National Research University Higher School of Economics

Perm, Russian Federation

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

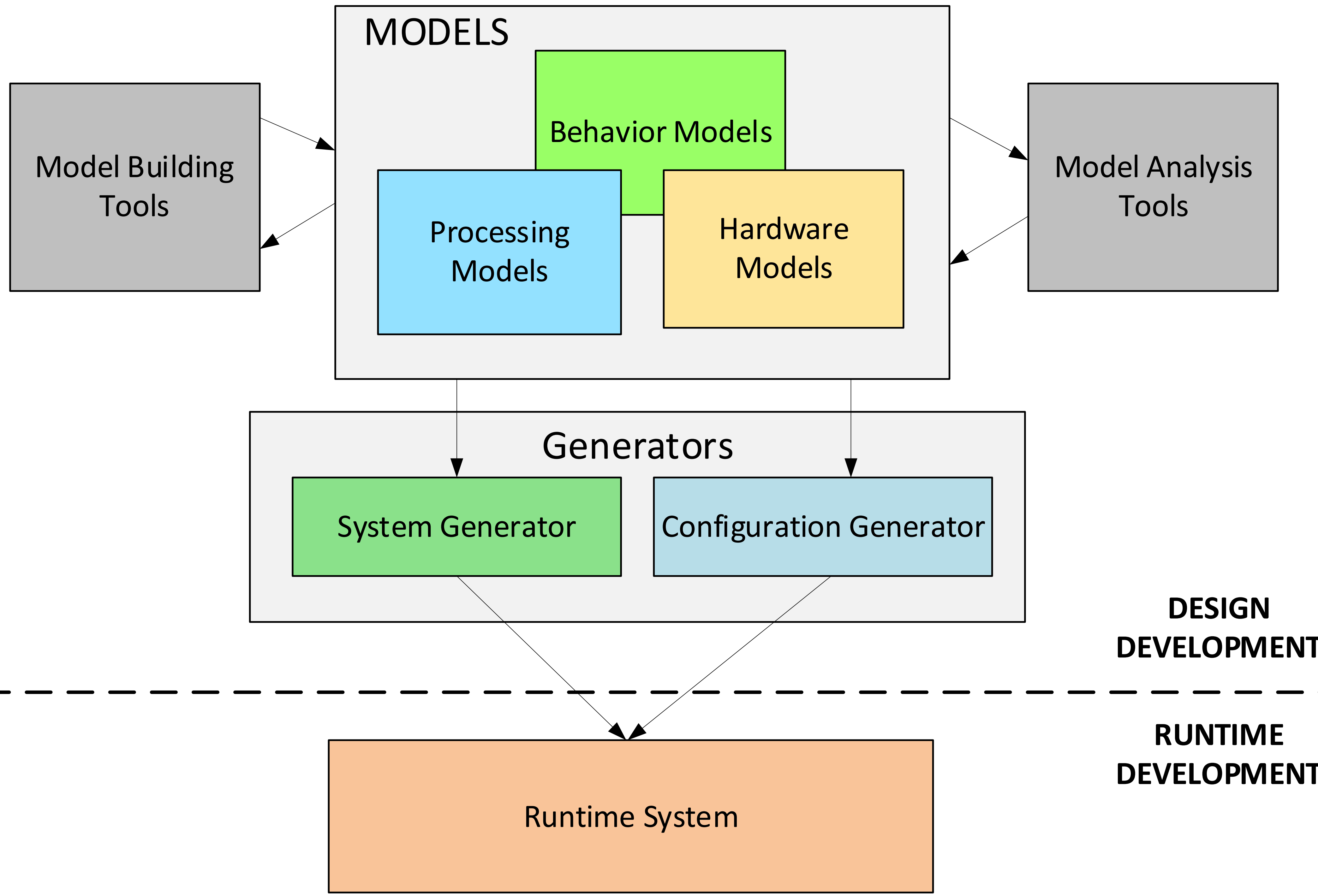
HP-GRAPH: DEFINITION AND APPROACHES TO OPTIMIZING ALGORITHMS FOR GRAPHS

Nikolai M. Suvorov

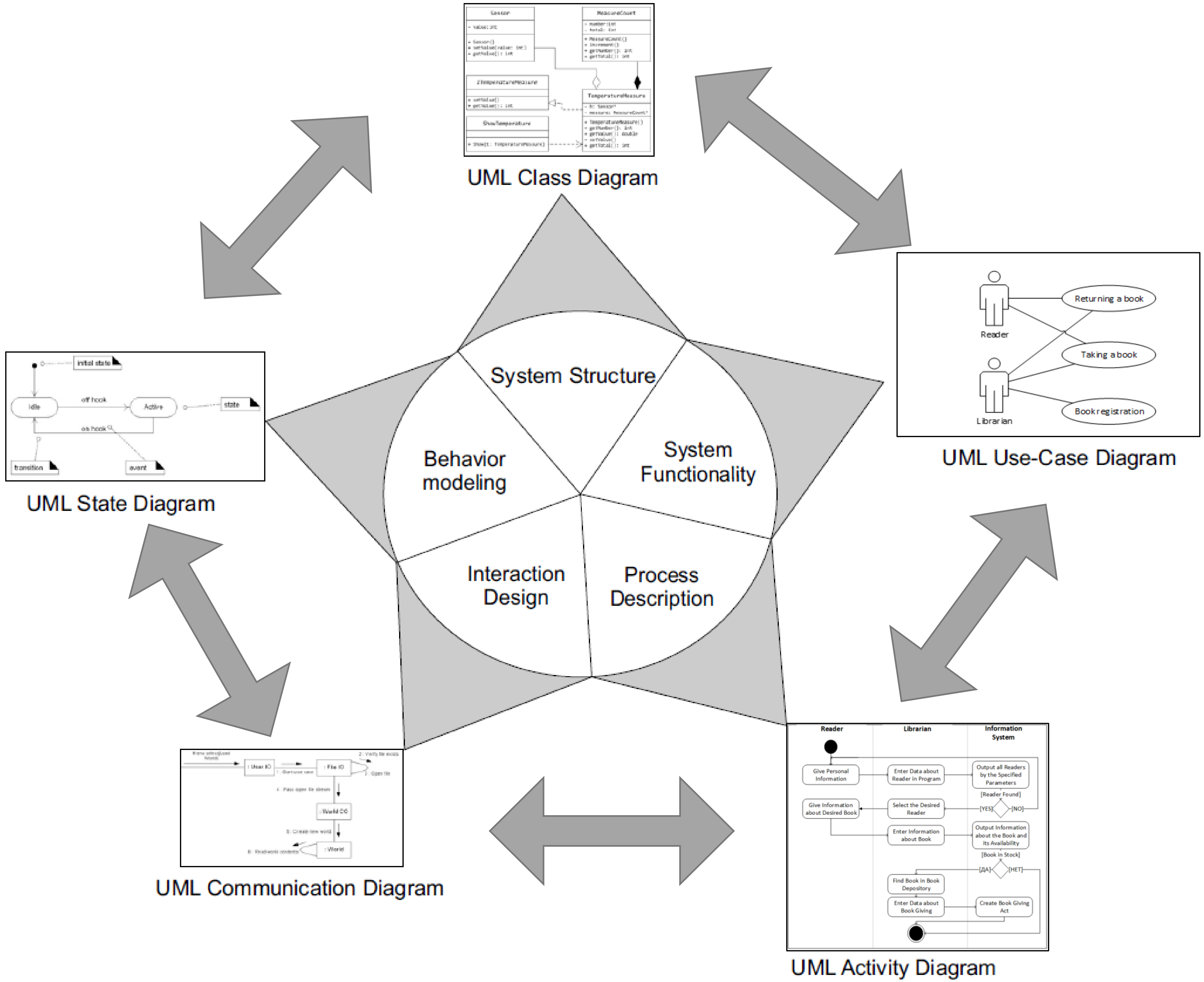
Lyudmila N. Lyadova

ITA, 2020

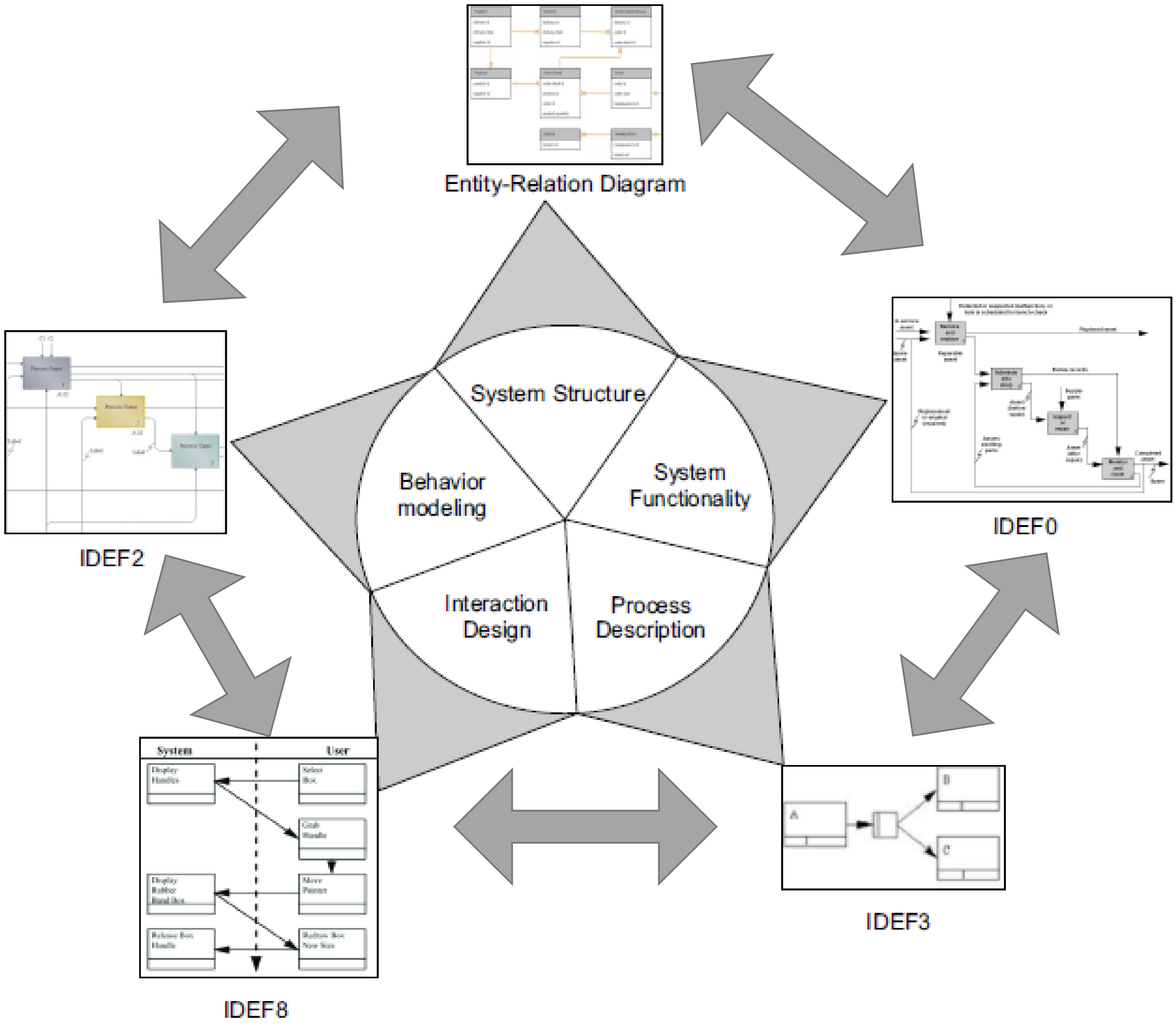
RESEARCH RELEVANCE:
MULTI-MODELING APPROACH



RESEARCH RELEVANCE:
RESEARCH AREA PROBLEM

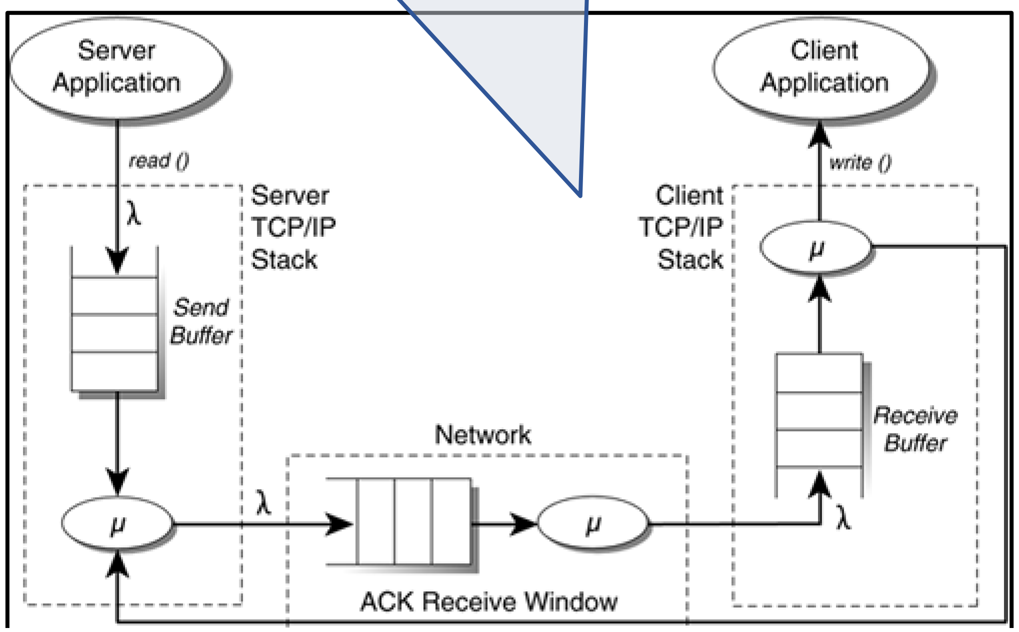


RESEARCH RELEVANCE:
RESEARCH AREA PROBLEM

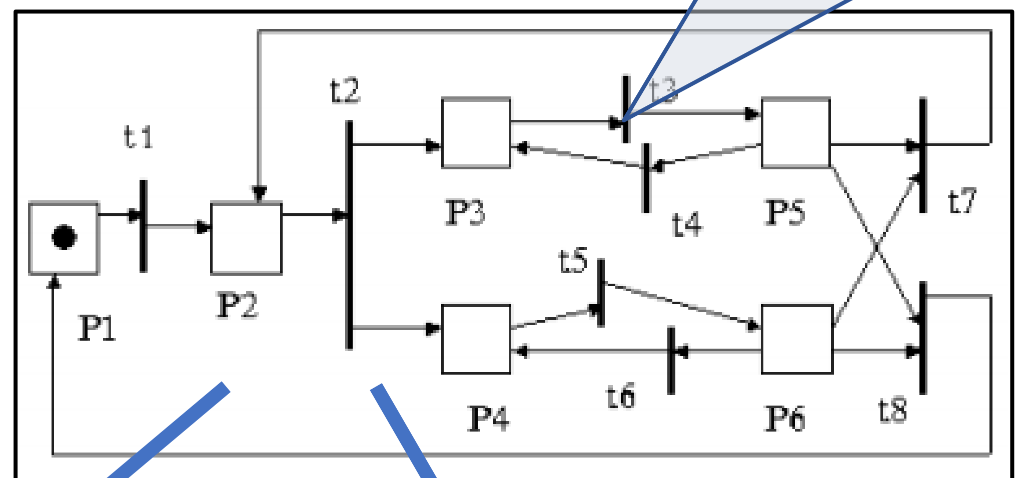


RESEARCH RELEVANCE:
RESEARCH AREA PROBLEM

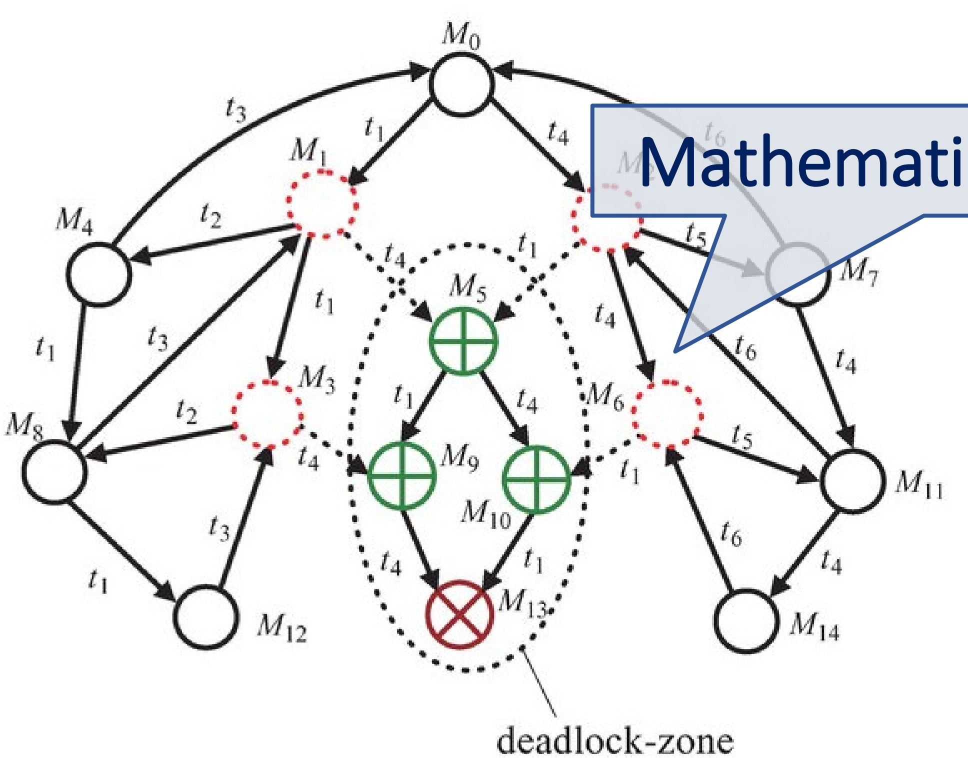
Queuing Systems Models



Petri Nets



Mathematical Modeling



good marking · dangerous marking · bad marking · dead marking

$M_0=3p_1+3p_6+2p_7+2p_8$

$M_1=2p_1+p_2+3p_6+p_7+2p_8$

$M_2=3p_1+p_4+2p_6+2p_7+p_8$

$M_3=p_1+2p_2+3p_6+2p_8$

$M_4=2p_1+p_3+3p_6+2p_7$

$M_5=2p_1+p_2+p_4+2p_6+p_7+p_8$

$M_6=3p_1+2p_4+p_6+2p_7$

$M_7=3p_1+p_5+2p_6+2p_8$

$M_8=p_1+p_2+p_3+3p_6+p_7$

$M_9=p_1+2p_2+p_4+2p_6+p_8$

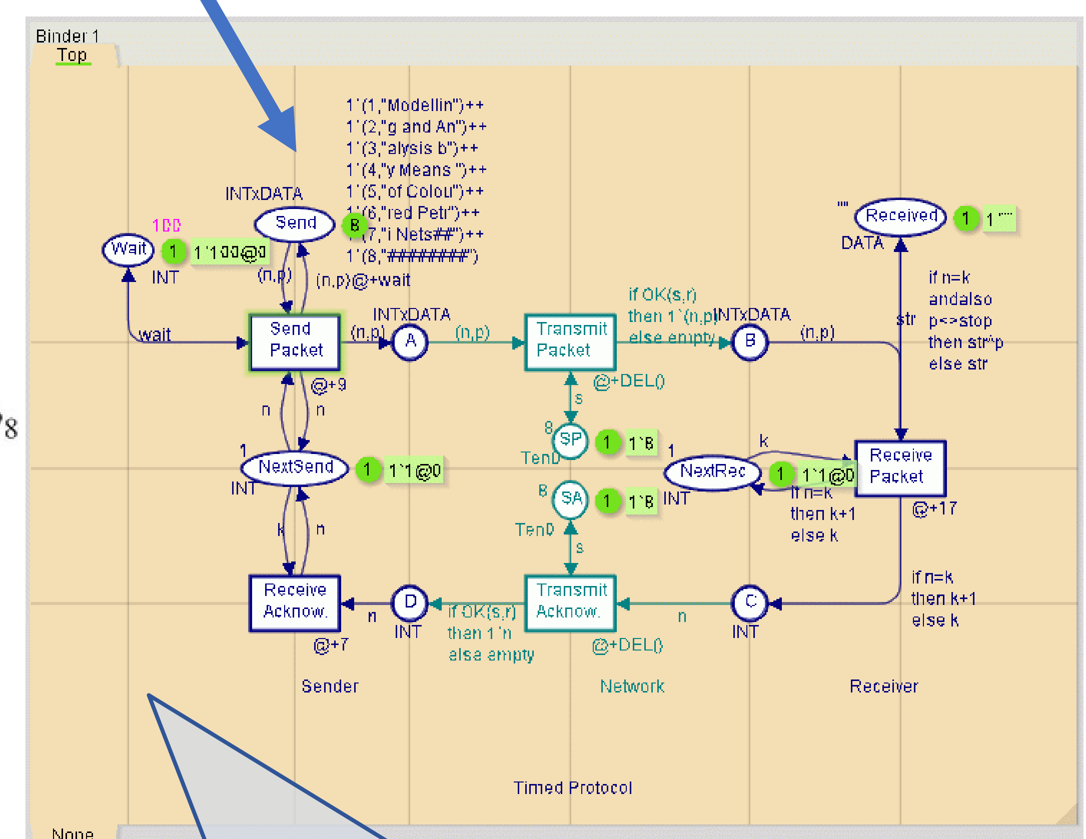
$M_{10}=2p_1+p_2+2p_4+p_6+p_7$

$M_{11}=3p_1+p_4+p_5+p_6+p_8$

$M_{12}=2p_2+p_3+3p_6$

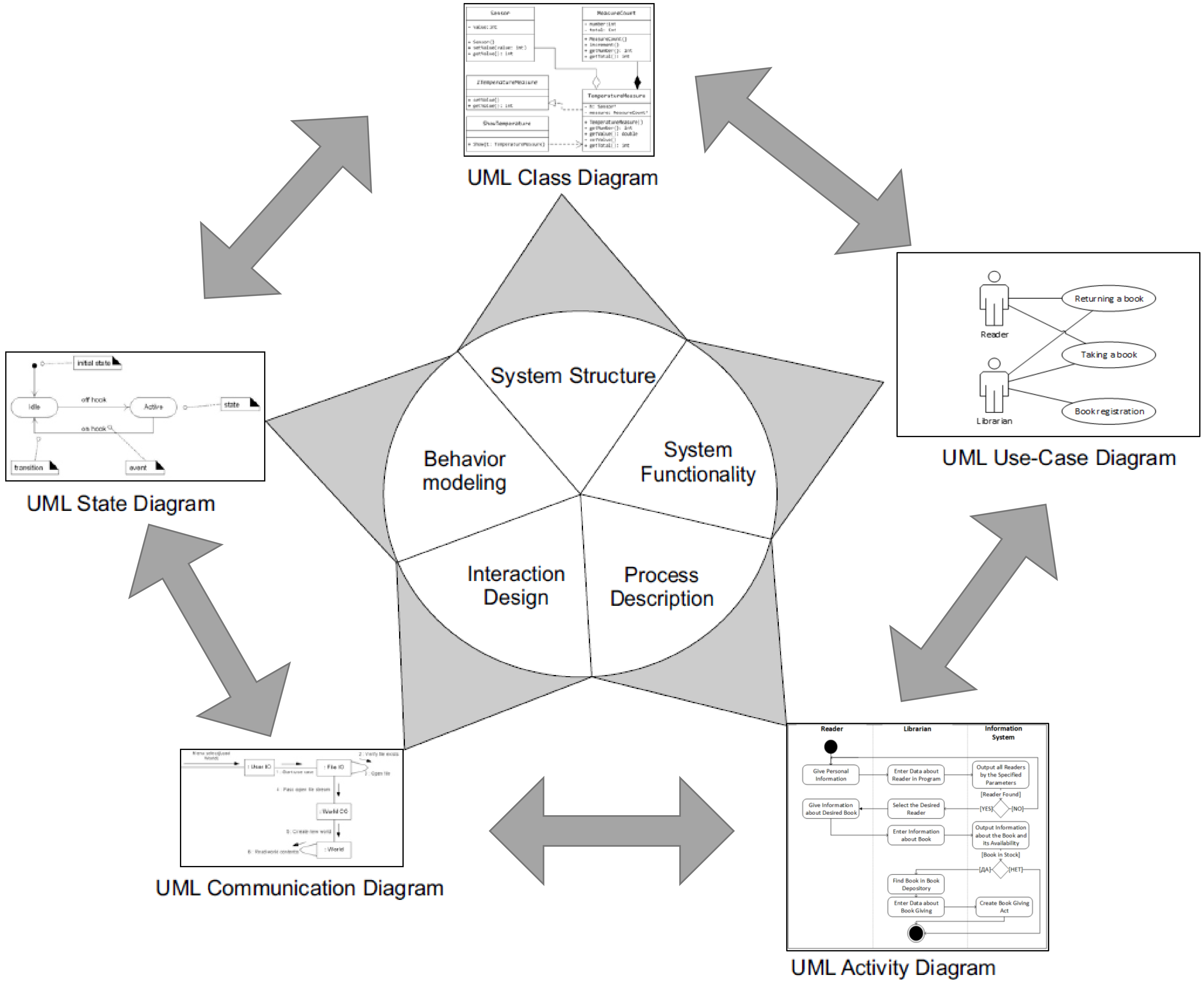
$M_{13}=p_1+2p_2+2p_4+p_6$

$M_{14}=3p_1+2p_4+p_5$

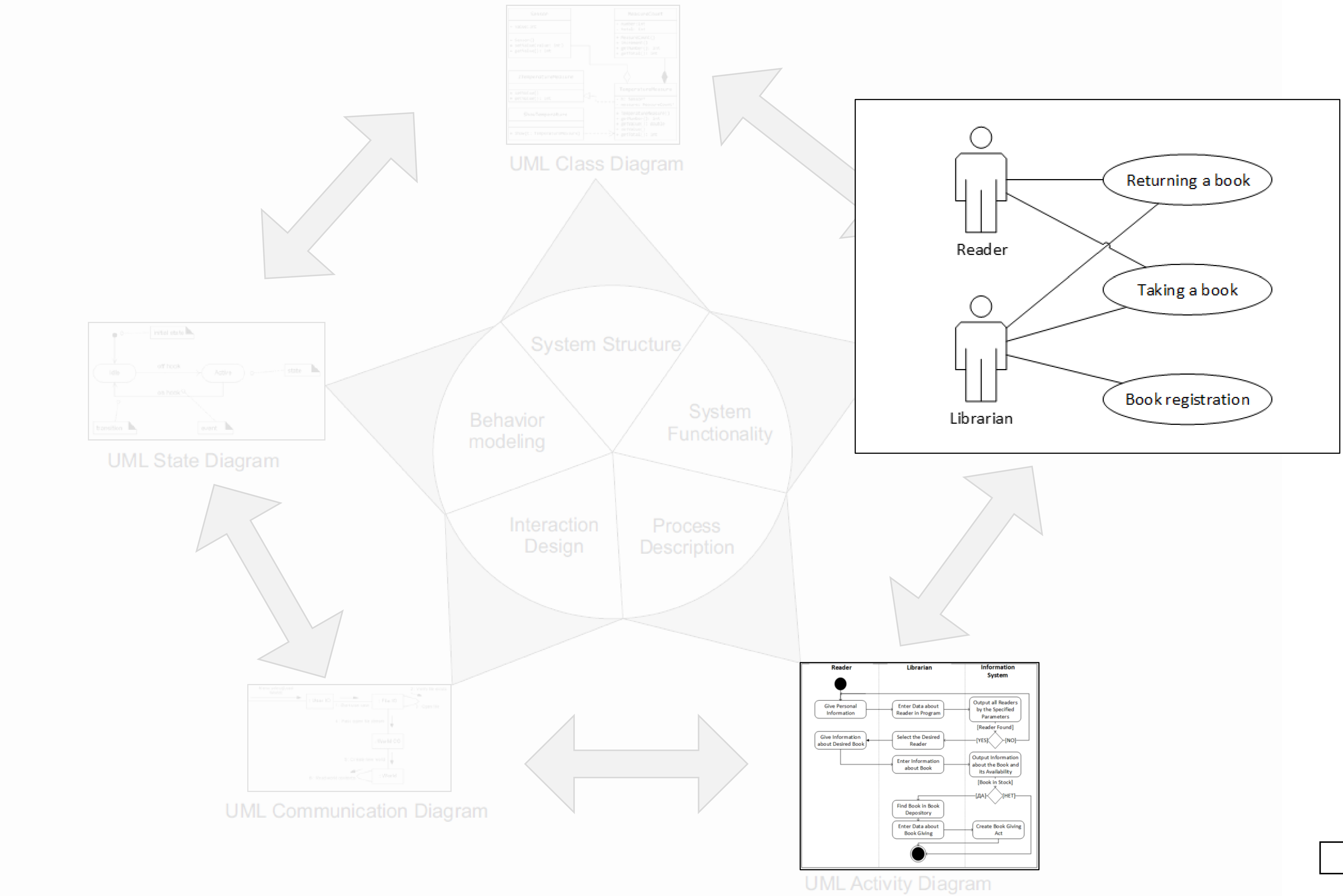


Simulation Models

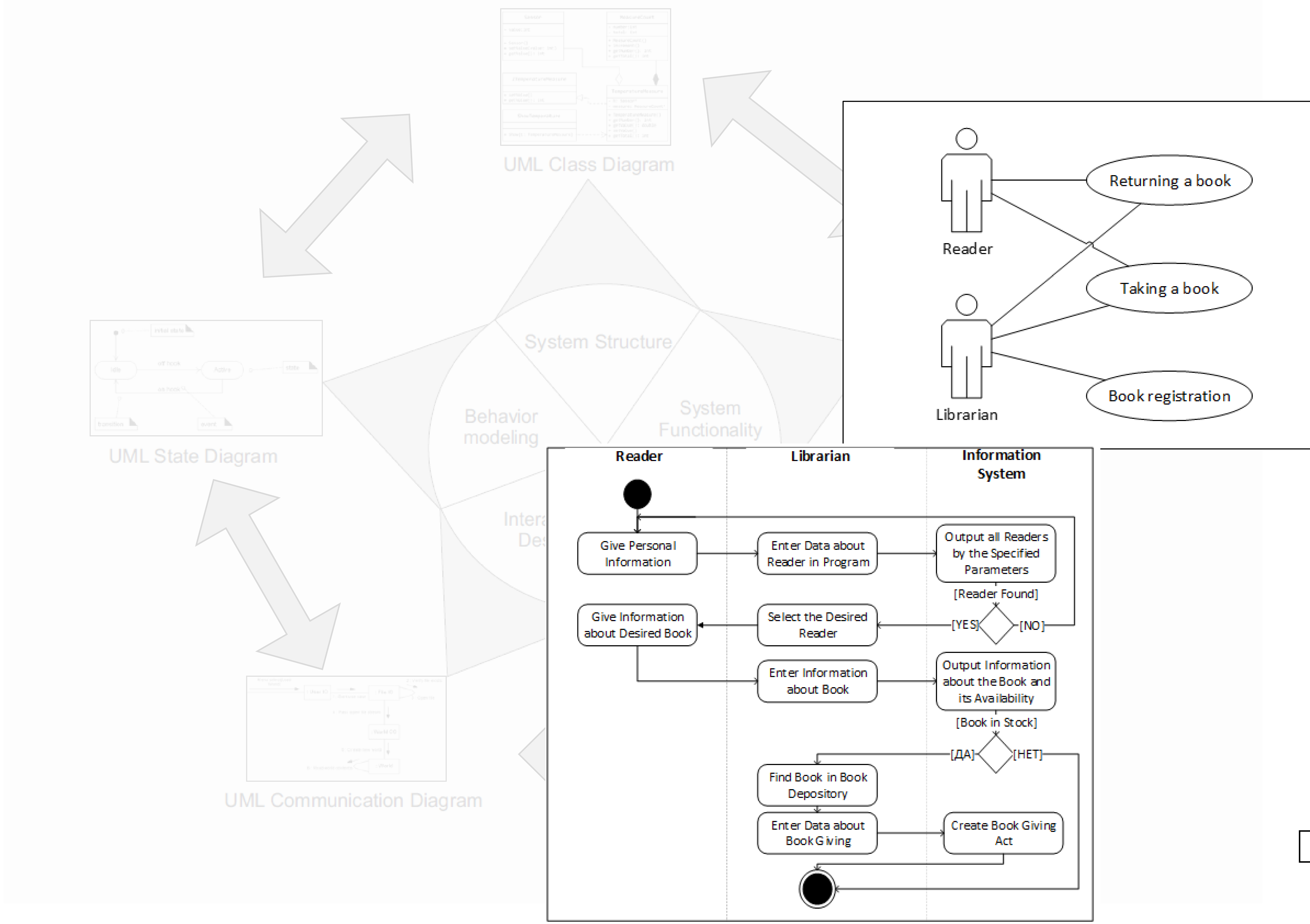
RESEARCH RELEVANCE:
RESEARCH AREA PROBLEM



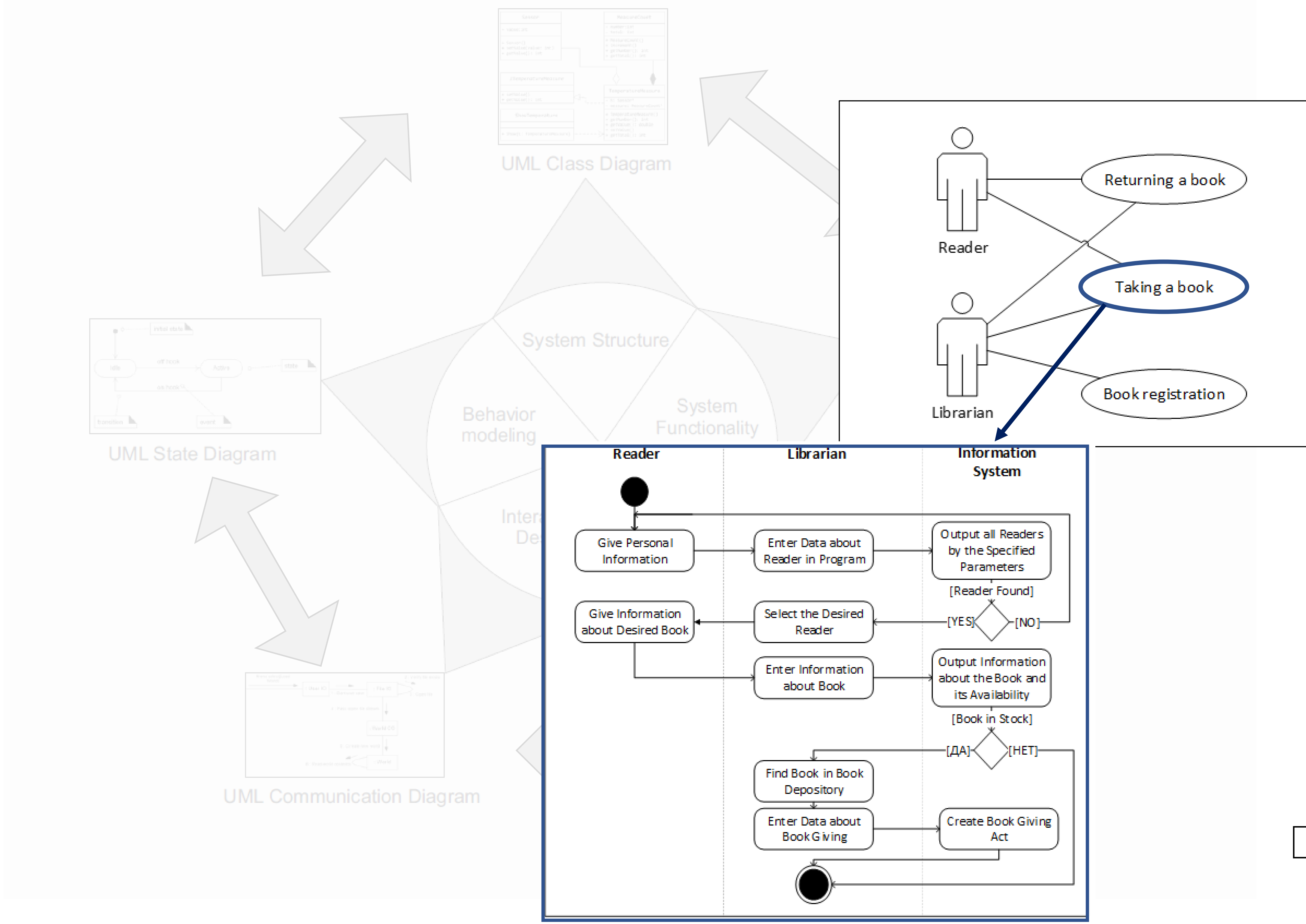
RESEARCH RELEVANCE: TRANSFORMATIONS



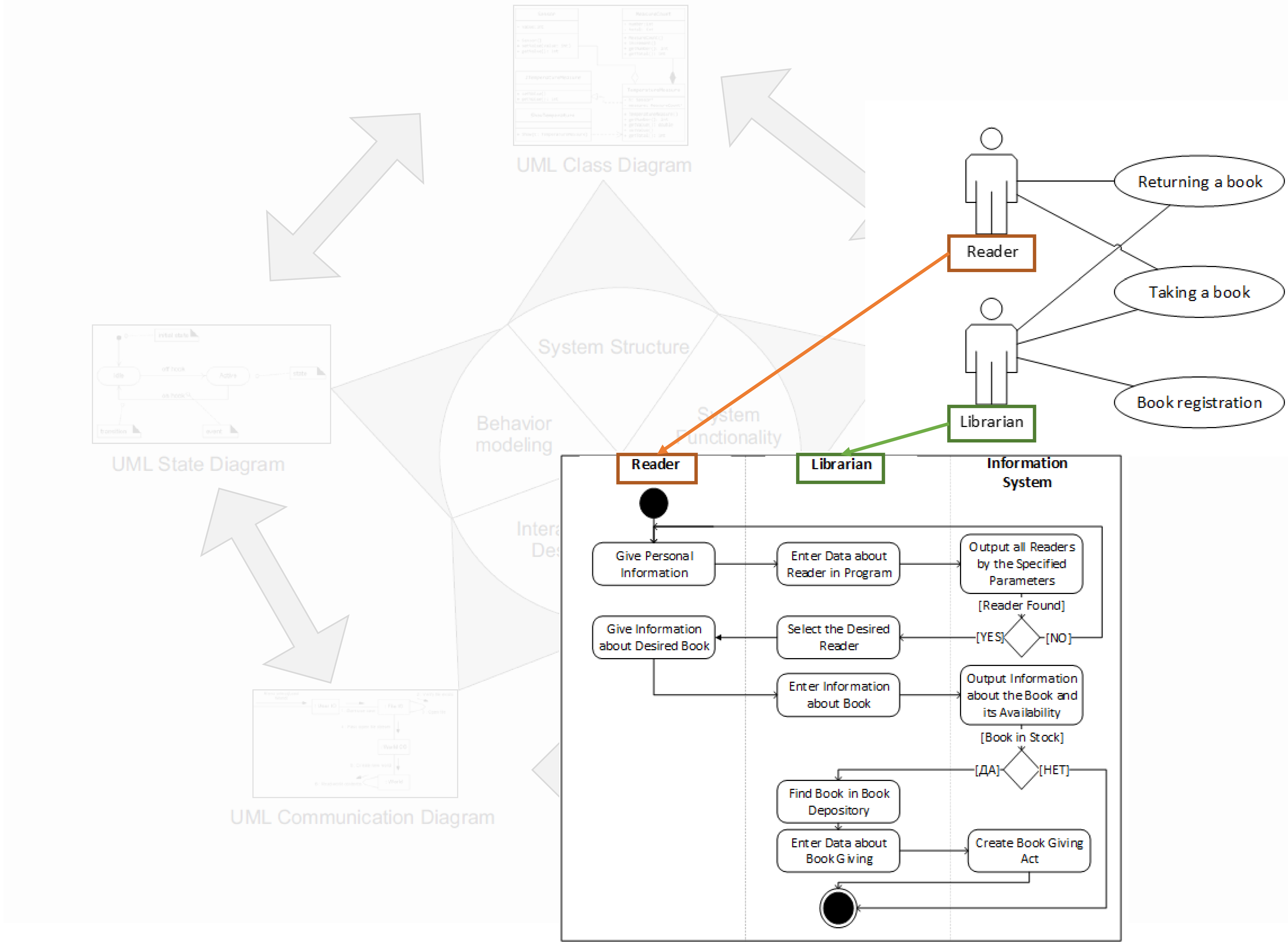
RESEARCH RELEVANCE:
TRANSFORMATIONS



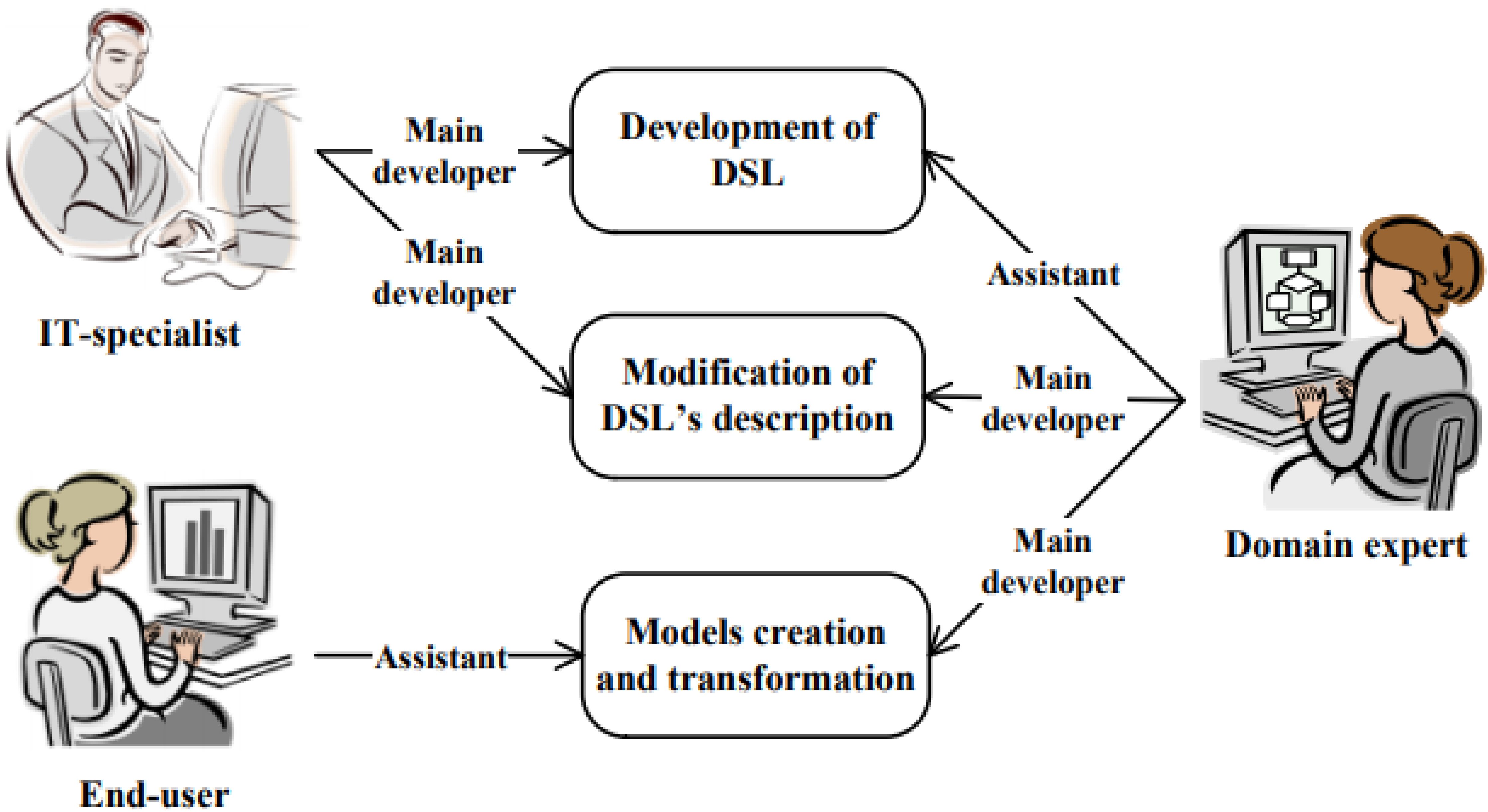
RESEARCH RELEVANCE: TRANSFORMATIONS



RESEARCH RELEVANCE: TRANSFORMATIONS



RESEARCH RELEVANCE:
LANGUAGE FOCUSED APPROACH



RESEARCH RELEVANCE:
RELATED WORKS

Graph model should:

Allow us to ***formalize the description of all elements***, included in visual languages	Allow us to ***implement general modeling principles***
Provide a possibility to ***implement transformations*** of various types of models	***Optimize the algorithms*** that should be implemented for modeling and solving tasks with models

RESEARCH PURPOSE AND TASKS

Purpose:

Defining a *new graph formalism* that can be used as a basis for a DSM platform development, providing a possibility to perform multi-level and multi-aspect modeling.

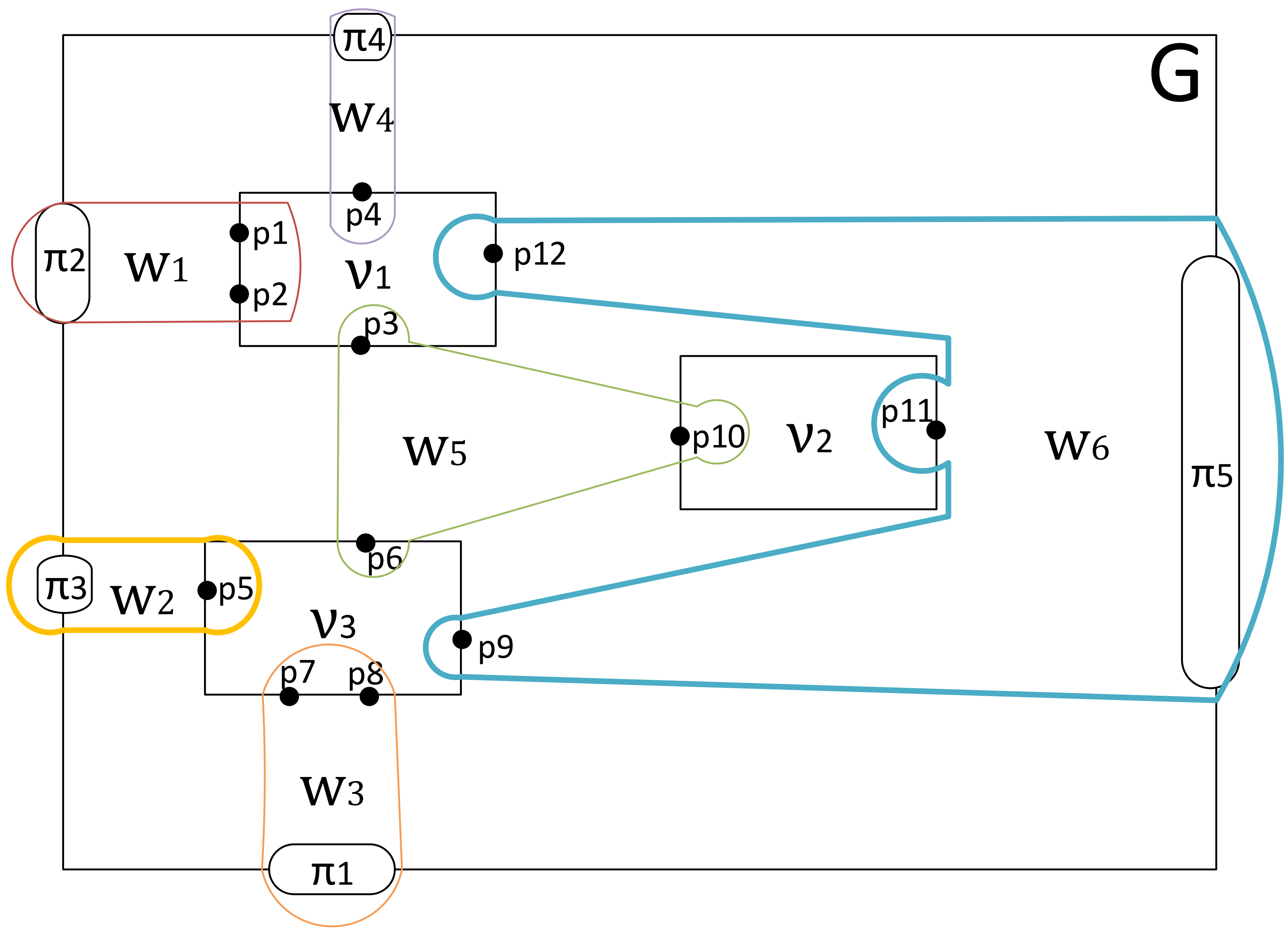
Tasks:

- Analysis of different types of graphs to determine how well they meet the mentioned requirements.
- Development of a new graph formalism and evaluating and comparing it with existing ones.
- Development of algorithms for the new graph formalism.
- Development of a visual editor object model.
- Development of a program, demonstrating the practical significance of the selected graph formalism.

GRAPH MODEL DEFINITION:
SETS OF HP-GRAPH

HP-graph is an ordered triple *G* = (*P*, *V*, *W*):

- $P = \{\pi_1,\ldots,\pi_n\}$ is a set of external poles of the graph.
- $V = \{v_1,\ldots,v_m\}$ is a non-empty set of mutually disjoint vertices, consisting of internal poles.
- $W = \{w_1,\ldots,w_l\}$ is a set of hyperedges, consisting of poles.
- *Pol* is a set of **all** poles of the graph

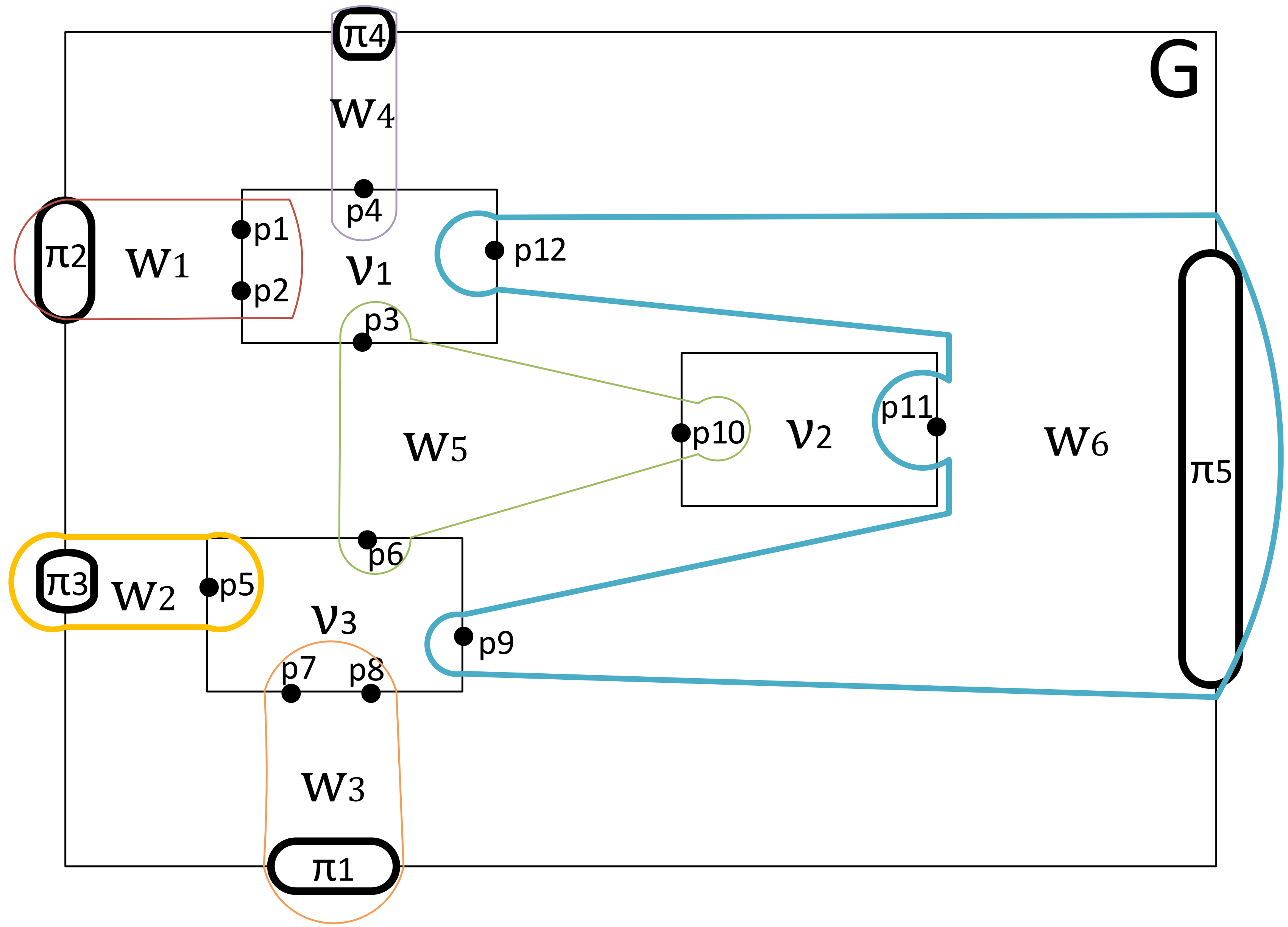


GRAPH MODEL DEFINITION:

SETS OF HP-GRAPH

HP-graph is an ordered triple $G = (P, V, W)$:

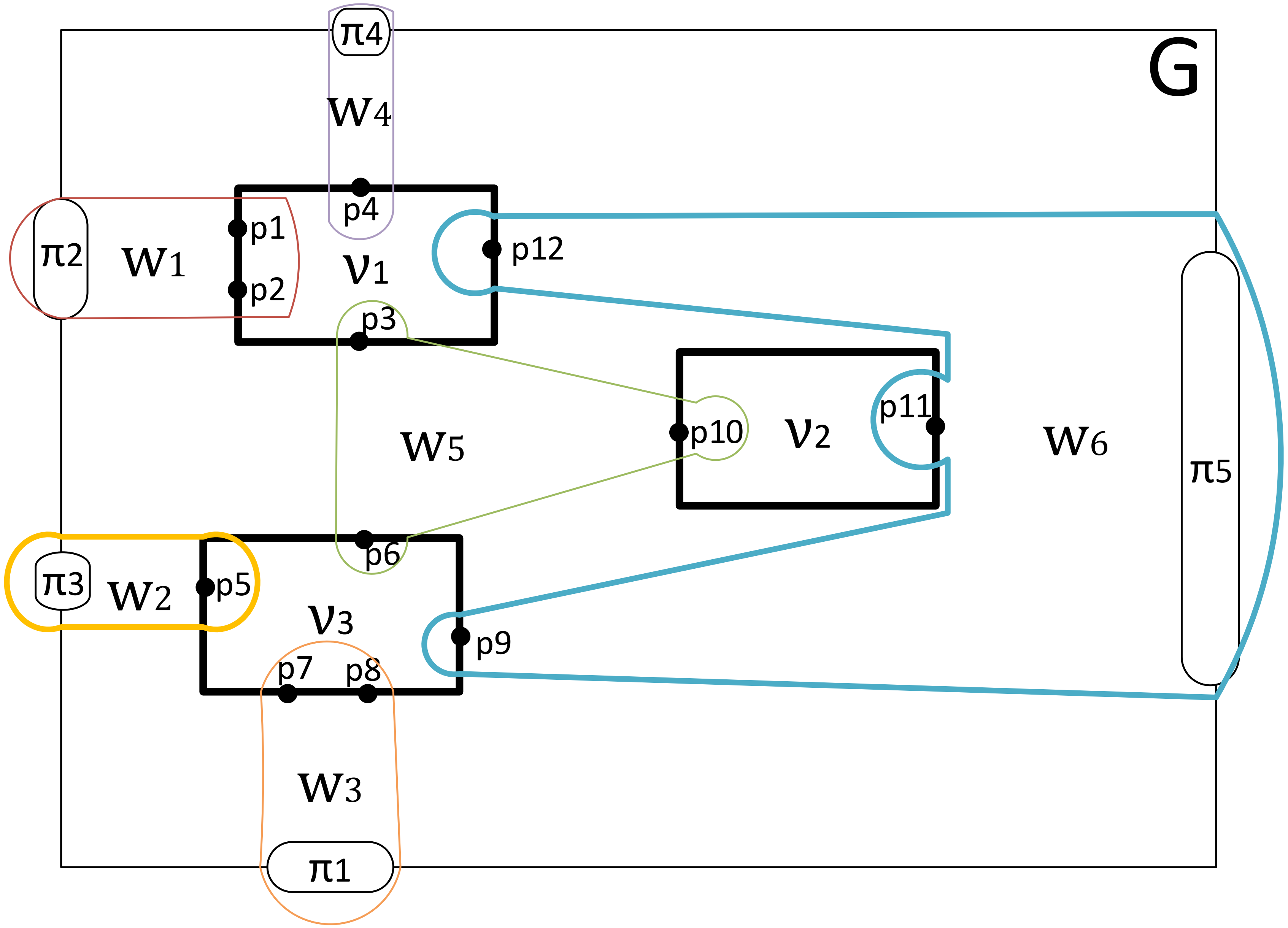
- ***P* = {π₁,…,πₙ}** is a set of external poles of the graph.
- $V = \{v_1,\ldots,v_m\}$ is a non-empty set of mutually disjoint vertices, consisting of internal poles.
- $W = \{w_1,\ldots,w_l\}$ is a set of hyperedges, consisting of poles.
- *Pol* is a set of **all** poles of the graph



GRAPH MODEL DEFINITION:
SETS OF HP-GRAPH

HP-graph is an ordered triple *G* = (*P*, *V*, *W*):

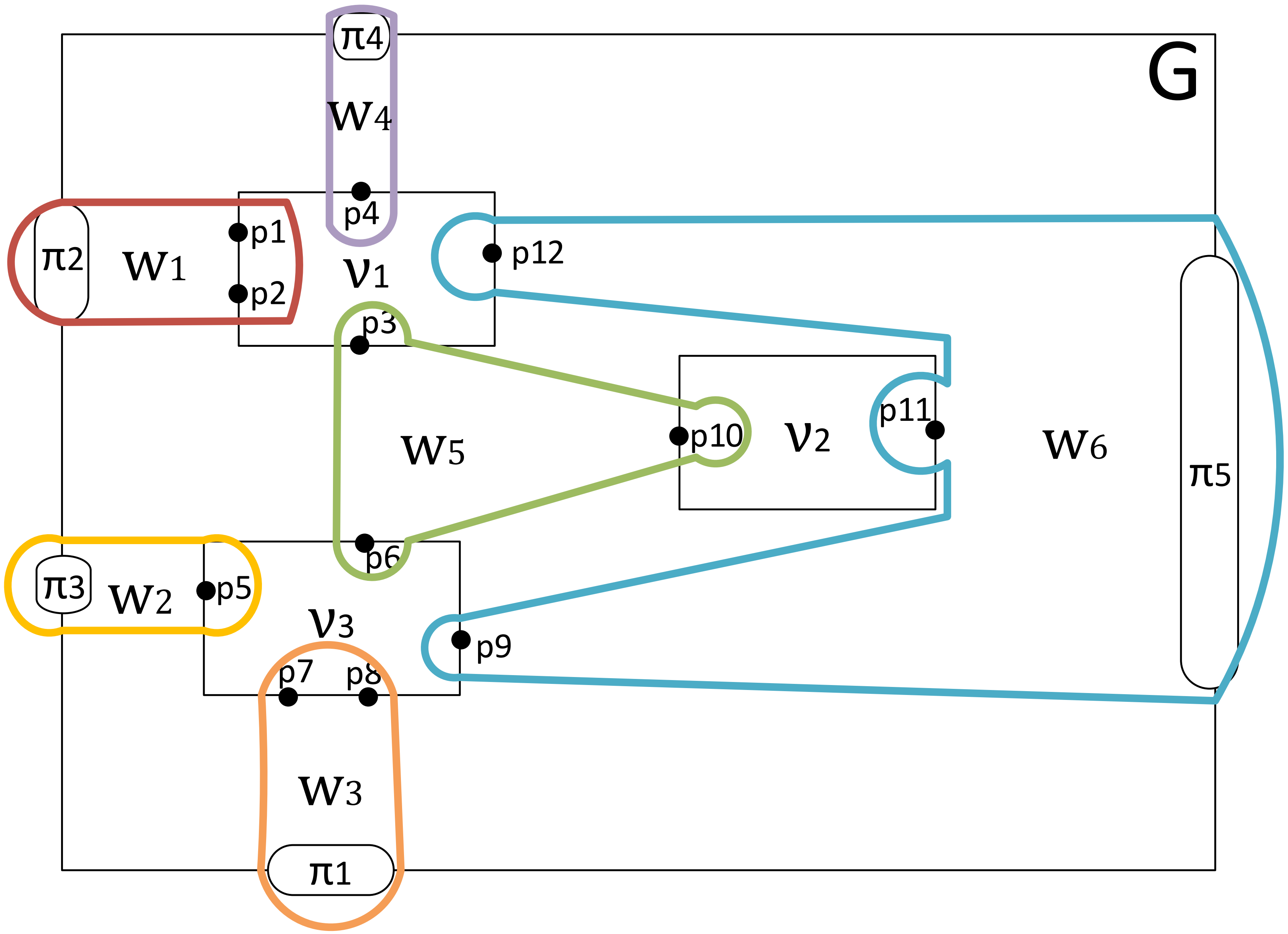
- $P = \{\pi_1,\dots,\pi_n\}$ is a set of external poles of the graph.
- $\boldsymbol{V = \{v_1,\dots,v_m\}}$ is a non-empty set of mutually disjoint vertices, consisting of internal poles.
- $W = \{w_1,\dots,w_l\}$ is a set of hyperedges, consisting of poles.
- *Pol* is a set of **all** poles of the graph



GRAPH MODEL DEFINITION:
SETS OF HP-GRAPH

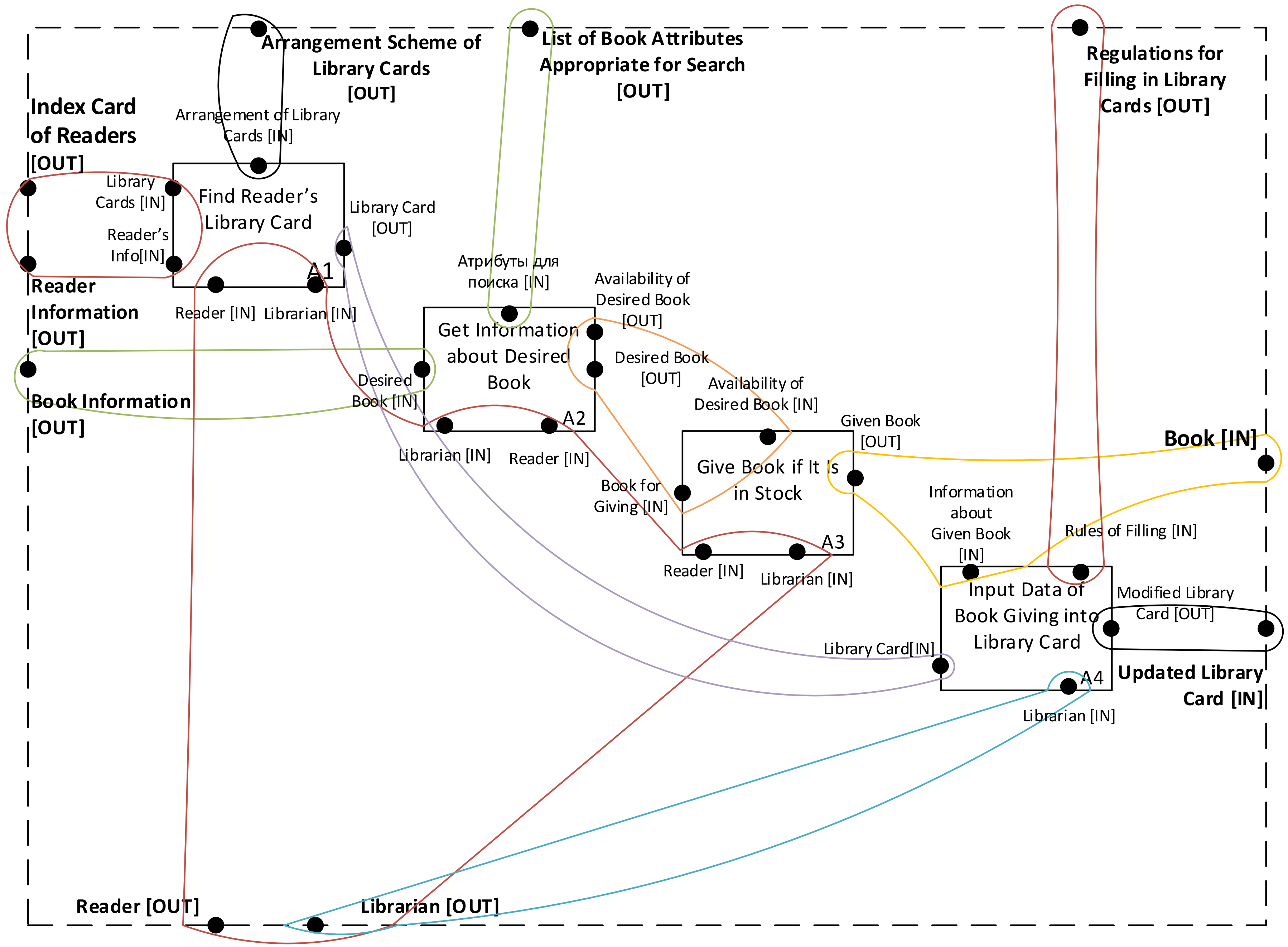
HP-graph is an ordered triple $G = (P, V, W)$:

- $P = \{\pi_1,\ldots,\pi_n\}$ is a set of external poles of the graph.
- $V = \{v_1,\ldots,v_m\}$ is a non-empty set of mutually disjoint vertices, consisting of internal poles.
- $\boldsymbol{W = \{w_1,\ldots,w_l\}}$ is a set of hyperedges, consisting of poles.
- *Pol* is a set of **all** poles of the graph



GRAPH MODEL DEFINITION:
INPUT AND OUTPUT POLES

- All the ***poles*** can be input or output (or both)
- Each ***vertex*** of the graph $v \in V$ is also represented by a set of *input* ($I(v)$) and *output* ($O(v)$) poles
- Each ***edge*** must contain at least one input pole and one output pole

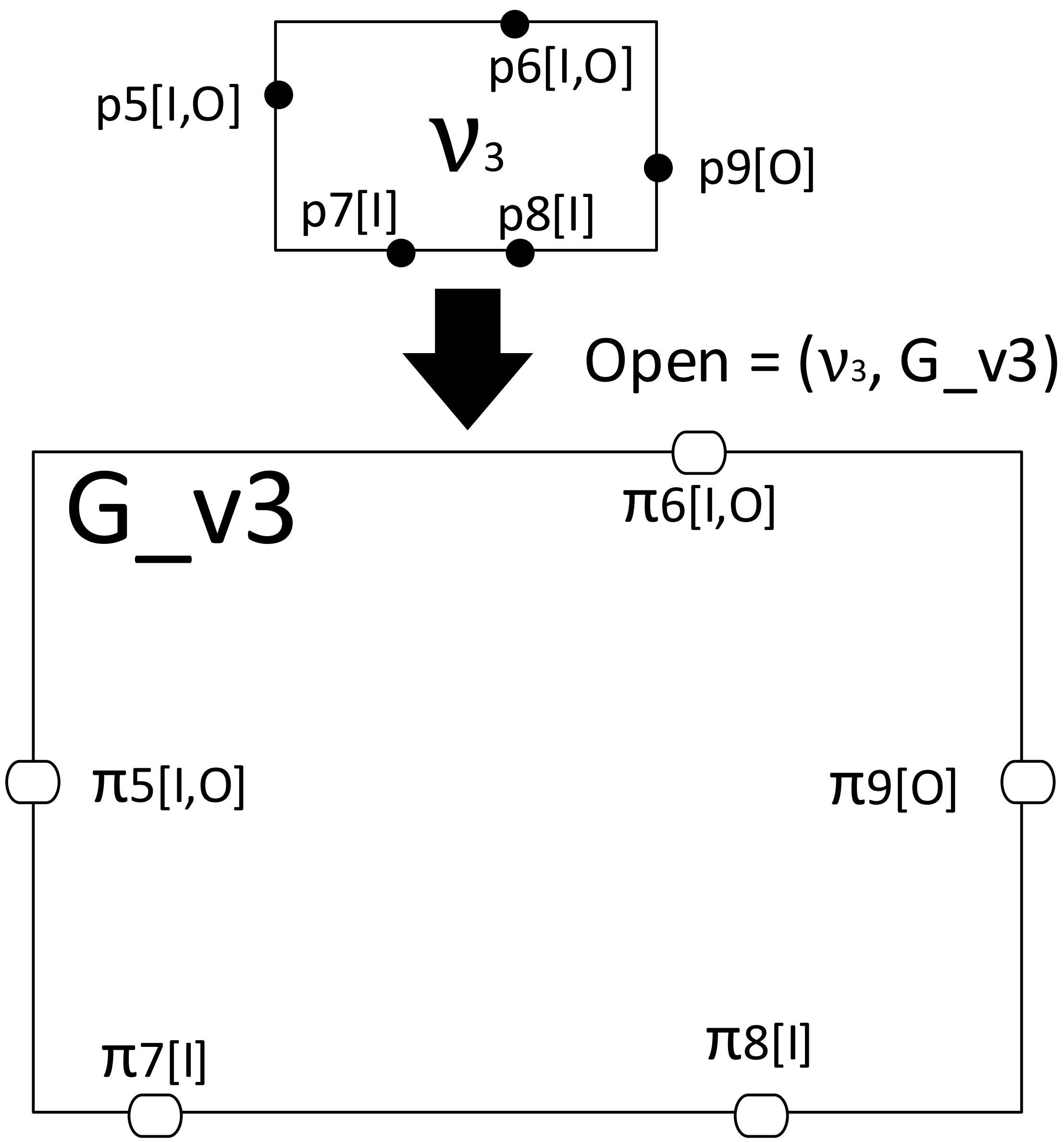


Business Process of Giving Book

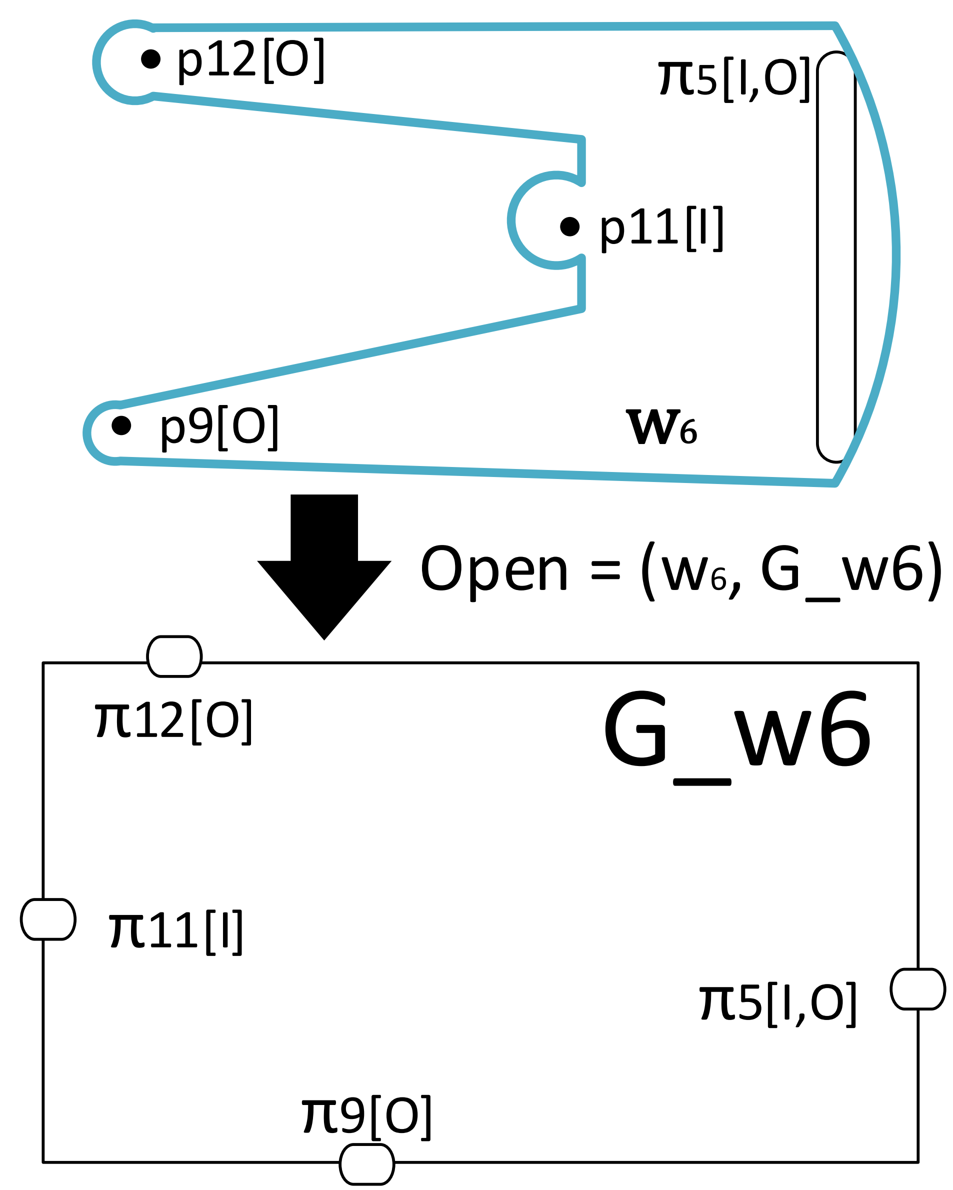
GRAPH MODEL DEFINITION:
BASIC OPERATIONS OF GRAPH EDITOR

Addition Operations	Removal Operations
1. v + p1 – Addition of the inner pole to the vertex	1. v − p1 – Removal of the inner pole from the node
2. G + v – Addition of the vertex to the graph	2. G − v – Removal of the node from the graph
3. G + w – Addition of the edge to the graph	3. G − w – Removal of the edge from the graph
4. w + p1 – Addition of the inner pole to the edge	4. w − p1 – Removal of the inner pole from the edge
5. w + p2 – Addition of the outer pole to the edge	5. w − p2 – Removal of the external pole from the edge
6. G + p2 – Addition of the outer pole to the graph	6. G − p2 – Removal of the outer pole from the graph

GRAPH MODEL DEFINITION:
DECOMPOSITION BY A NEW GRAPH



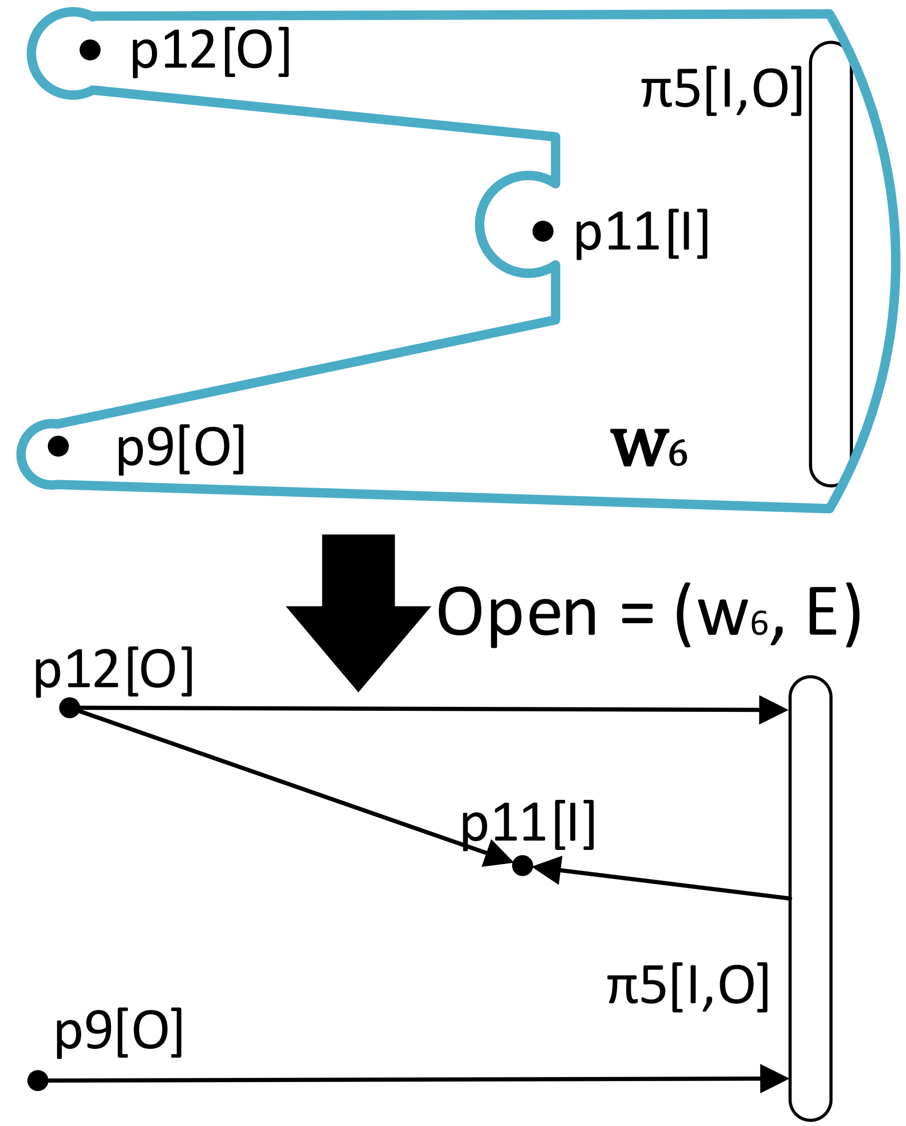
Vertex Decomposition by a new HP-graph



Edge Decomposition by a new HP-graph

GRAPH MODEL DEFINITION:
DECOMPOSITION BY A NEW GRAPH

- Every edge can be opened by ordinary links between poles.
- For every edge $w \in W$, a set of links is defined: $E_w = \{e_1, \ldots, e_n\} \subset I(w) \times O(w)$, where every link is a pair (p, r) provided that $p \in I(w)$, $r \in O(w)$.
- As $E_w \subset I(w) \times O(w)$, some input and output poles can be unconnected such as poles *p*9[*O*] and *p*11[*I*]



Edge Decomposition by ordinary links

COMPARING GRAPH MODELS

Graph model	Representation in the HP-graph $G' = (P', V', W')$
Oriented Graph $G = (V, E)$	$V = P' = V'$, where $\forall v' \in V'$: $[\lvert v' \rvert = 1]$ $E = W'$, where $\forall w' \in W'$: $[\lvert w' \rvert = 2])$
Hypergraph $G = (X, E)$	$X = P' = V'$, where $\forall v' \in V'$: $[\lvert v' \rvert = 1]$ $E = W'$
Hi-graph $G = (X, E)$	$\{x \mid x \in X \;\&\; \lvert x \rvert = 1\} = P' = V'$, where $\forall v' \in V'$: $[\lvert v' \rvert = 1]$ $E \cup \{x \mid x \in X \;\&\; \lvert x \rvert > 1\} = W'$
Metagraph $G = (V, MV, E)$	$V = P' = V'$, where $\forall v' \in V'$: $[\lvert v' \rvert = 1]$ $E \cup MV = W'$
P-graph $G = (P, V, W)$	$P = P'$ $V = V'$ $W = W'$, where $\forall w' \in W'$: $[\lvert w' \rvert = 2]$

EXAMPLES OF USING: ALGORITHMS

- As is seen, the decomposition of edges and vertices is almost equal, therefore, it is possible to define **a common opening algorithm** for these structures.
- Let us define a set of structures $Str = V \cup W$.
- Hence, str $\in$ Str is a *structure* which can be either a vertex or an edge.
- For every structure *str* several decoding operations can be defined:

$$Open_{str} \subset str \times G_{all}$$

Procedure DecomposeStructure

G = *new HPGraph*();
foreach $p \in str$:
    if ($p \in I(str)$):
        $I(G) = I(G) \cup p$;
    if ($p \in O(str)$):
        $O(G) = O(G) \cup p$;
      $Open_{str} = Open_{str} \cup (str, G)$

EXAMPLES OF USING: ALGORITHMS

Let us define a subgraph of an HP-graph:

A *subgraph* of the HP graph $G = (P, V, W)$ is an HP-graph $G' = (P', V', W')$ that is part of the graph G and fulfills the condition $Open' \subset Open$.

A subgraph can contain vertices called *incomplete* (partials) whose sets of poles can only be part of the sets of poles of the vertices of the original graph.

Transformation can be divided into 2 parts:

- **Removal of a subgraph, isomorphic to a pattern**
- Addition of a replacement graph to the original graph

Function DeleteGraph(HostG, G_L)

```
G' = Find_Isomorphic_Subgraph(HostG, G_L);
partials = {}
foreach w'∈W(G'):
    W(HostG) = W(HostG) \ {w'};
foreach v'∈V(G'):
    if (v'∈V(HostG)):
        V(HostG) = V(HostG) \ {v'};
    else:
        partials = partials ∪ {v'};
foreach p'∈P(G'):
    if (¬∃w∈W(HostG)[p'∈w]):
        P(HostG) = P(HostG)\p';
return partials;
```

EXAMPLES OF USING: ALGORITHMS

Let us define a subgraph of an HP-graph:

A *subgraph* of the HP graph $G = (P, V, W)$ is an HP-graph $G' = (P', V', W')$ that is part of the graph G and fulfills the condition $Open' \subset Open$.

A subgraph can contain vertices called *incomplete* (partials) whose sets of poles can only be part of the sets of poles of the vertices of the original graph.

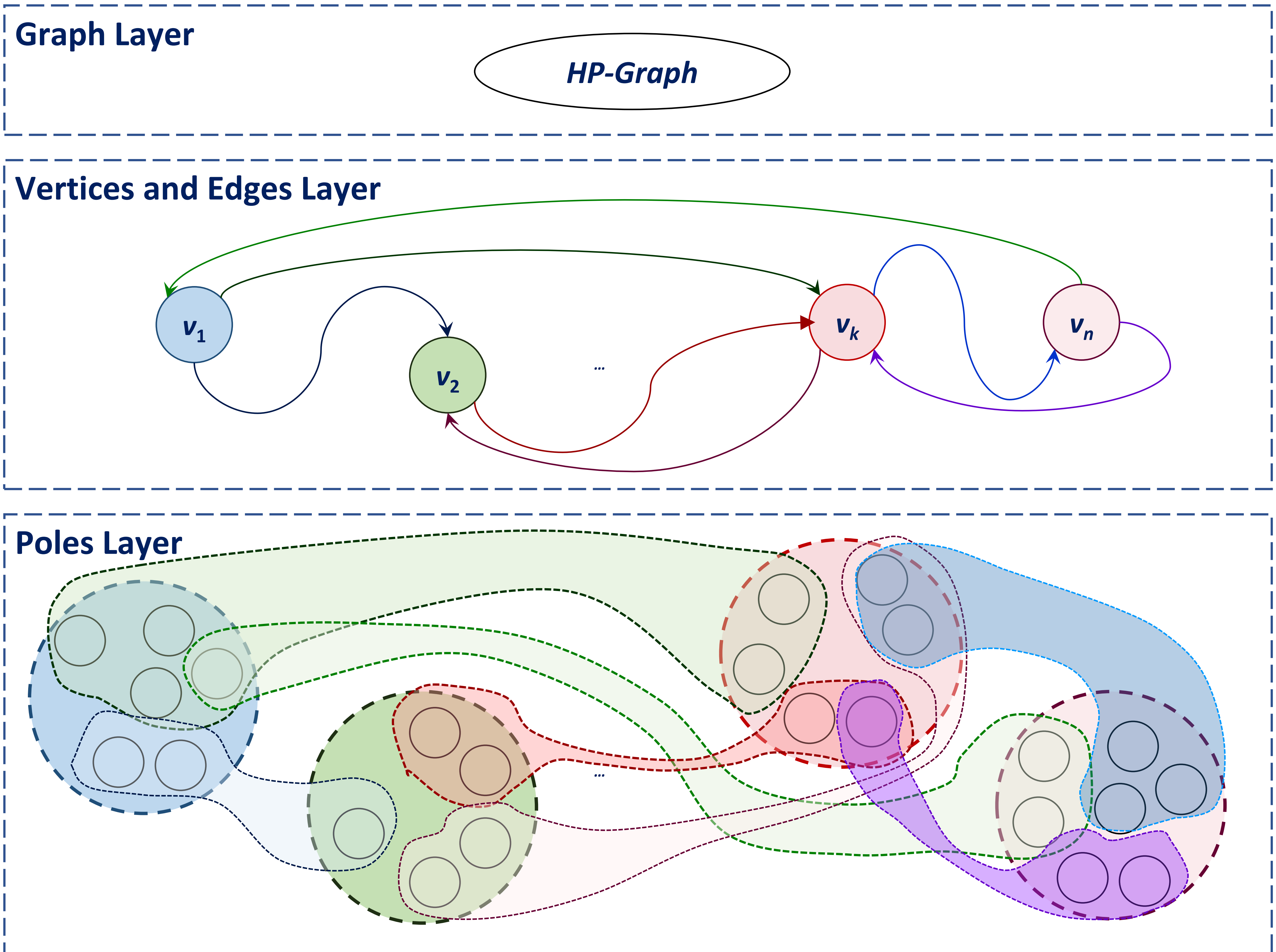
Transformation can be divided into 2 parts:

- Removal of a subgraph, isomorphic to a pattern
- **Addition of a replacement graph to the original graph**

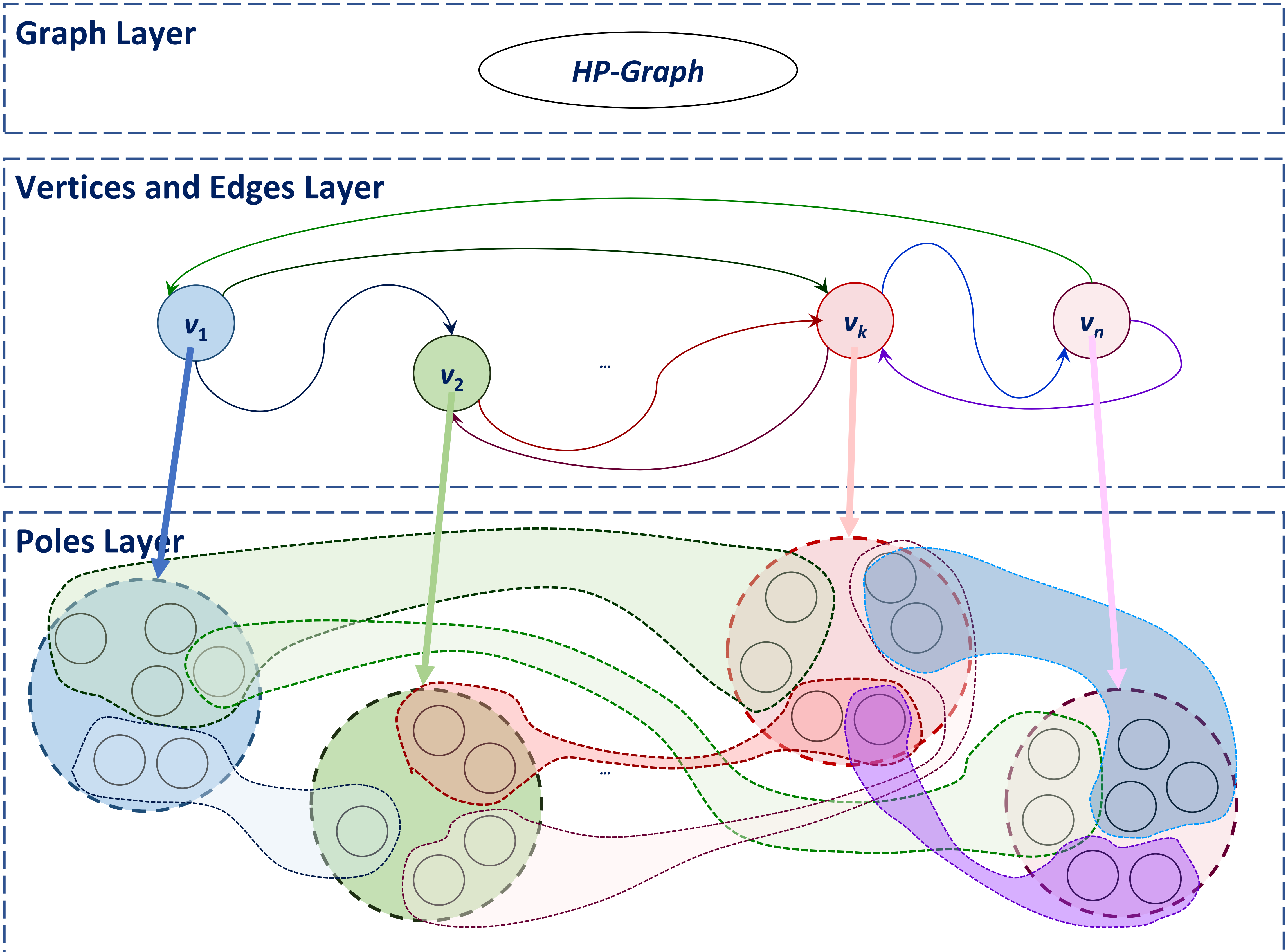
Procedure AddGraph(HostG, G_R, partials)

foreach $p \in P(G_R)$:
  if $p \notin P(HostG)$:
    $P(HostG) = P(HostG) \cup \{p\}$;
foreach $v \in V(G_R)$:
  if $(v \notin Partials)$:
    $V(HostG) = V(HostG) \cup \{v\}$;
foreach $w \in W(G_R)$:
    $W(HostG) = W(HostG) \cup \{w\}$;

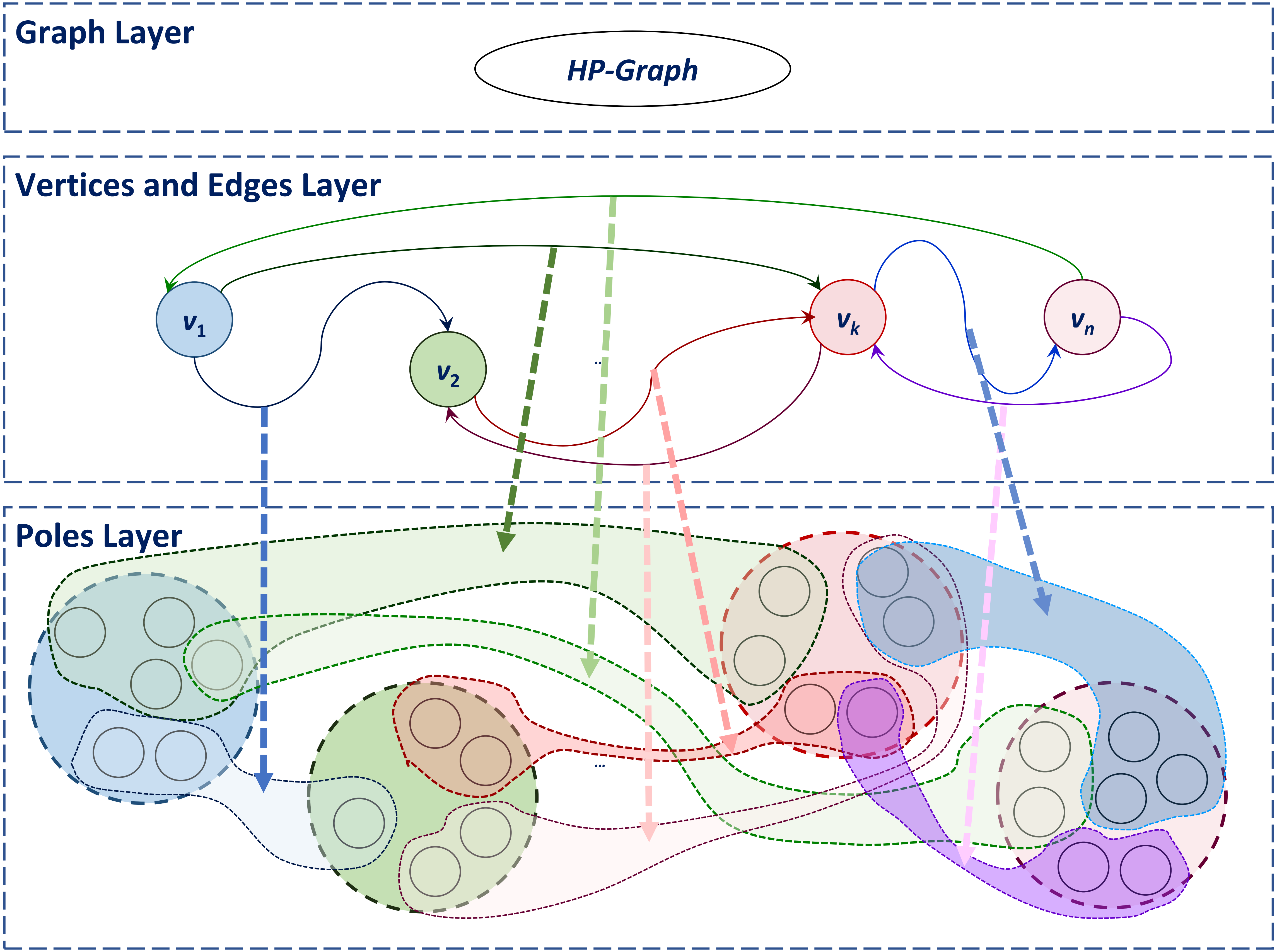
TIME COMPLEXITY:
LAYER STRUCTURE FOR OPTIMIZING ALGORITHMS



TIME COMPLEXITY:
LAYER STRUCTURE FOR OPTIMIZING ALGORITHMS



TIME COMPLEXITY:
LAYER STRUCTURE FOR OPTIMIZING ALGORITHMS



TIME COMPLEXITY:
LAYER STRUCTURE FOR OPTIMIZING ALGORITHMS

Graph Layer

HP-Graph

Vertices and Edges Layer

Pol

A	v_1	v_2	…	v_j	…	$v_{n\text{-}1}$	v_n
v_1	a_{11}	a_{12}	…	$a_{1,j}$	…	$a_{1,n\text{-}1}$	$a_{1,n}$
…	…	…	…	…	…	…	…
v_i	$a_{i,1}$	$a_{i,2}$	…	$a_{i,j}$	…	$a_{2,n\text{-}1}$	$a_{2,n}$
…	…	…	…	…	…	…	…
v_n	$a_{n,1}$	$a_{n,2}$	…	$a_{n,j}$	…	$a_{n,n\text{-}1}$	$a_{n,n}$

TIME COMPLEXITY:
LAYER STRUCTURE FOR OPTIMIZING ALGORITHMS

Graph Layer

HP-Graph

Vertices and Edges Layer

A	v_1	v_2	…	v_j	…	v_{n-1}	v_n
v_1	a_{11}	a_{12}	…	$a_{1,j}$	…	$a_{1,n-1}$	$a_{1,n}$
…		…	…	…	…	…	…
v_i	$a_{i,1}$	$a_{i,2}$	…	$a_{i,j}$	…	$a_{2,n-1}$	$a_{2,n}$
…	…	…					…
v_n	$a_{n,1}$	$a_{n,2}$	…	$a_{n,j}$	…	$a_{n,n-1}$	$a_{n,n}$

```
G = HostG();
n = |V(G)|;
A = New Matrix([n, n]);
    for i = 1 to n step 1:
        for j = 1 to n step 1:
            for each w∈W(G):
                a_{i,j} = P(v_i) ∩ P(w) ≠ ø and P(v_j) ∩ P(w) ≠ ø
return A;
```

Pol

TIME COMPLEXITY:
LAYER STRUCTURE FOR OPTIMIZING ALGORITHMS

Graph Layer

HP-Graph

Vertices and Edges Layer

A	v_1	v_2	…	v_j	…	v_{n-1}	v_n
v_1	[illegible]	a_{12}	…	$a_{1,j}$	…	$a_{1,n-1}$	$a_{1,n}$
…	…	…	…	…	…	…	…
v_i	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$a_{2,n}$
…	…	…	…	…	…	…	…
v_n	$a_{n,1}$	$a_{n,2}$	…	$a_{n,j}$	…	$a_{n,n-1}$	$a_{n,n}$

Pol

```
G = HostG();
n = |V(G)|;
A = New Matrix([n, n]);
    for i = 1 to n step 1:
        for j = 1 to n step 1:
            for each w∈W(G):
                a_{i,j} = O(v_i) ∩ P(w) ≠ ø and I(v_j) ∩ P(w) ≠ ø
return A;
```

CONCLUSION

- The definition of the mathematical apparatus underlying the visual model editor was given above.
- Algorithms for decoding vertices and edges, as well as algorithms for performing transformations, were described.
- The HP-graph unites expressive possibilities of various types of graphs, therefore, algorithms that are designed for these types of graphs can also be implemented for HP graphs.
- The time complexity of model transformation algorithms can be reduced.

The paper proves that HP-graph allows the creation of a flexible visual model editor based on this graph formalism for a DSM-platform. Representing both vertices and links as sets of poles simplifies the object model of DSM editor and visual model editing algorithms.

THANKS FOR ATTENTION!

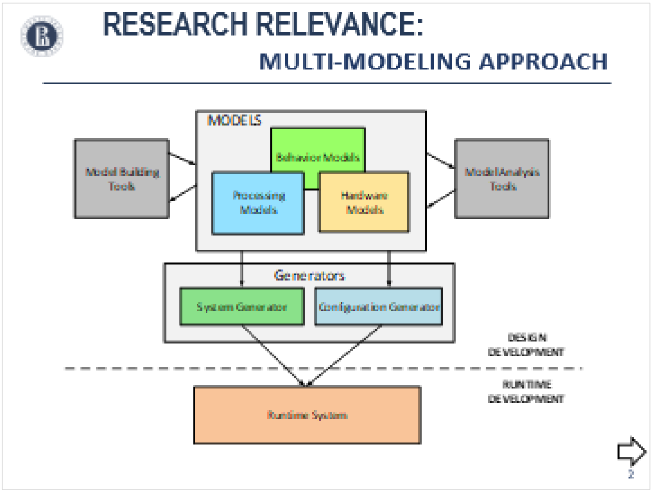
Suvorov Nikolai Mikhailovich

E-mail: *SuvorovNM@gmail.com*

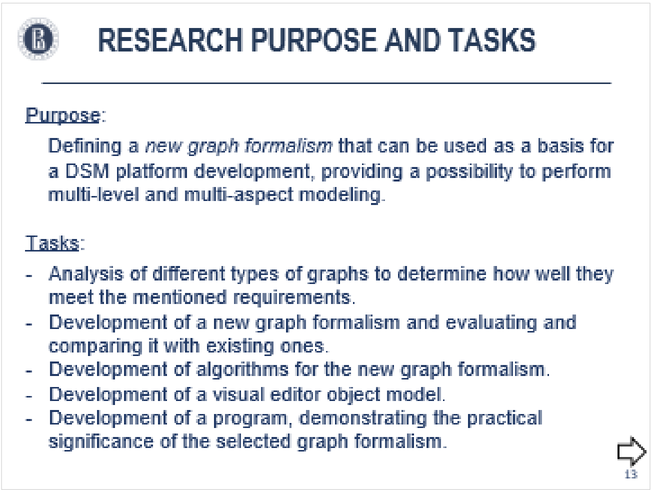
Lyadova Lyudmila Nickolaevna

E-mail: *LNLyadova@gmail.com*

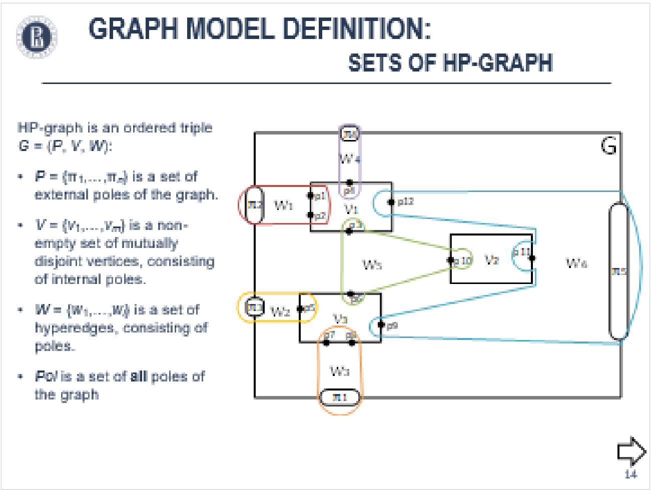
2

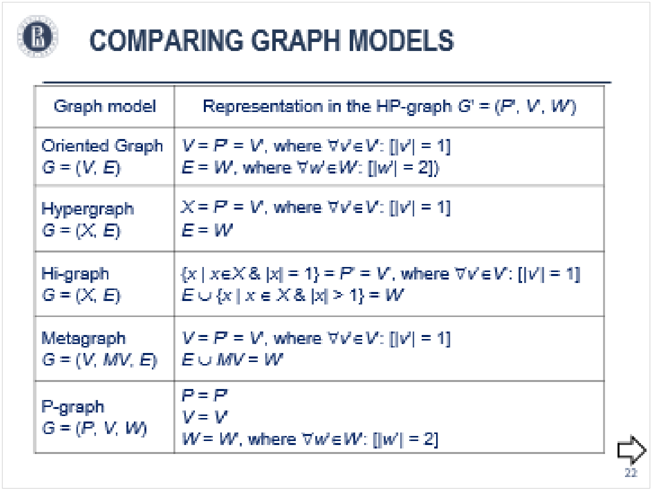


12

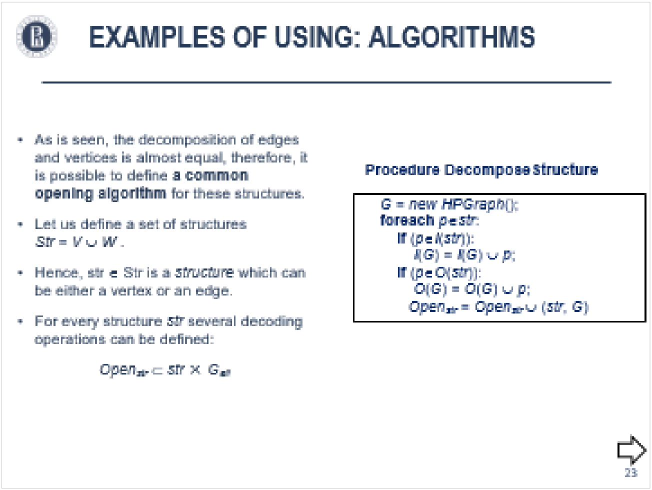


13

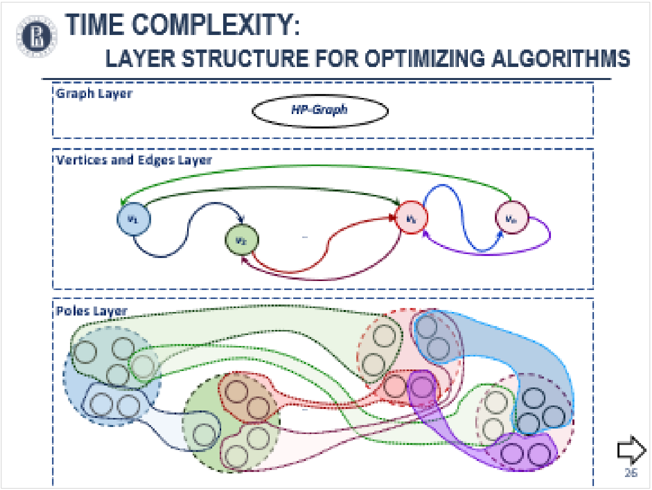




21

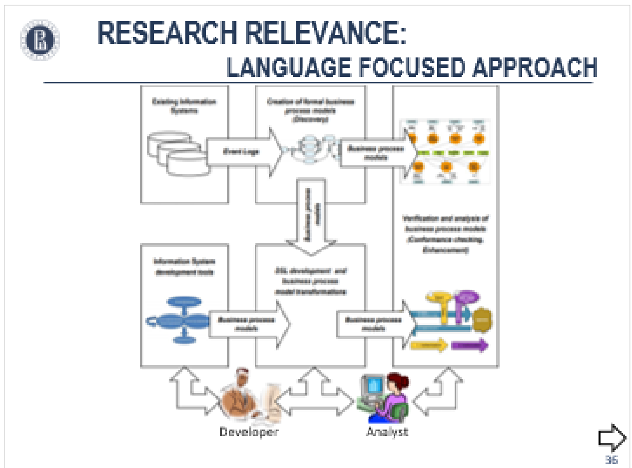


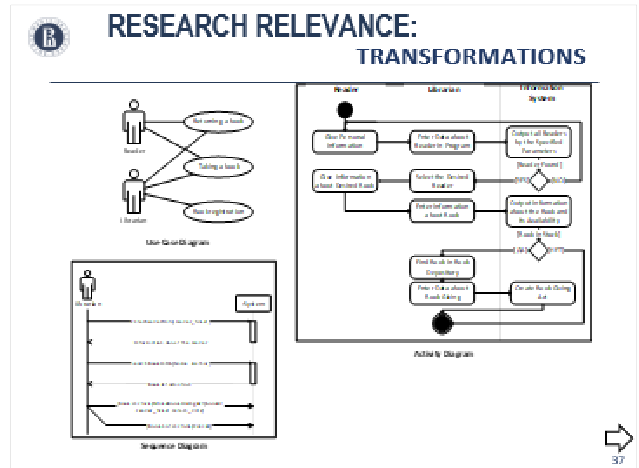
22

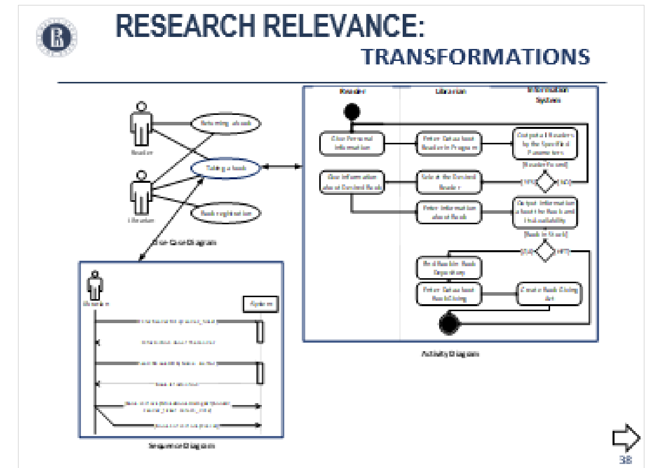


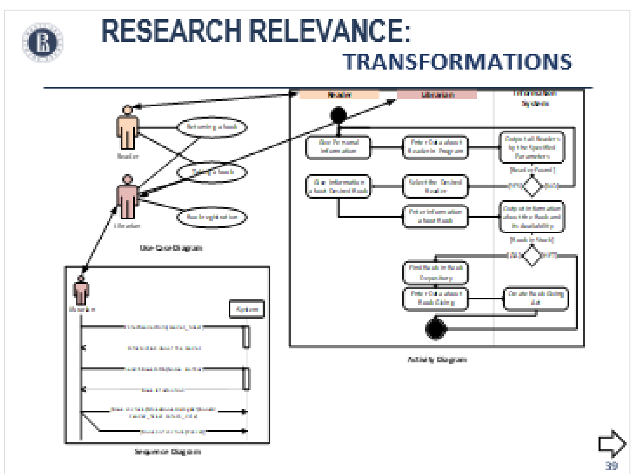
25

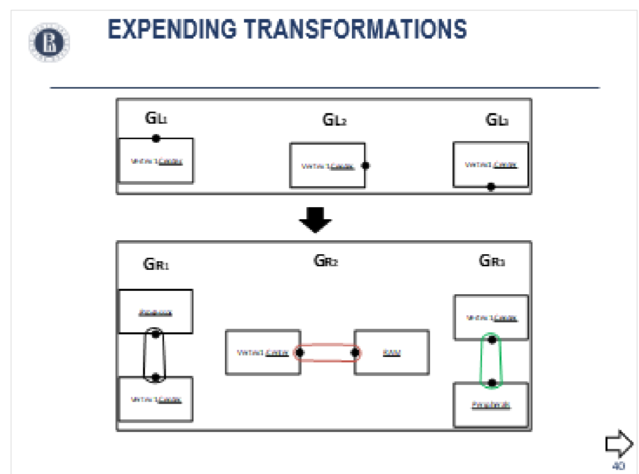
APPENDIX…

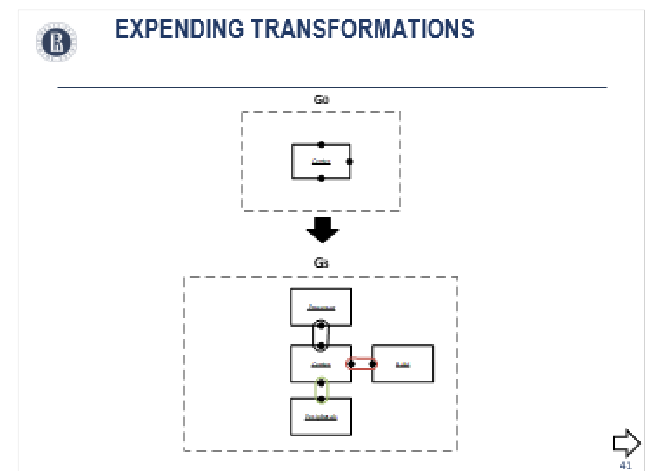

RESEARCH RELEVANCE:
LANGUAGE FOCUSED APPROACH
Developer
Analyst
36

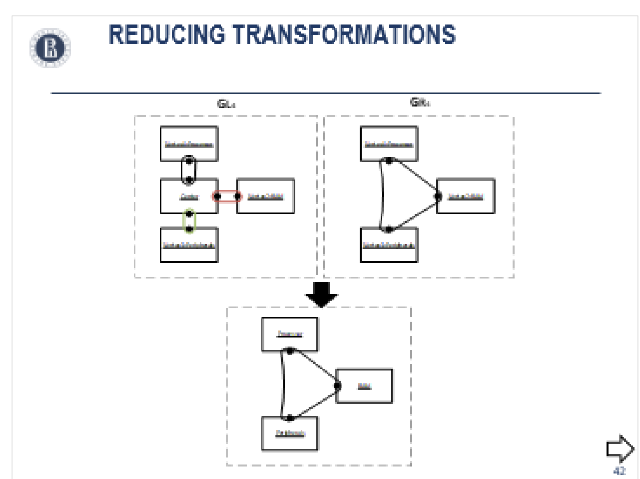

RESEARCH RELEVANCE:
TRANSFORMATIONS
37


RESEARCH RELEVANCE:
TRANSFORMATIONS
38

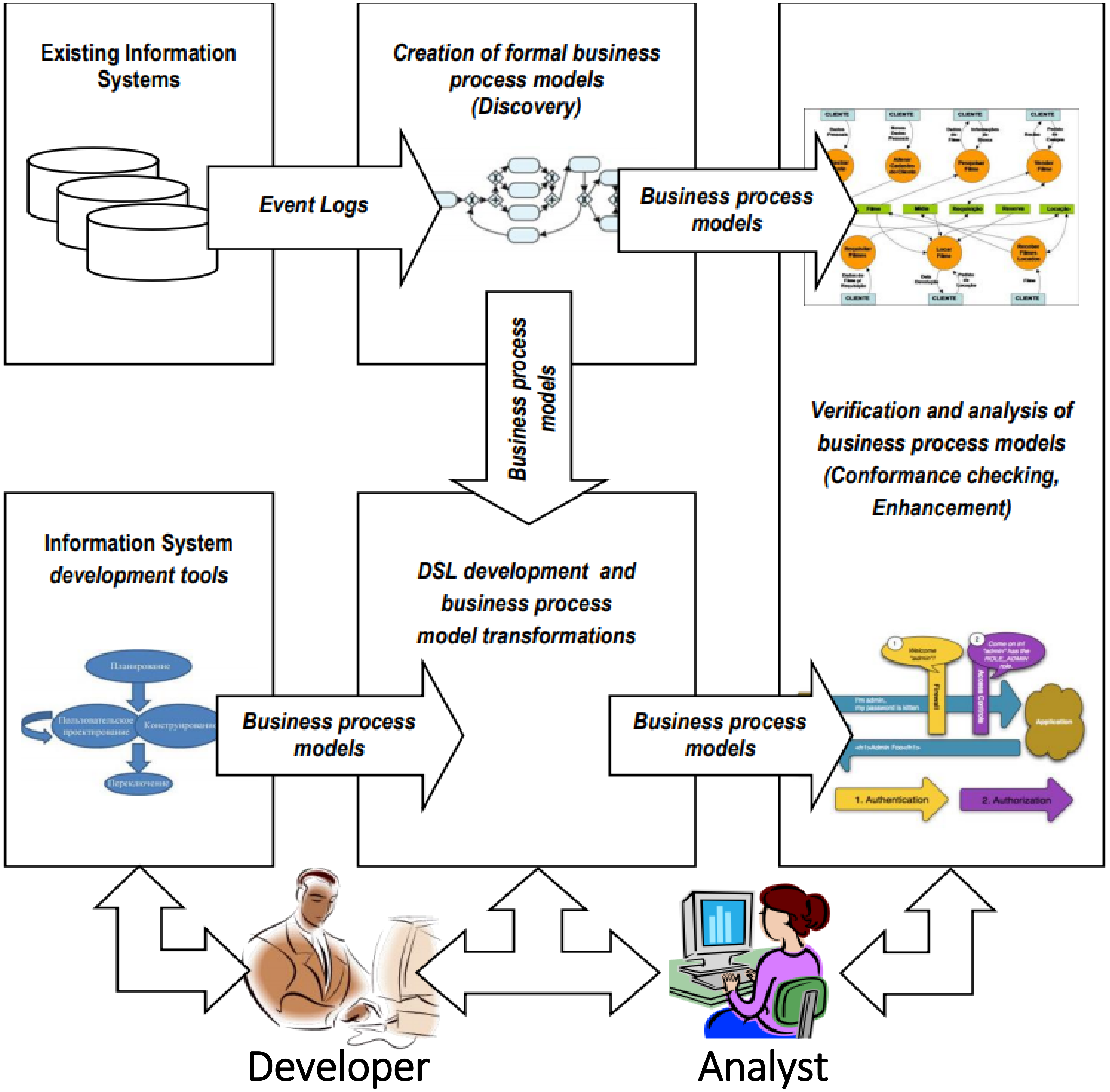

RESEARCH RELEVANCE:
TRANSFORMATIONS
39


EXPENDING TRANSFORMATIONS
40

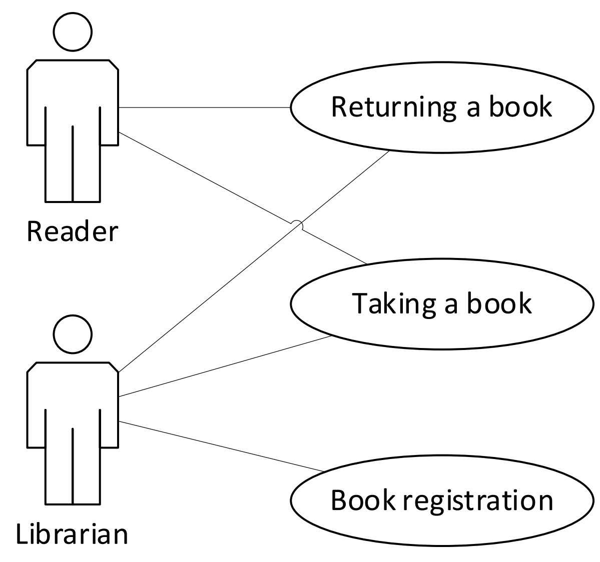

EXPENDING TRANSFORMATIONS
41


REDUCING TRANSFORMATIONS
42

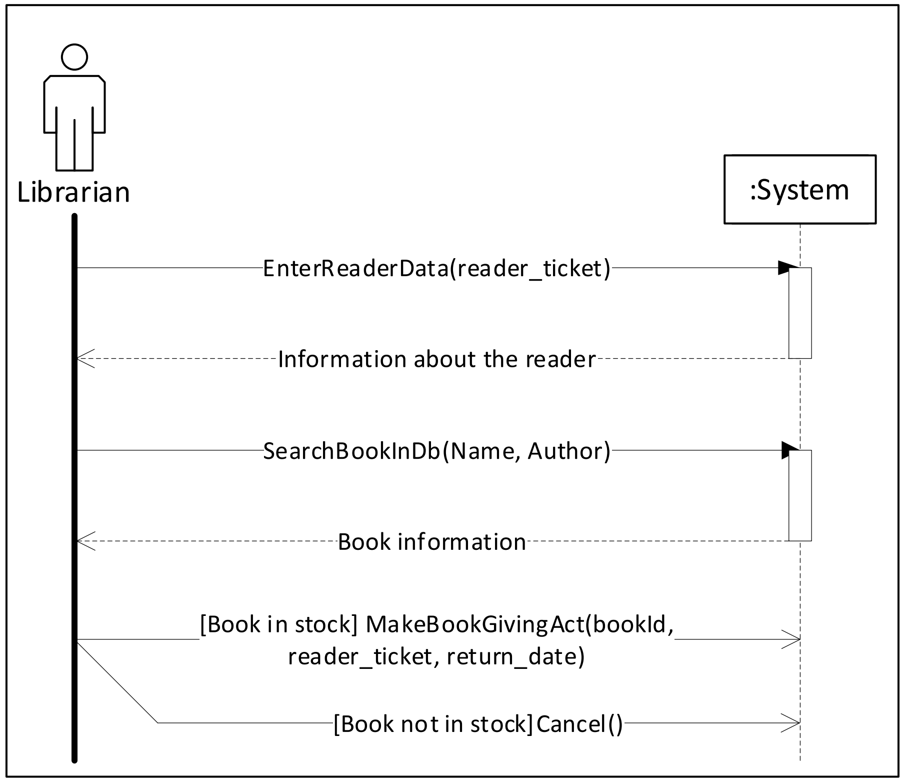
RESEARCH RELEVANCE:
LANGUAGE FOCUSED APPROACH



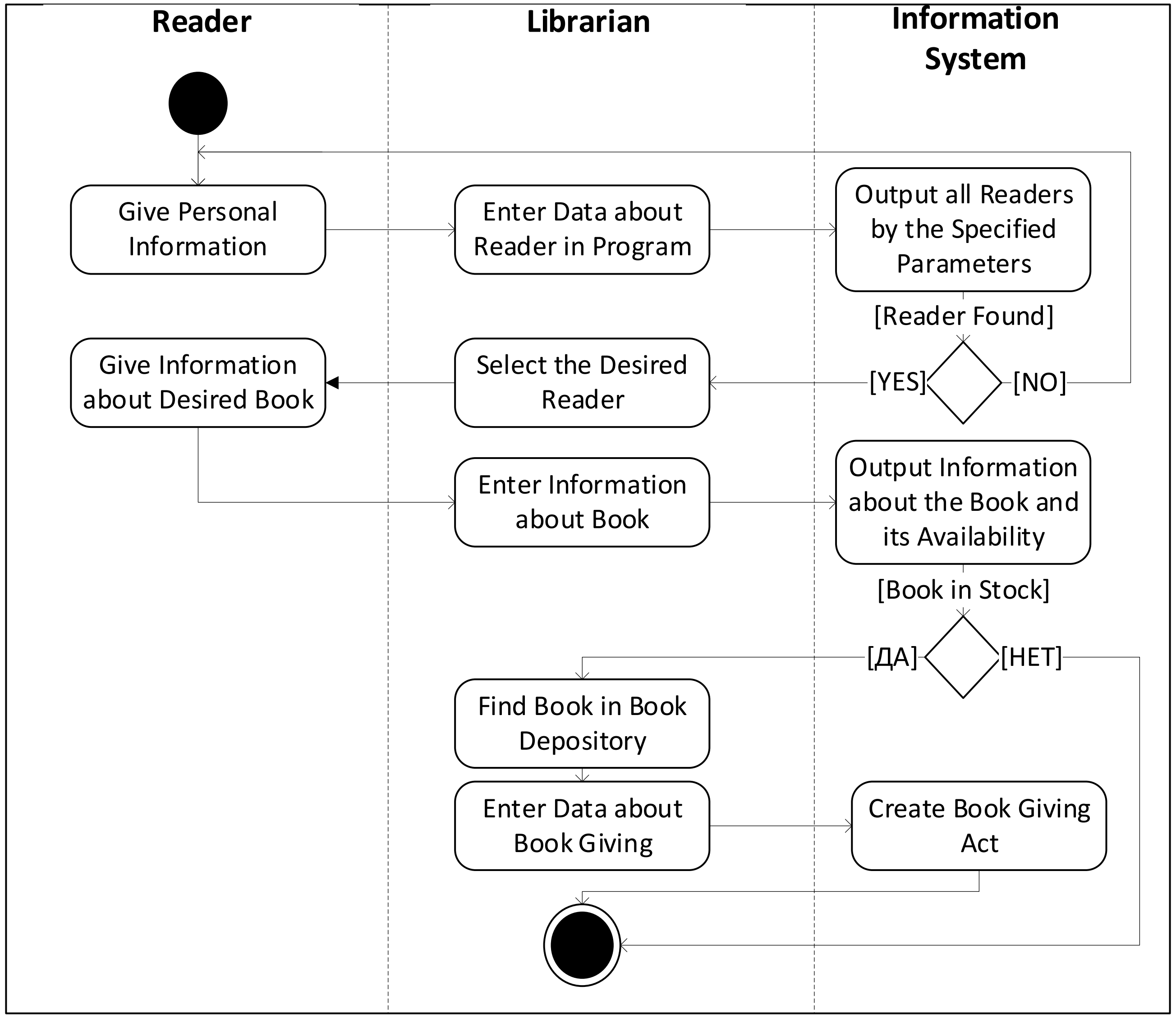
RESEARCH RELEVANCE: TRANSFORMATIONS



Use-Case Diagram

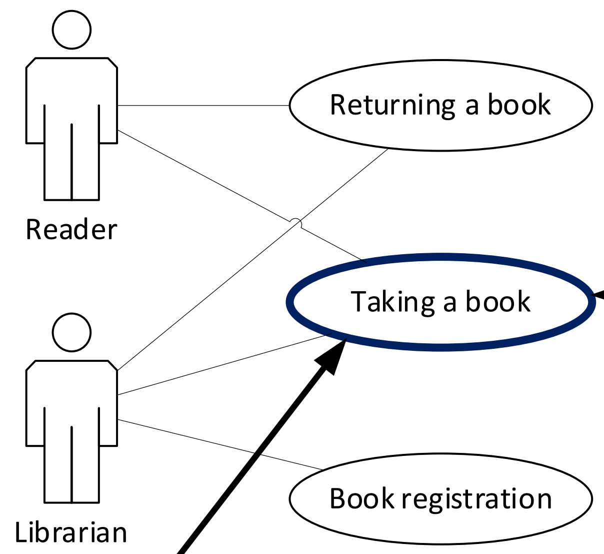


Sequence Diagram

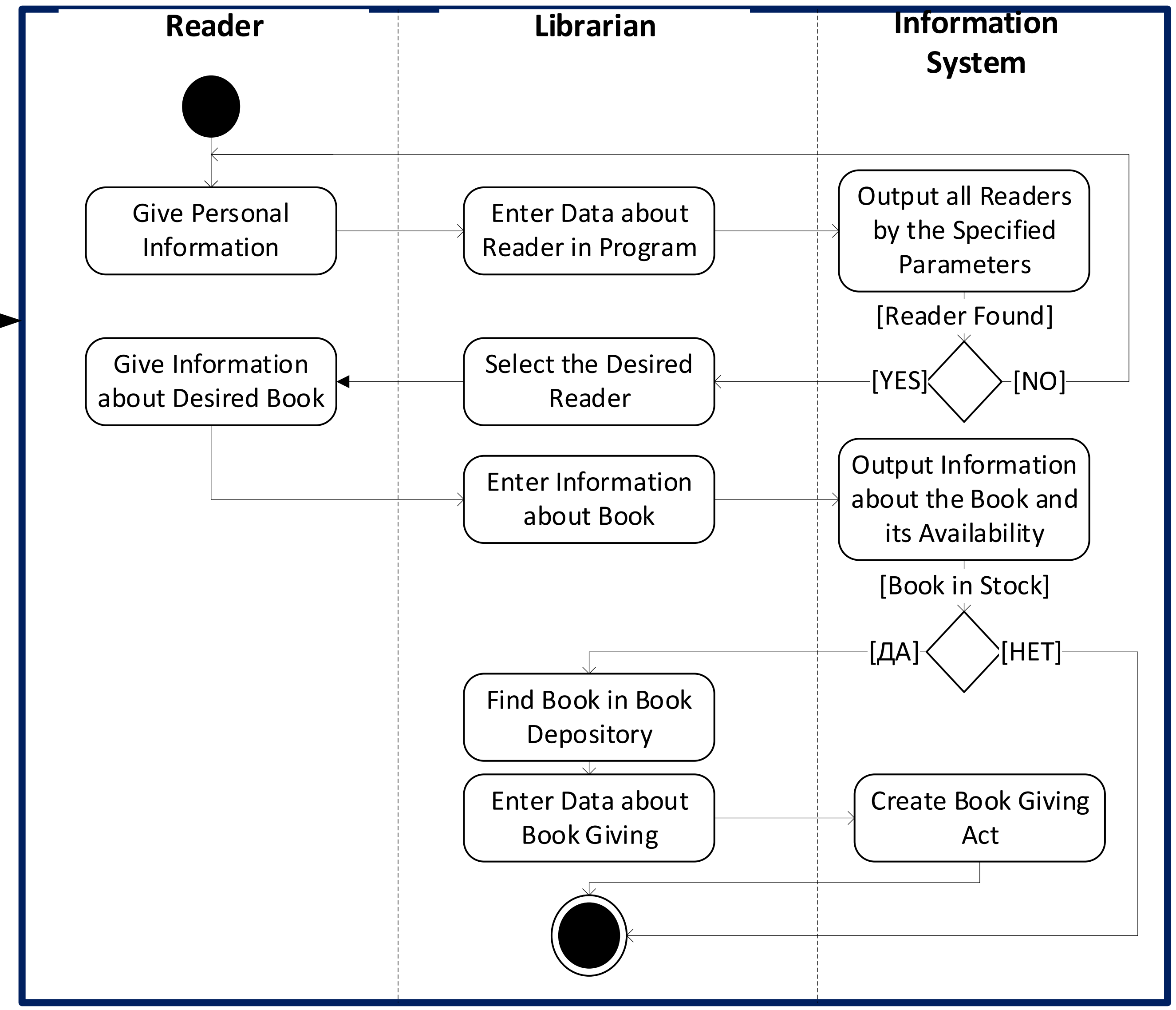


Activity Diagram

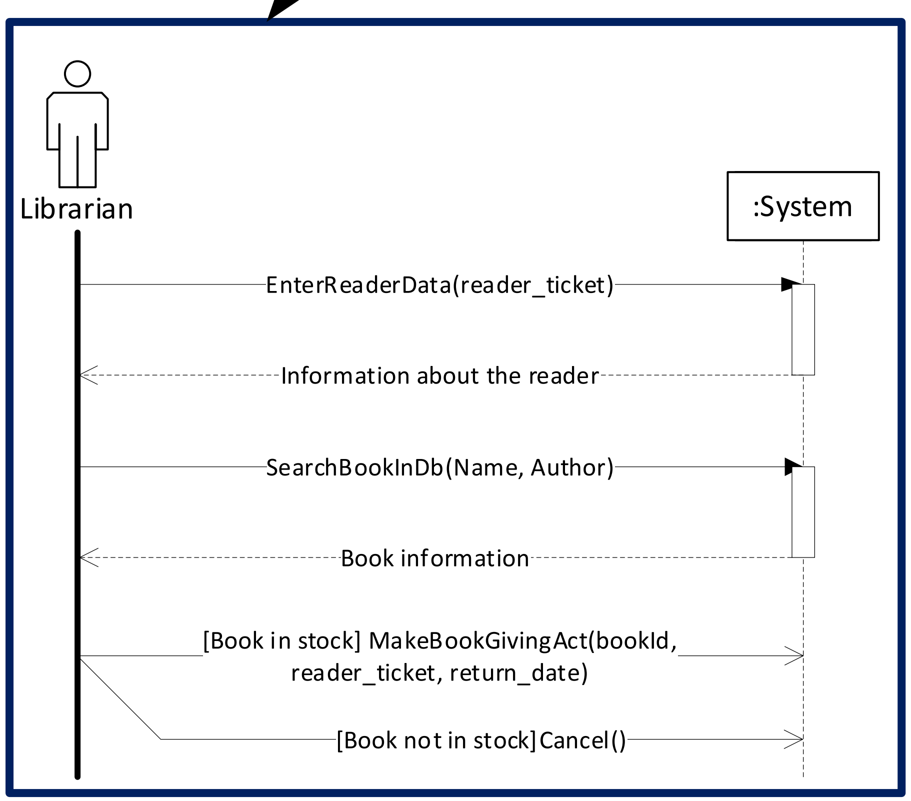
RESEARCH RELEVANCE: TRANSFORMATIONS



Use-Case Diagram

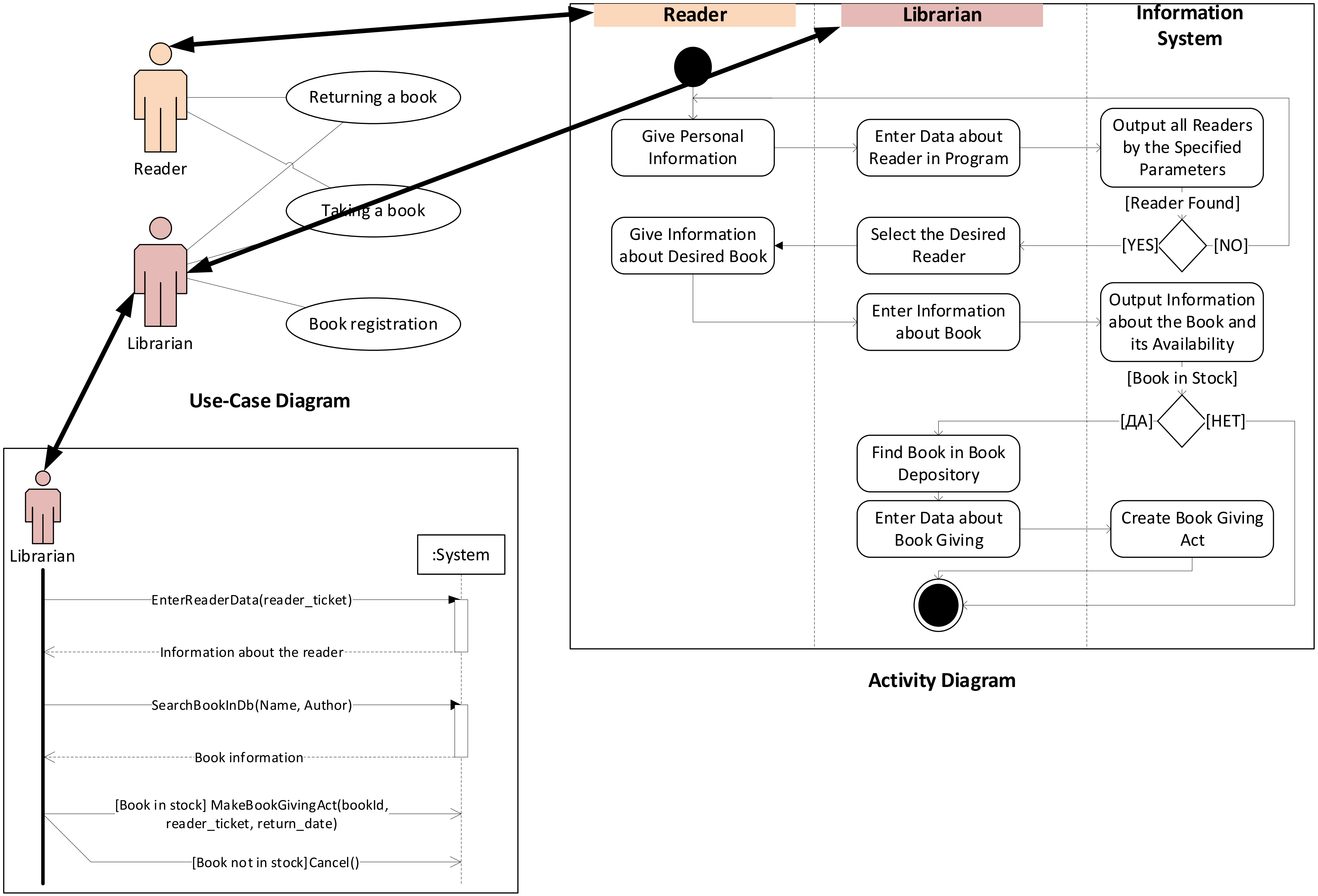


Activity Diagram



Sequence Diagram

RESEARCH RELEVANCE: TRANSFORMATIONS

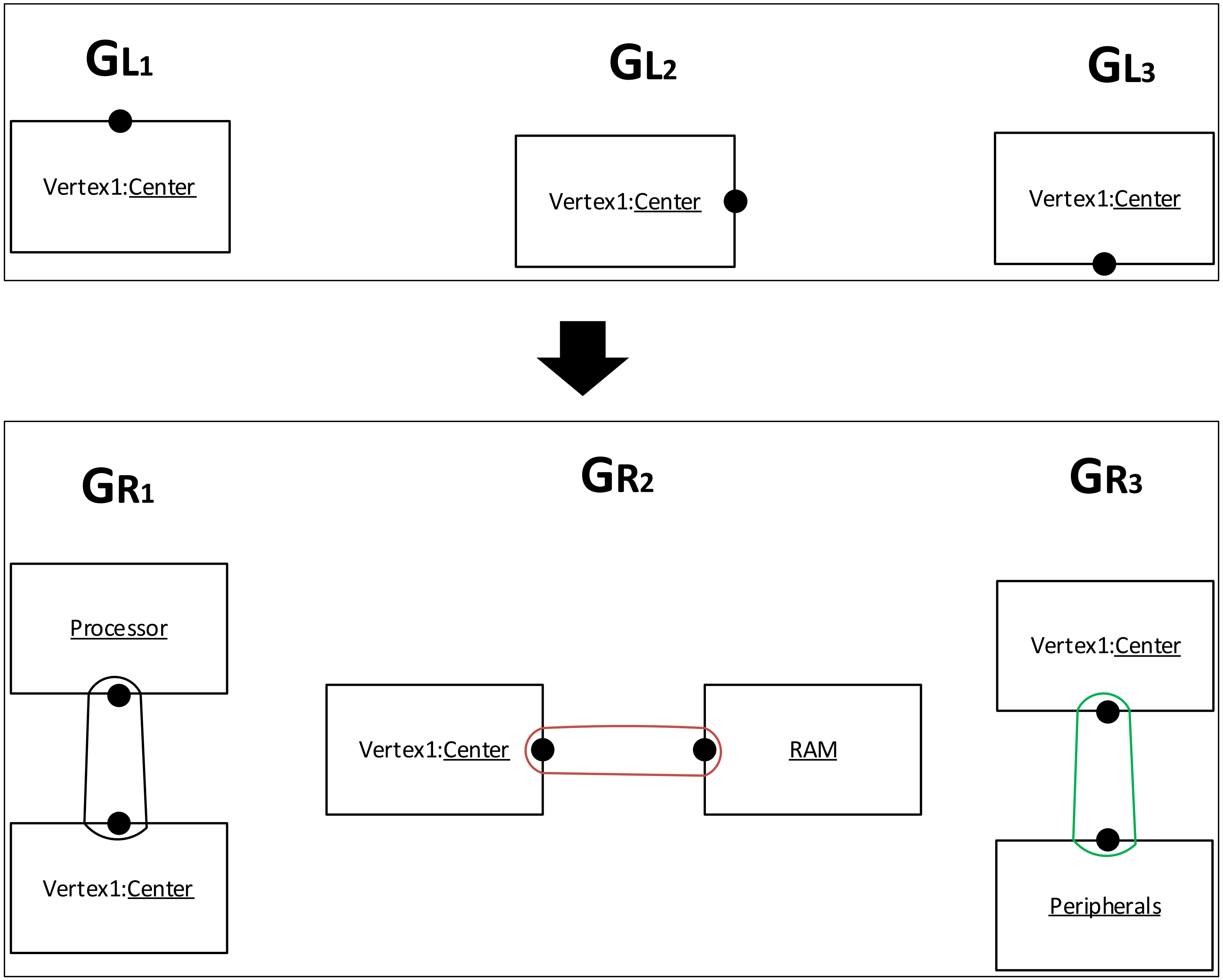


Use-Case Diagram

Activity Diagram

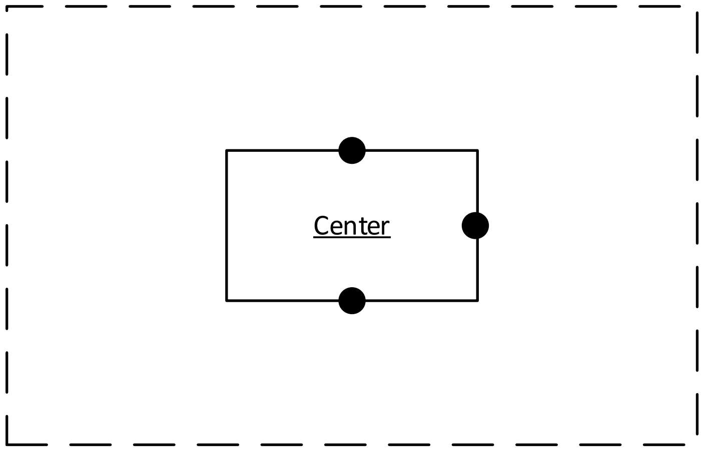
Sequence Diagram

EXPENDING TRANSFORMATIONS

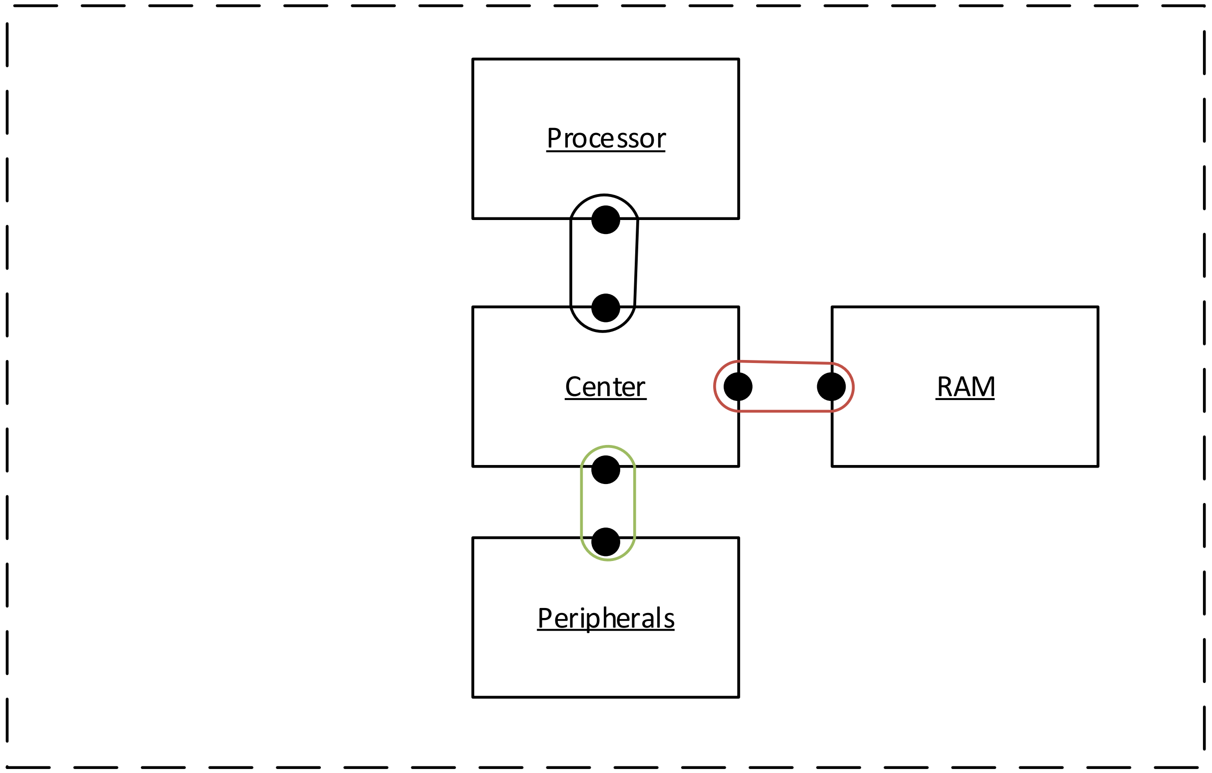


EXPENDING TRANSFORMATIONS

G0



G3



REDUCING TRANSFORMATIONS