target="/50" origin="/79"/>
  <HFPN:NormalArc_TP target="/51" origin="/79"/>
  <HFPN:NormalArc_TP target="/52" origin="/80"/>
  <HFPN:NormalArc_TP target="/53" origin="/80"/>
  <HFPN:NormalArc_TP target="/53" origin="/81"/>
  <HFPN:NormalArc_TP target="/52" origin="/81"/>
  <HFPN:NormalArc_TP target="/56" origin="/82"/>
  <HFPN:NormalArc_TP target="/58" origin="/82"/>
  <HFPN:NormalArc_TP target="/57" origin="/82"/>
  <HFPN:NormalArc_TP target="/61" origin="/83"/>
  <HFPN:NormalArc_TP target="/60" origin="/83"/>
  <HFPN:NormalArc_TP target="/62" origin="/84"/>
  <HFPN:NormalArc_TP target="/63" origin="/85"/>
  <HFPN:NormalArc_TP target="/64" origin="/86"/>
  <HFPN:NormalArc_TP target="/66" origin="/86"/>
  <HFPN:NormalArc_PT origin="/44" target="/77"/>
  <HFPN:NormalArc_PT origin="/45" target="/77"/>
  <HFPN:NormalArc_PT origin="/46" target="/78"/>
  <HFPN:NormalArc_PT origin="/48" target="/79"/>
  <HFPN:NormalArc_PT origin="/49" target="/77"/>
  <HFPN:NormalArc_PT origin="/50"/>
  <HFPN:NormalArc_PT origin="/53" target="/82"/>
  <HFPN:NormalArc_PT origin="/55" target="/82"/>
  <HFPN:NormalArc_PT origin="/54" target="/82"/>
  <HFPN:NormalArc_PT origin="/56" target="/83"/>
  <HFPN:NormalArc_PT origin="/59" target="/83"/>
  <HFPN:NormalArc_PT origin="/61" target="/84"/>
  <HFPN:NormalArc_PT origin="/62" target="/85"/>
  <HFPN:NormalArc_PT origin="/65" target="/86"/>
```

Figure. 10 HFPN model content generated by SBGN2HFPN transformation in eclipse

differences between both models. This simple difference does not influence the representation of the modelled pathway either its behaviour. Indeed, in order to simulate the behaviour of HFPN models, we use the Cell Illustrator [31] tool.

This simulation requires parameters to be introduced, which represent among others, the entities' initial values and velocities of processes and reactions.

Figs**.** 12 and 13, show respectively the simulation results of the HFPN model resulting from SBGN2HFPN transformation and the existing HFPN model taken from [21].

These simulations allow to confirm the behavioural similarity of both HFPN **models (Cf.** Fig. 11) even that they have relatively some structural differences.

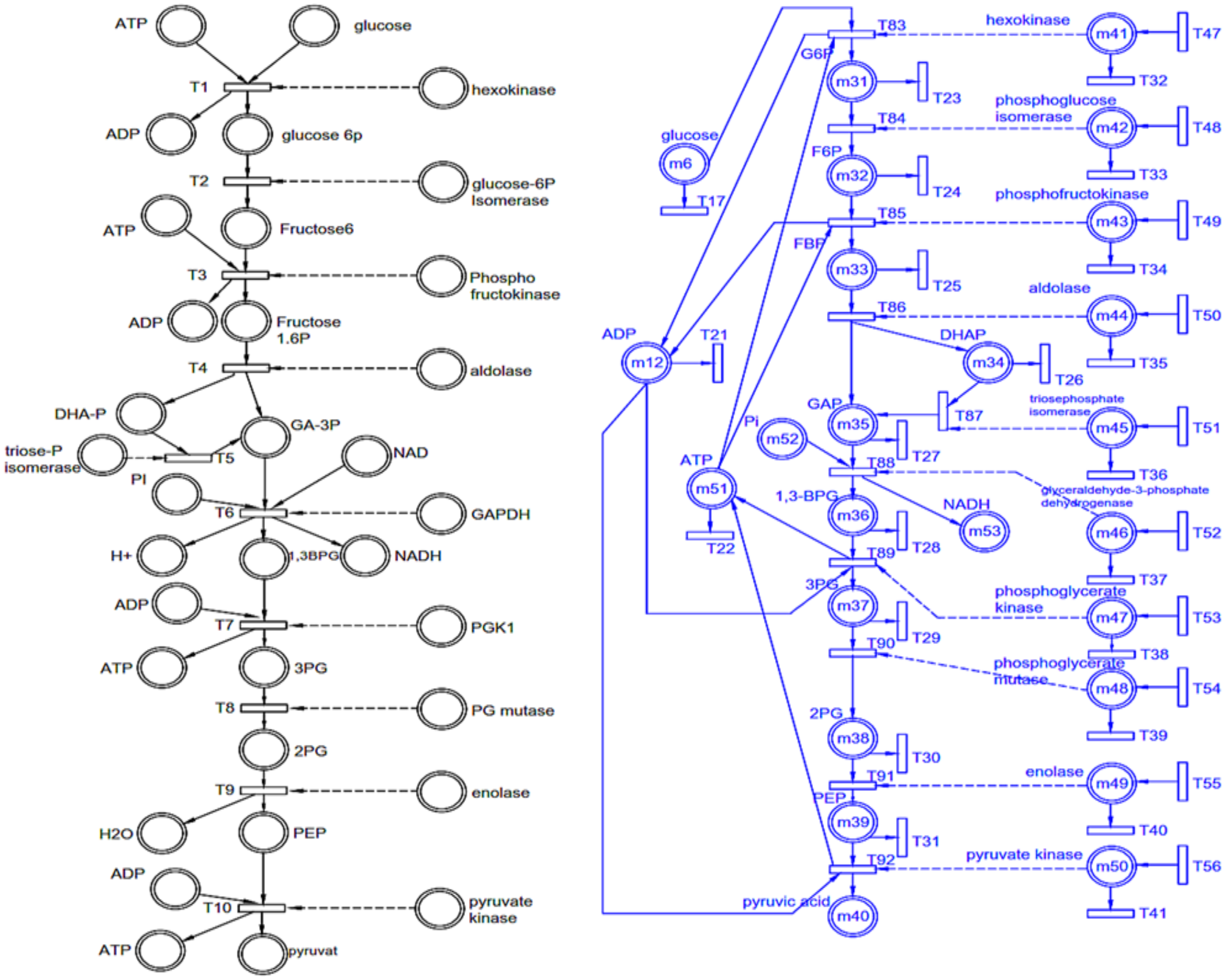


Figure. 11 (Left) The HFPN model created from the interpretation of the XMI file contents resulting from the SGBN2HFPN transformation. (Right) The HFPN model representing the Glycolysis Pathway taken from [21]

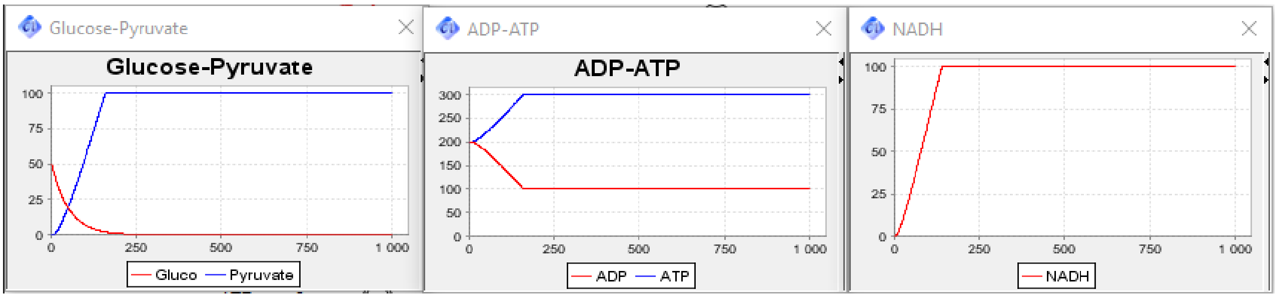


Figure. 12 Simulation result of the HFPN model resulting from SBGN2HFPN transformation

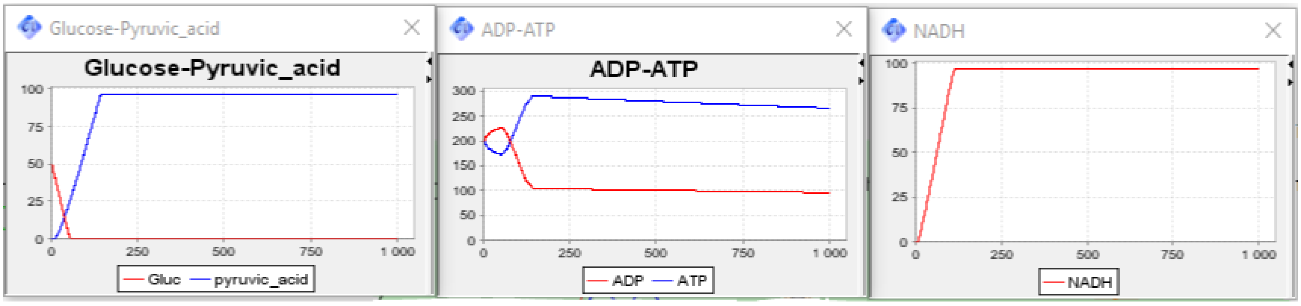


Figure. 13 Simulation result of the existing HFPN model proposed in [21]

We can therefore argue that the resulting model from the SBGN2HFPN transformation, models properly the studied system namely, the glycolysis pathway and confirm the effectiveness and validation of the proposed transformation.

5. Related works

There are not many contributions that can be directly related to our work however, there are some formalisations of SBGN' sub-languages in formalisms other than the Petri nets.

We present below some of those we have considered as relatively close to our proposal.

From an operating viewpoint of SBGN graphical notation, the proposal of SBGN-PD translation to SBGNtext is presented in [9], more detailed in [10].

SBGNtext is a textual representation of an abstraction of SBGN-PD maps called Process Flow Abstraction (PFA). Authors give as an example the creation of a Bio-PEPA model from a map represented in SBGNtext via SBGNtext2BioPEPA, their translation tool. They illustrate the process of this translation by applying a simple model of the MAPK signalling cascade [9]. They also illustrate their proposal on the metabolic pathway of the cholesterol synthesis in [8].

Regarding the transformation between SBGN-maps, a translation of SBGN-PD into SBGN-AF is proposed in [6]. Authors have proposed a semi-automatic translation of the SBGN-PD to SBGN-AF, implemented in SBGN-ED [32]. This last, is an SBGN maps editing software.

From a formal point of view, a transformation into mathematical logic has been proposed in [7]. Two first-order logic languages, called SBGNLog-PD and SBGNLog-AF, built from the SBGN-PD and SBGN-AF languages respectively are proposed. These languages make it possible to represent the reaction networks and the influence graphs in the form of sets of instantiated atoms, and to reason automatically on these networks.

Our proposal is standing in the analysis of biological systems using formal specification languages while keeping an intuitive graphical aspect of these systems. To do so, we use petri nets which combine perfectly both aspects.

6. Conclusion & future works

6.1 Conclusion

In this paper, we have proposed a model-to-model transformation from SBGN-PD to HFPN. It is based on Model Driven Engineering key principles, namely, meta-modelling and model transformation. The goal, is to facilitate the mapping between biopathways graphically modelled and formal models in the purpose of studying, understanding and analysing the modelled systems behaviour.

We have defined a meta-model for each of both languages, SBGN-PD and HFPN including their OCL well-completeness constraints, as well as transformation rules allowing the mapping of SBGN-PD to HFPN which are implemented using ATL language.

The SBGN2HFPN transformation was applied to many metabolic pathways, we presented in this paper the Glycolysis pathway as a case of study. Actually, our approach is general and can be applied to other metabolic.

A brief behavioural study in order to compare the HFPN model resulting from the SBGN2HFPN transformation and the existing model through simulation using Cell Illustrator was also presented.

These simulations have allowed to confirm the behavioural similarity of the both models for this case of study and generally allowed to confirm the effectiveness and validation of the proposed transformation.

6.2 Future works

We are currently intending to extend our transformation from SBGN to Petri nets. We are exploring the feasibility of transforming the other SBGN sub-languages to suitable formal languages for the purpose of studying and analysing the biosystems behaviour while combining the formal and graphical aspects.

References

[1] A. Finney, and M. Hucka, "Systems biology markup language: Level 2 and beyond", *Biochemical Society Transactions*, Vol. 31, No. 6, pp. 1472-1473, 2003.

[2] S. Moodie, N. Le Novere, E. Demir, H. Mi, and A. Villeger. "Systems biology graphical notation: process description language level 1 version 1.3", *Journal of Integrative Bioinformatics*, Vol.12, No. 2, pp. 213-280, 2015.

[3] H. Kitano. "Computational systems biology", *Nature*, Vol.420, No.6912, pp. 206–210, 2002.

[4] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano, "Biopathways representation and simulation on hybrid functional petri net", S*ilico Biology*, Vol.3, No.3, pp. 389–404, 2003.

[5] H. Matsuno, A. Doi, Y. Hirata, and S. Miyano, "Xml documentation of biopathways and their simulations in genomic object net", *Genome Informatics*, Vol.12, pp. 54–62, 2001.

[6] T. Vogt, T. Czauderna, and F. Schreiber, "Translation of sbgn maps: process description to activity flow", *BMC Systems Biology*, Vol.7, No.1, pp. 115-134, 2013.

[7] A. Rougny, "Méthodes qualitatives pour la construction et l'analyse des réseaux moléculaires SBGN", *PhD Dissertation*, Paris Saclay, France, 2016.

[8] L. Loewe, M. L. Guerriero, S. Watterson, S. Moodie, P. Ghazal, and J. Hillston, "Translation from the quantified implicit process flow abstraction in sbgn-pd diagrams to bio-pepa illustrated on the cholesterol pathway", *Transactions on Computational Systems Biology*, Vol. XIII, pp.13–38, 2011.

[9] L. Loewe, S. Moodie, and J. Hillston, "Quantifying the implicit process flow abstraction in sbgn-pd diagrams with bio-pepa", *arXiv preprint arXiv:0910.1410*, 2009.

[10] L. Loewe, S. Moodie, and, J. Hillston. "Technical report: Defining a textual representation for sbgn process diagrams and translating it to biopepa for quantitative analysis of the mapk signal transduction cascade", *Technical Report of the School of Informatics*, University of Edinburgh http://csbe. bio. ed. ac. uk/SBGNtext2BioPEPA/index. php, 2009.

[11] D. C. Schmidt, "Model-driven engineering", *Computer-IEEE Computer Society*, Vol.39, No.2, pp.25-34, 2006.

[12] H. Kitano, "Computational systems biology", *Nature*, Vol.420, No.6912, pp.206–210, 2002.

[13] A. Rougny, C. Froidevaux, L. Calzone, and L. Paulevé, "Qualitative dynamics semantics for sbgn process description*", BMC Systems Biology*, Vol.10, No.1, pp.42-66, 2016.

[14] M. P. Van Iersel, A. C. Villéger, T. Czauderna, S. E. Boyd, F. T. Bergmann, A. Luna, E. Demir, A. Sorokin, U. Dogrusoz, Y. Matsuoka, A. Funahashi, M.I. Aladjem, H.Mi, S. Moodie, H. Kitano, N. Le Novere, and F. Schreiber, "Software support for sbgn maps: Sbgn-ml and libsbgn", *Bioinformatics*, Vol.28, No.15, pp.2016–2021, 2012.

[15] "COPASI: Biochemical System Simulator", *http: //copasi.org/,* 2017. [Online accessed 30 March-2018].

[16] "The Biochemical Abstract Machine BIOCHAM3", *http://contraintes.inria.fr/biocham/,* 2017. [Online accessed 29-March-2018].

[17] "CellDesigner: A modeling tool of biochemical networks", *http://www.celldesigner.org/,* 2017. [Online accessed 28-March-2018].

[18] "Systems Biology Graphical Notation: Software Support", *http://sbgn.github.io/sbgn/Software_support/,* 2016. [Online accessed 1-April-2018].

[19] "Documents Associated With Object Constraint Language (OCL), Version 2.4", *http://www. omg.org/spec/OCL/,* 2014. [Online accessed 26-March-2017].

[20] H. Alla and R. David, "Continuous and hybrid petri nets", *Journal of Circuits, Systems, and Computers*, Vol.8, No.01, pp.159–188, 1998.

[21] S. Miyano and H. Matsuno, "How to model and simulate biological pathways with petri nets-a new challenge for systems biology", In: *Proc. of the 25th International Conference on Application and Theory of Petri nets*, 2004.

[22] R. Drath, "Hybrid object nets: An object oriented concept for modeling complex hybrid systems", In: *Proc. of Hybrid Dynamical Systems, 3rd International Conference on Automation of Mixed Processes ADPM'98*, pp.436–442, 1998.

[23] R. Hofestadt and S. Thelen, "Quantitative modeling of biochemical networks", *Silico Biology*, Vol.1, No.1, p39–53, 1998.

[24] F. Jouault and I. Kurtev, "Transforming models with atl*",* In: *Proc. of International Conference on Model Driven Engineering Languages and Systems, MoDELS Conference*, pp.128-138, 2005.

[25] A. Doi, S. Fujita, H. Matsuno, M. Nagasaki, and S. Miyano, "Constructing biological pathway models with hybrid functional petri nets", *Silico Biology*, Vol.4, No.3, pp.271–291, 2004.

[26] "ATL/Concepts", *https://wiki.eclipse.org/ ATL/Concepts,* 2012. [Online accessed 26-December-2017].

[27] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev, "Atl: A model transformation tool", *Science of Computer Programming*, Vol.72, No.1-2, pp.31– 39, 2008.

[28] F. Buttner, J. Cabot, and M. Gogolla, "On validation of atl transformation rules by transformation models", In: *Proc. of the 8th International Workshop on Model-Driven Engineering, Verification and Validation*, Wellington, New Zealand, pp.9-17, 2011.

[29] "Documents Associated With Meta Object Facility (MOF) 2.0

Query/View/Transformation (QVT), V1.2." *http://www.omg.org/spec/QVT/ 1.2/*, 2015. [Online accessed 26-December-2017].

[30] "Atl use cases" *http://www.eclipse.org/atl/.* [Online accessed 20-April-2018].

[31] "Cell illustrator 5.0", *http://www. cellillustrator.com/,* 2009. [Online accessed 25-April-2017].

[32] T. Czauderna, C. Klukas, and F. Schreiber, "Editing, validating and translating of sbgn maps", *Bioinformatics,* Vol.26, No.18, pp.2340–2341, 2010.

[33] S. Cherdal and S. Mouline, "A petri net model for a composed biological system", In: *Proc. of the 5th International Conference on Information & Communication Technology and Accessibility*, pp. 1—6, 2016.

[34] S. Cherdal and S. Mouline, "Petri nets for modelling and analysing a complex system related to alzheimer's disease", In: *Proc. of the 31st Annual ACM Symposium on Applied Computing,* pp. 309—312, 2016.

[35] S Cherdal and S. Mouline, "A Petri Net model to simulate and analyse cerebral folate deficiency and hyperhomocysteinemia effects in Autism Spectrum Disorder", In: *Proc. of the 7th international conference on Sciences of Electronics, Technologies of Information and Telecommunications,* pp. 279—284, 2016.