National Research University Higher School of Economics

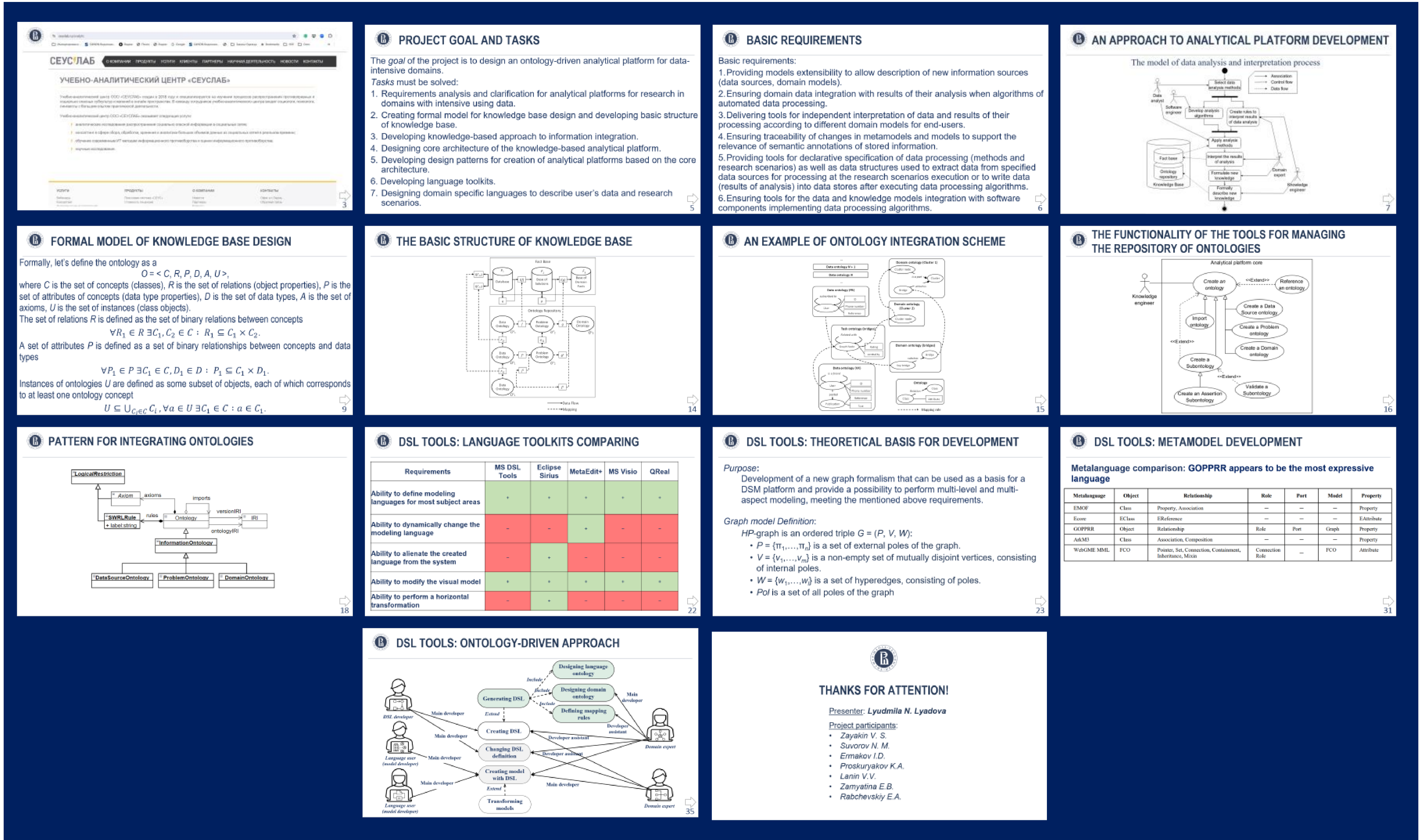
Perm, Russian Federation

AN ONTOLOGY-DRIVEN APPROACH TO THE ANALYTICAL PLATFORM DEVELOPMENT FOR DATA-INTENSIVE DOMAINS

Presenter: ***<u>Lyudmila Lyadova</u>***

ACM SIGMOD - 2024

PRESENTATION CONTENT



seuslab.ru/analytic

Импортировано... CANON Видеокам... Яндекс Почта Яндекс Google CANON Видеокам... Заказы-Одежда Bookmarks HSE Онто

СЕУСЛАБ

О КОМПАНИИ ПРОДУКТЫ УСЛУГИ КЛИЕНТЫ ПАРТНЕРЫ НАУЧНАЯ ДЕЯТЕЛЬНОСТЬ НОВОСТИ КОНТАКТЫ

УЧЕБНО-АНАЛИТИЧЕСКИЙ ЦЕНТР «СЕУСЛАБ»

Учебно-аналитический центр ООО «СЕУСЛАБ» создан в 2018 году и специализируется на изучении процессов распространения противоправных и социально опасных субкультур и явлений в онлайн пространстве. В команду сотрудников учебно-аналитического центра входят социологи, психологи, лингвисты с большим опытом практической деятельности.

Учебно-аналитический центр ООО «СЕУСЛАБ» оказывает следующие услуги:

- аналитические исследования распространения социально опасной информации в социальных сетях;
- консалтинг в сфере сбора, обработки, хранения и аналитики больших объемов данных из социальных сетей в реальном времени;
- обучение современным ИТ методам информационного противоборства и оценки информационного противоборства;
- научные исследования.

УСЛУГИ

Вебинары
Консалтинг

ПРОДУКТЫ

Поисковая система «СЕУС»
Стоимость лицензии

О КОМПАНИИ

Новости
Партнеры

КОНТАКТЫ

Офис в г.Пермь
Обратная связь

DESIGN CHALLENGES

Design challenges:

1. Integration of information (data sources models, data processing problems descriptions, domain models) at the level of source metamodels which are described independently of each other.
2. Automated interpretation of data based on formal description of logical constraints and domain rules.
3. DSL integration for specification of data processing modules for gathering, preprocessing, analyzing and interpreting data.
4. Integration of task solvers based on declarative specifications of platform modules.
5. Composition of data processing modules into pipelines based on input and output data structures descriptions.

PROJECT GOAL AND TASKS

The *goal* of the project is to design an ontology-driven analytical platform for data-intensive domains.

Tasks must be solved:

1. Requirements analysis and clarification for analytical platforms for research in domains with intensive using data.
2. Creating formal model for knowledge base design and developing basic structure of knowledge base.
3. Developing knowledge-based approach to information integration.
4. Designing core architecture of the knowledge-based analytical platform.
5. Developing design patterns for creation of analytical platforms based on the core architecture.
6. Developing language toolkits.
7. Designing domain specific languages to describe user's data and research scenarios.

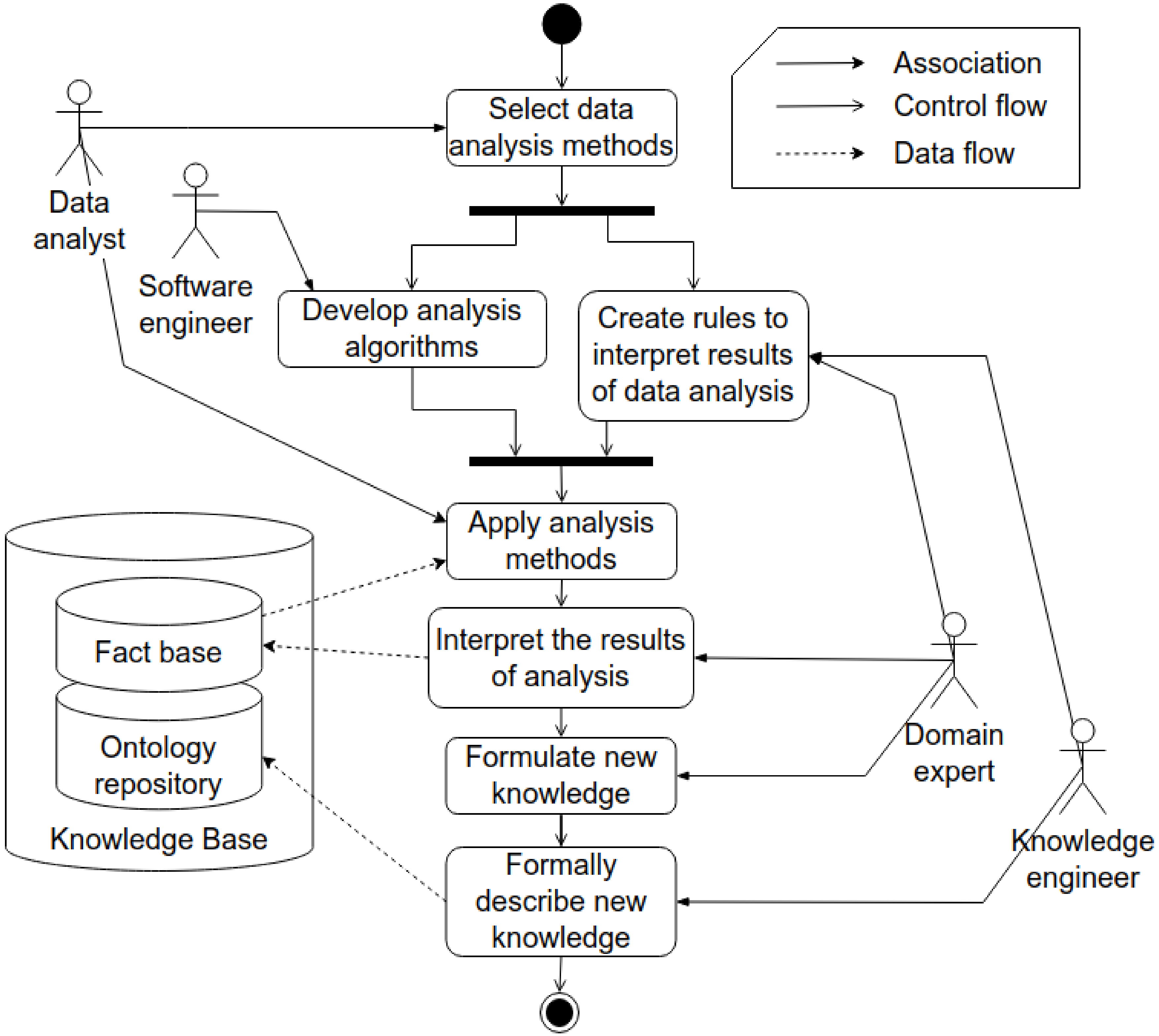
BASIC REQUIREMENTS

Basic requirements:

1. Providing models extensibility to allow description of new information sources (data sources, domain models).
2. Ensuring domain data integration with results of their analysis when algorithms of automated data processing.
3. Delivering tools for independent interpretation of data and results of their processing according to different domain models for end-users.
4. Ensuring traceability of changes in metamodels and models to support the relevance of semantic annotations of stored information.
5. Providing tools for declarative specification of data processing (methods and research scenarios) as well as data structures used to extract data from specified data sources for processing at the research scenarios execution or to write data (results of analysis) into data stores after executing data processing algorithms.
6. Ensuring tools for the data and knowledge models integration with software components implementing data processing algorithms.

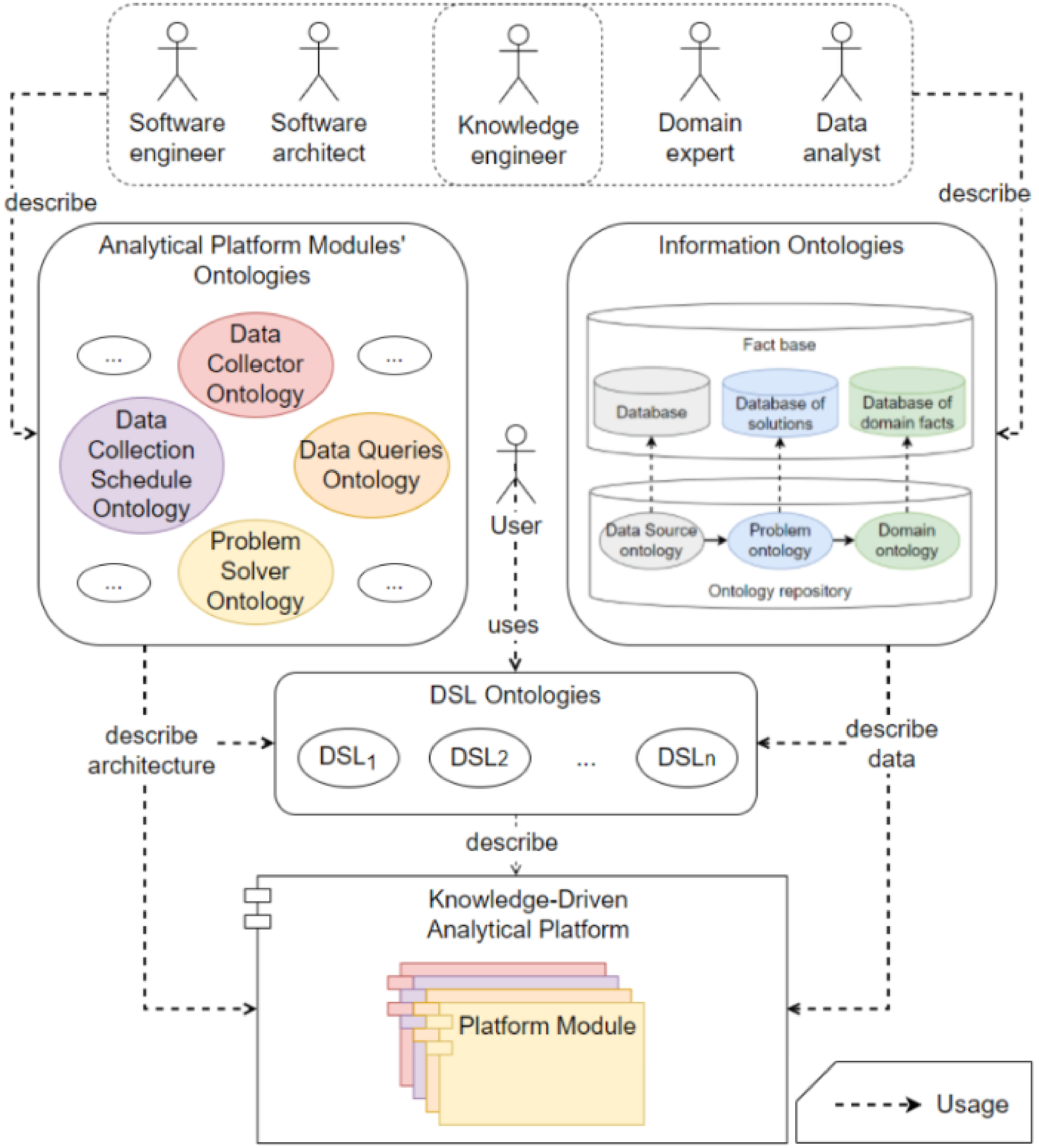
AN APPROACH TO ANALYTICAL PLATFORM DEVELOPMENT

The model of data analysis and interpretation process



AN APPROACH TO ANALYTICAL PLATFORM DEVELOPMENT

Generalized Structure Of the Analytical Platform



FORMAL MODEL OF KNOWLEDGE BASE DESIGN

Formally, let's define the ontology as a

$$O = < C, R, P, D, A, U >,$$

where *C* is the set of concepts (classes), *R* is the set of relations (object properties), *P* is the set of attributes of concepts (data type properties), *D* is the set of data types, *A* is the set of axioms, *U* is the set of instances (class objects).

The set of relations *R* is defined as the set of binary relations between concepts

$$\forall R_1 \in R\ \exists C_1, C_2 \in C :\ R_1 \subseteq C_1 \times C_2 .$$

A set of attributes *P* is defined as a set of binary relationships between concepts and data types

$$\forall P_1 \in P\ \exists C_1 \in C, D_1 \in D :\ P_1 \subseteq C_1 \times D_1 .$$

Instances of ontologies *U* are defined as some subset of objects, each of which corresponds to at least one ontology concept

$$U \subseteq \bigcup_{C_i \in C} C_i\ , \forall a \in U\ \exists C_1 \in C : a \in C_1 .$$

FORMAL MODEL OF KNOWLEDGE BASE DESIGN

Let's define the set of axioms *A* as the set of statements regarding instances of ontology of the following types:

1. Instance *a* belongs to concept C_1.
2. Instances *a* and *b* are in relation R_1.
3. Instance *a* has a P_1 property value of *d* of data type D_1.

For the simplification of next definitions let's designations for these types of axioms in accordance with the following formal definitions:

$$C_1(a) \Leftrightarrow (a \in U, C_1 \in C, a \in C_1),$$

$$R_1(a, b) \Leftrightarrow (a, b \in U, R_1 \in R, aR_1b, (\exists C_1, C_2 \in C : R_1 \subseteq C_1 \times C_2)),$$

$$P_1\big(a, D_1(d)\big) \Leftrightarrow (a \in U, d \in D_1, P_1 \in P, aP_1d, (\exists C_1 \in C, D_1 \in D : P_1 \subseteq C_1 \times D_1)).$$

FORMAL MODEL OF KNOWLEDGE BASE DESIGN

Set *A* is only a small fraction of the set of all axioms $\mathcal{A}$ that can theoretically be annotated using the ontology *O*. The fact base described by ontology *O* is denoted *F* and it is defined formally as a subset of set $\mathcal{A}$. As a result, we have

$$\mathcal{A} = \bigcup_{C_i \in C} \{C_i(a) \mid a \in U\} + \bigcup_{R_j \in R} \{R_j(a, b) \mid a, b \in U\} +$$

$$+ \bigcup_{P_k \in P, D_l \in D} \{P_k(a, D_l(d)) \mid a \in U, d \in D_l\}, A, F \subseteq \mathcal{A}, |F| \gg |A|.$$

set *A* is only a small fraction of the set of all axioms $\mathcal{A}$ that can theoretically be annotated using the ontology *O*. The fact base described by ontology *O* is denoted *F* and it is defined formally as a subset of set $\mathcal{A}$. As a result, we have

$$\mathcal{A} = \bigcup_{C_i \in C} \{C_i(a) \mid a \in U\} + \bigcup_{R_j \in R} \{R_j(a, b) \mid a, b \in U\} +$$

$$+ \bigcup_{P_k \in P, D_l \in D} \{P_k(a, D_l(d)) \mid a \in U, d \in D_l\}, A, F \subseteq \mathcal{A}, |F| \gg |A|.$$

FORMAL MODEL OF KNOWLEDGE BASE DESIGN

Set A is only a small fraction of the set of all axioms $\mathcal{A}$ that can theoretically be annotated using the ontology O. The fact base described by ontology O is denoted F and it is defined formally as a subset of set $\mathcal{A}$. As a result, we have

$$\mathcal{A} = \bigcup_{C_i \in C} \{C_i(a) \,|a \in U\} + \bigcup_{R_j \in R} \{R_j(a,b) \,|a,b \in U\} +$$

$$+ \bigcup_{P_k \in P, D_l \in D} \{P_k(a, D_l(d)) \,|a \in U, d \in D_l\}, A, F \subseteq \mathcal{A}, |F| \gg |A|.$$

Finally, the knowledge base K can be formally defined as a $K = < O, F >$. let O_1, O_2, O_3 be ontologies, $f : O_1 \rightsquigarrow O_2$ and $g : O_2 \rightsquigarrow O_3$ be ontological mappings.

FORMAL MODEL OF KNOWLEDGE BASE DESIGN

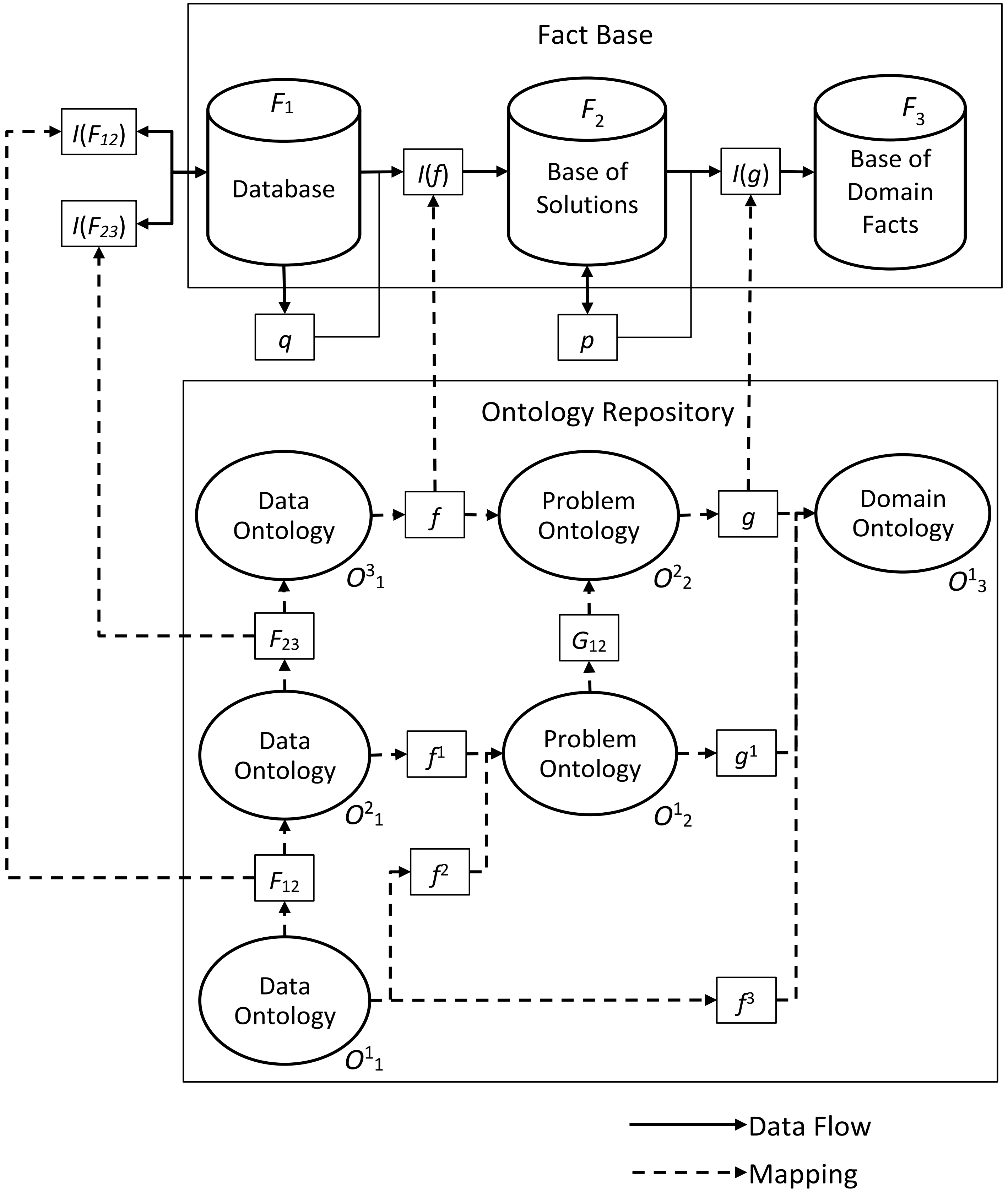
Then the definition of the process of replenishing the knowledge base with new facts when applying the procedures for processing and interpreting subject data can be formally presented as a composition of relationships

$$[I(g)] \circ p \circ [I(f)] \circ q \subseteq 2^{Q} \times 2^{\mathcal{A}(O_3)}, q \subseteq 2^{Q} \times 2^{\mathcal{A}(O_1)}, p \subseteq 2^{\mathcal{A}(O_2)} \times 2^{\mathcal{A}(O_2)}.$$

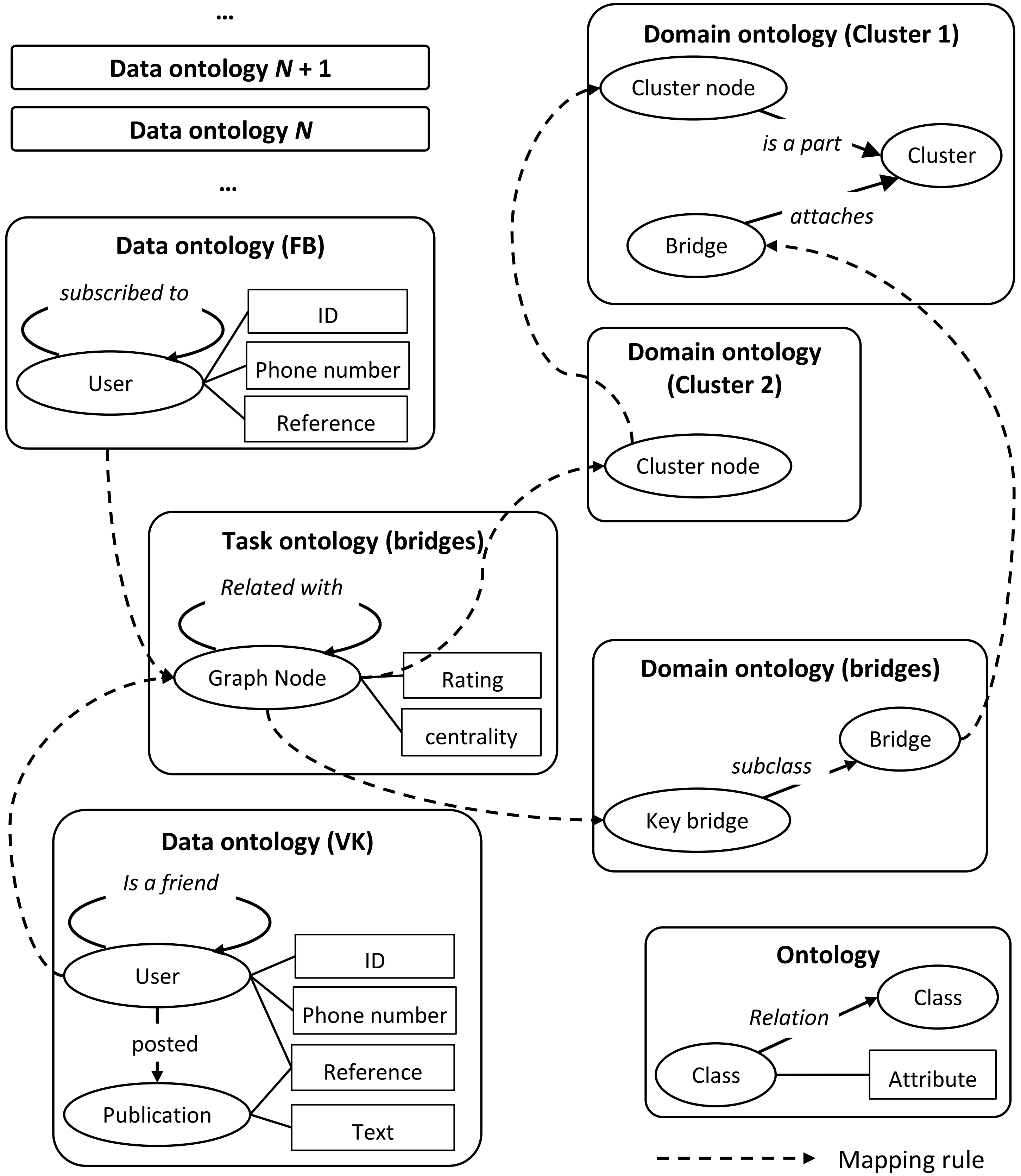
Here, *q* is a procedure for generating input data with structure described using the ontology O_1. Next, the interpretive mapping *I*(*f*) is used to structure this data according to the ontology O_2, which declaratively describes the task of processing this data using the procedure *p*. Finally, the interpretive mapping *I*(*g*) is an interpretation of the results of applying the data processing procedure for structuring according to the O_3 ontology.

Between different versions of ontologies using production rules, one can define the corresponding ontological mappings F_{ij}: $O_i \rightsquigarrow O_j$, where *i*, *j* are versions of ontologies.

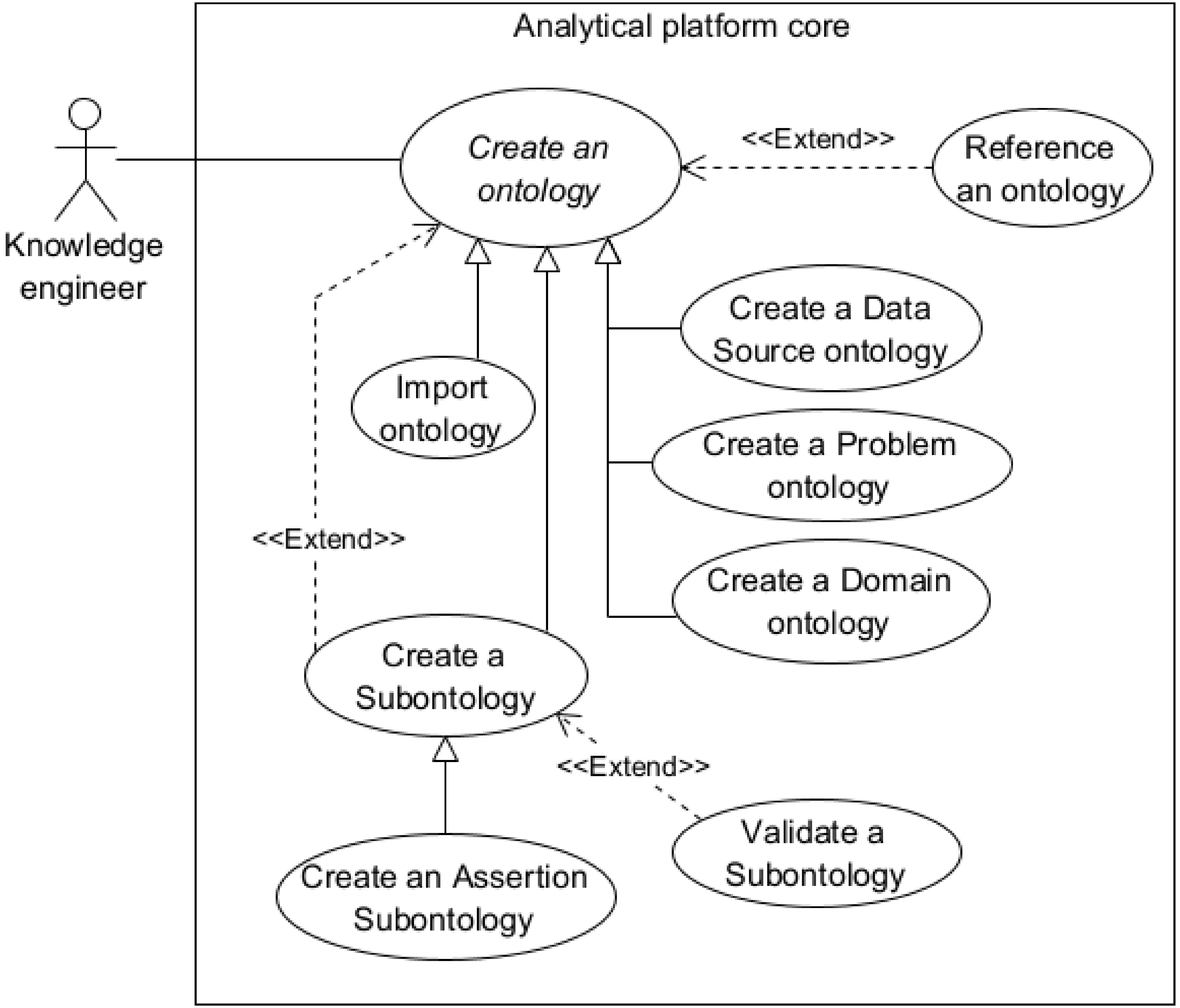
THE BASIC STRUCTURE OF KNOWLEDGE BASE



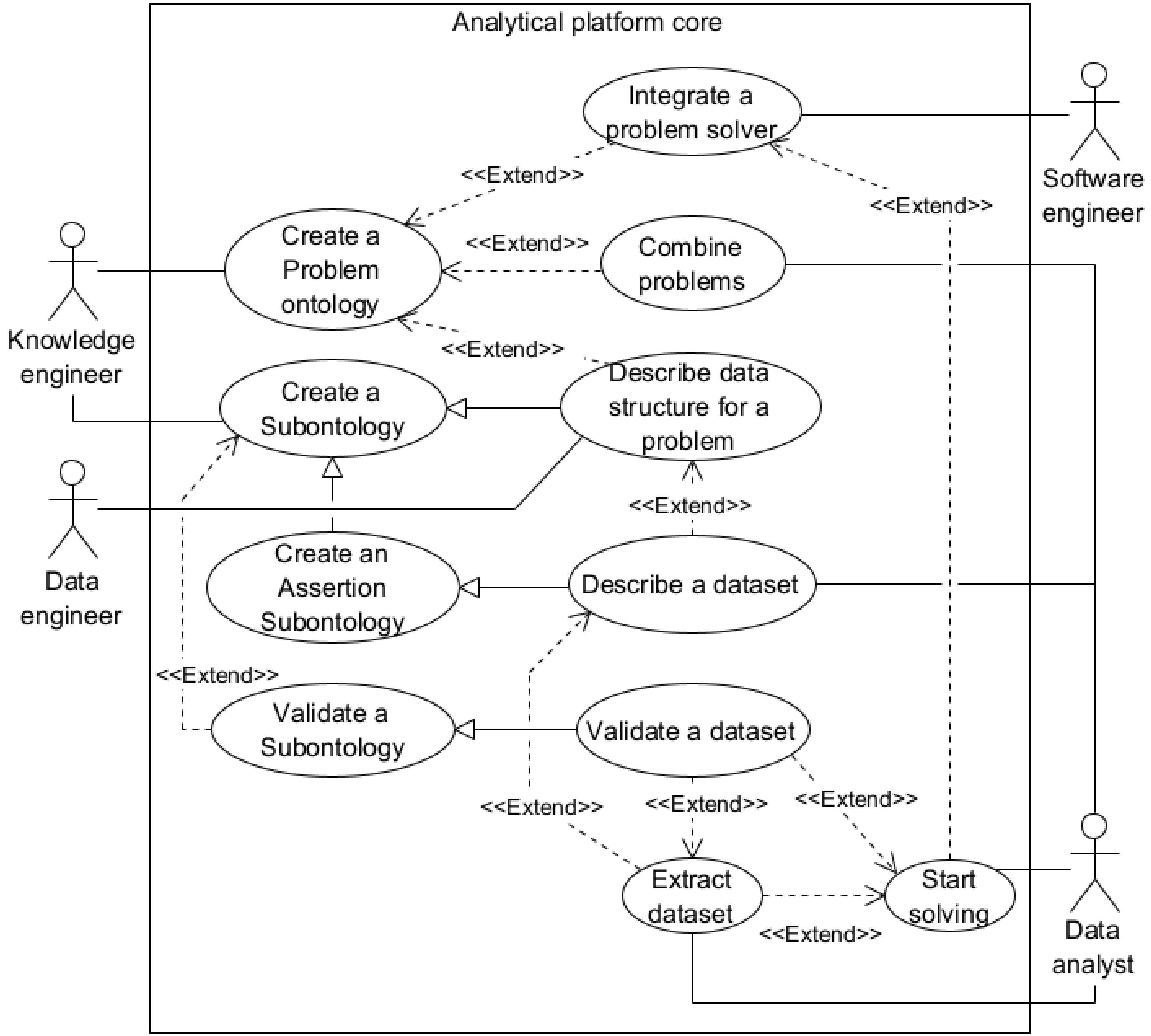
AN EXAMPLE OF ONTOLOGY INTEGRATION SCHEME



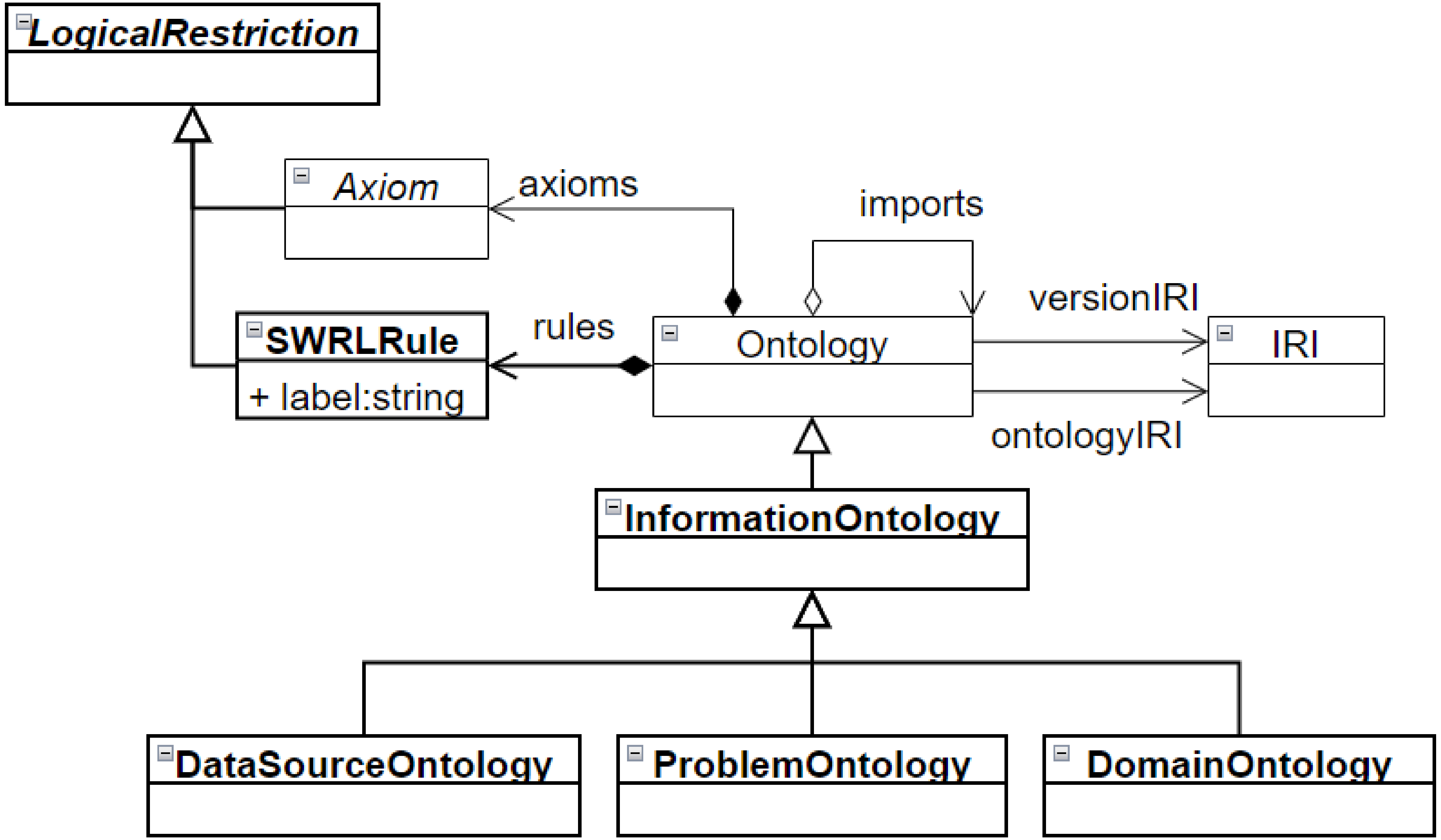
THE FUNCTIONALITY OF THE TOOLS FOR MANAGING THE REPOSITORY OF ONTOLOGIES



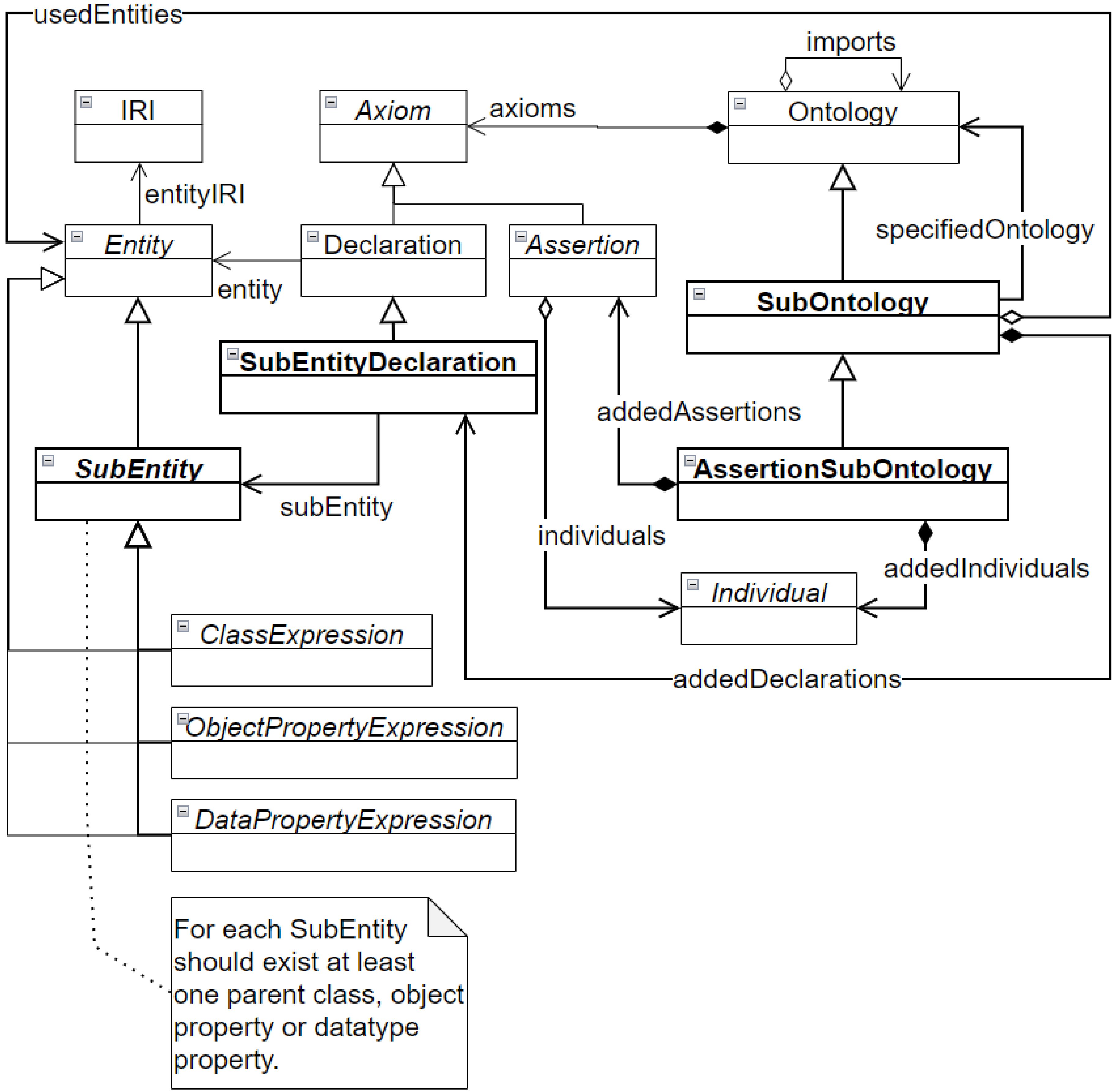
THE FUNCTIONALITY OF DSL-TOOLS FOR THE DATA ENGINEER



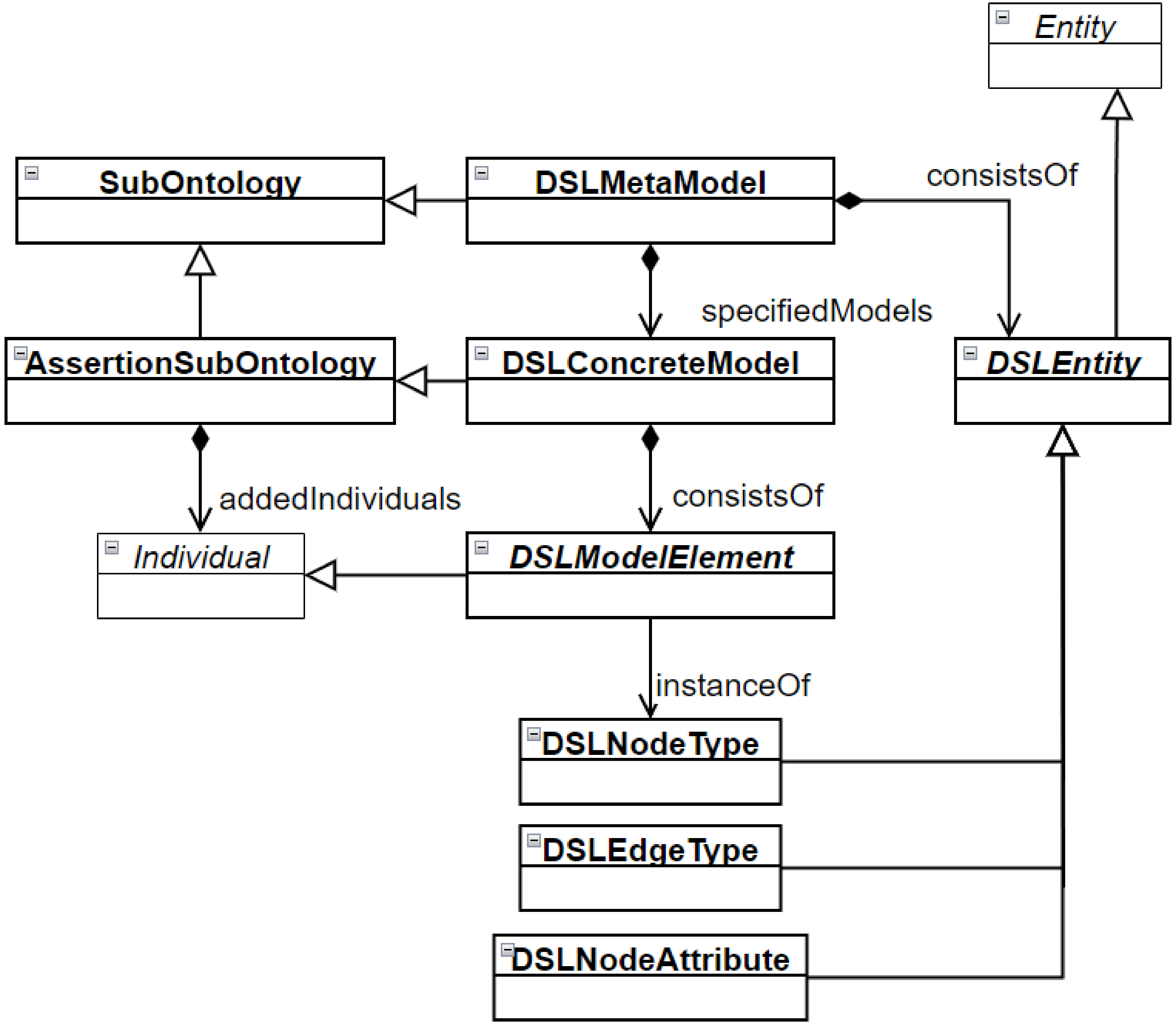
PATTERN FOR INTEGRATING ONTOLOGIES



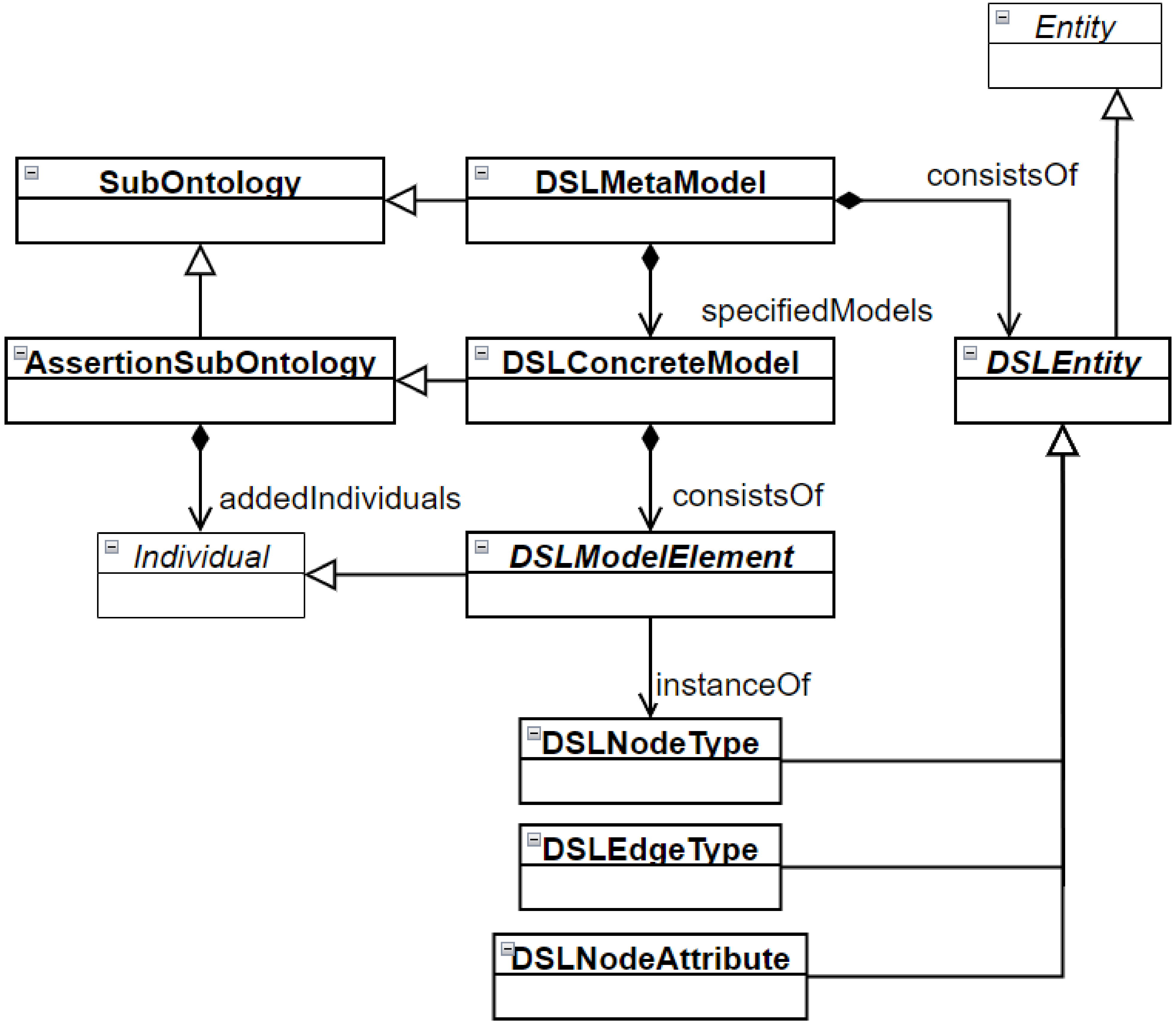
PATTERN FOR ONTOLOGY-BASED METAMODELING



PATTERN FOR INTEGRATING DSL



PATTERN FOR INTEGRATING PROBLEM SOLVERS



DSL TOOLS: LANGUAGE TOOLKITS COMPARING

Requirements	MS DSL Tools	Eclipse Sirius	MetaEdit+	MS Visio	QReal
Ability to define modeling languages for most subject areas	+	+	+	+	+
Ability to dynamically change the modeling language	–	–	+	–	–
Ability to alienate the created language from the system	–	+	–	–	–
Ability to modify the visual model	+	+	+	+	+
Ability to perform a horizontal transformation	–	+	–	–	–

DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

Purpose:

Development of a new graph formalism that can be used as a basis for a DSM platform and provide a possibility to perform multi-level and multi-aspect modeling, meeting the mentioned above requirements.

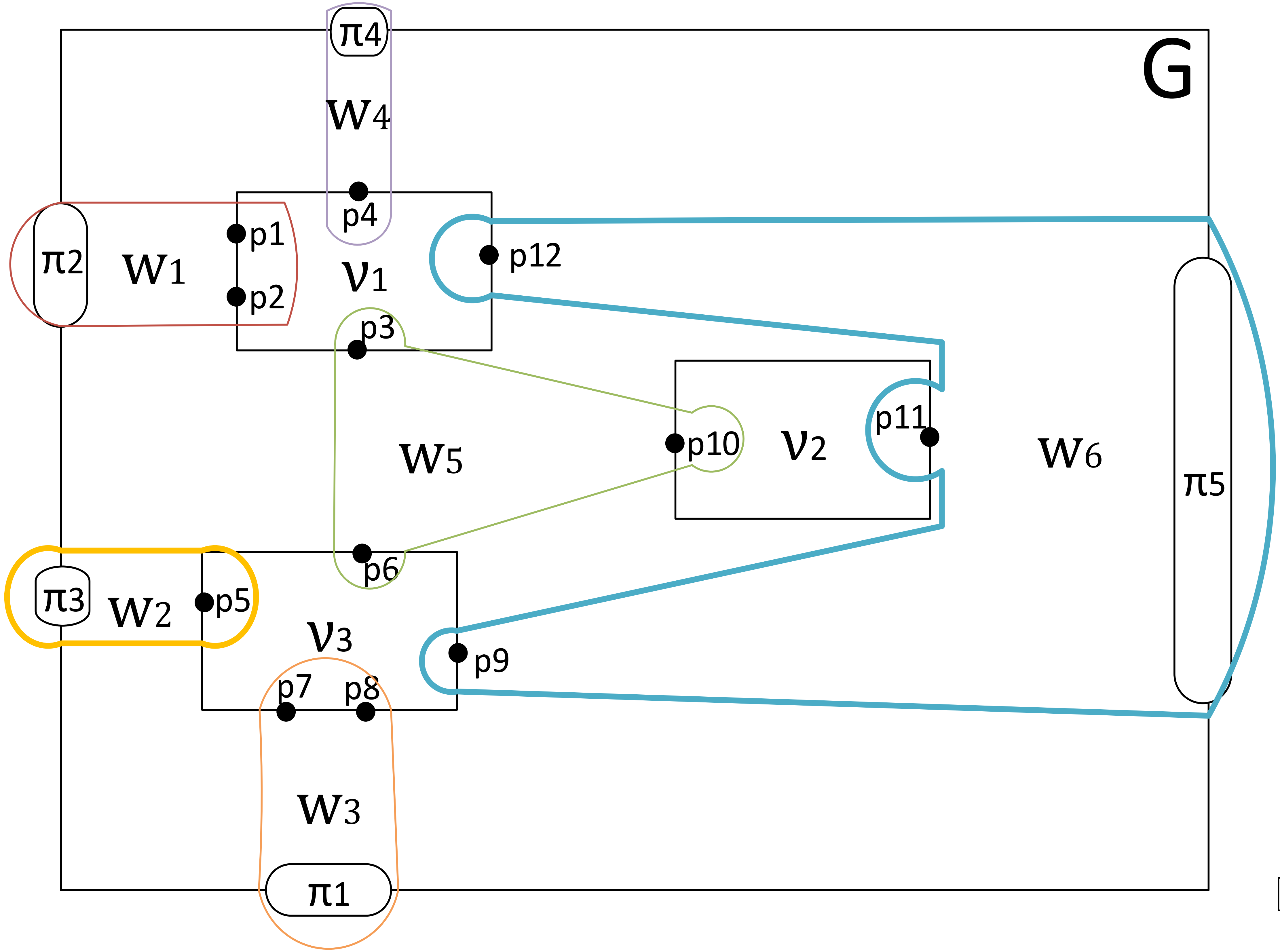
Graph model Definition:

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

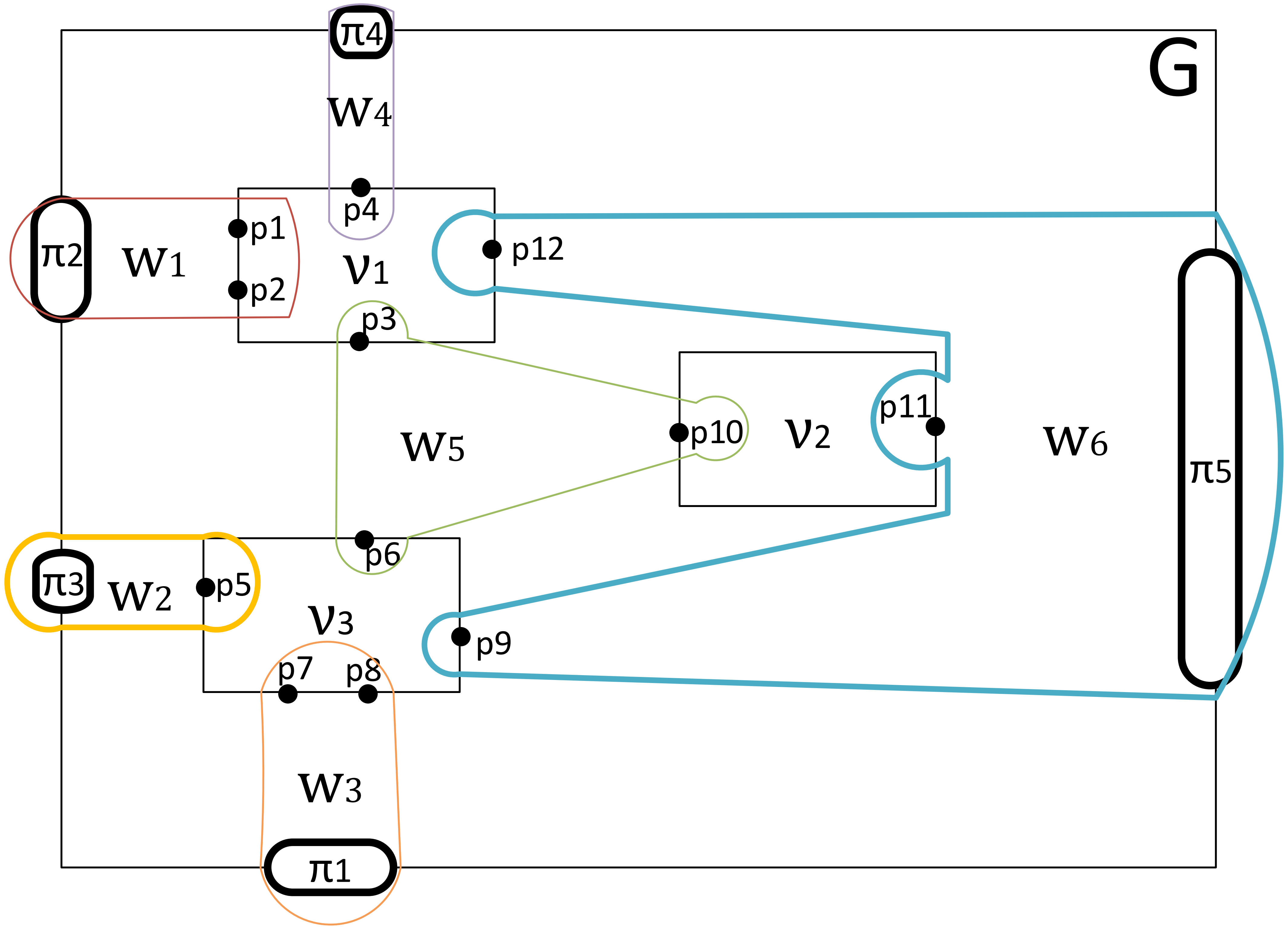
DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

Pol is a set of all poles of the graph



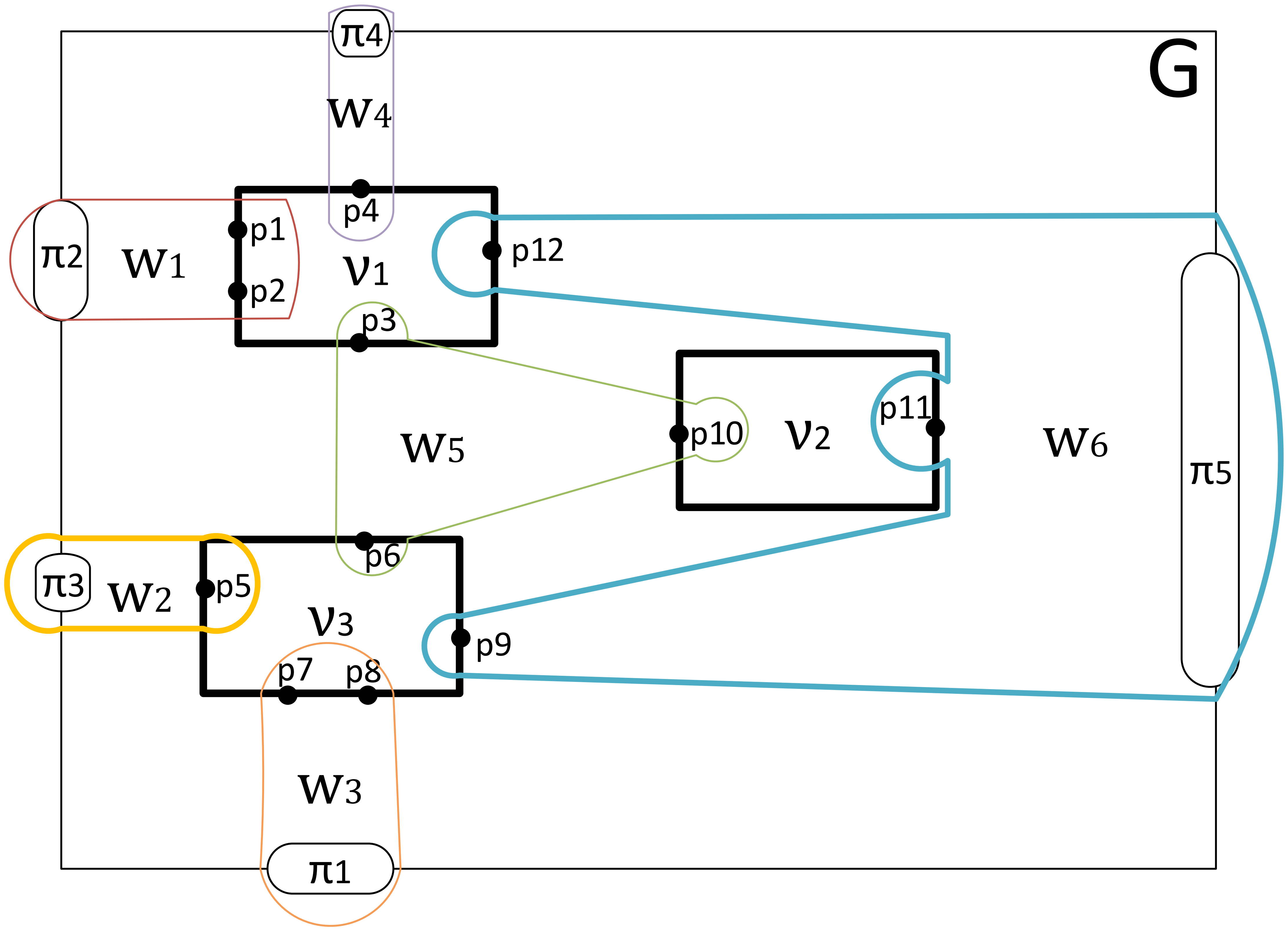
DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

$P = \{\pi_1, \ldots, \pi_n\}$ is a set of external poles of the graph



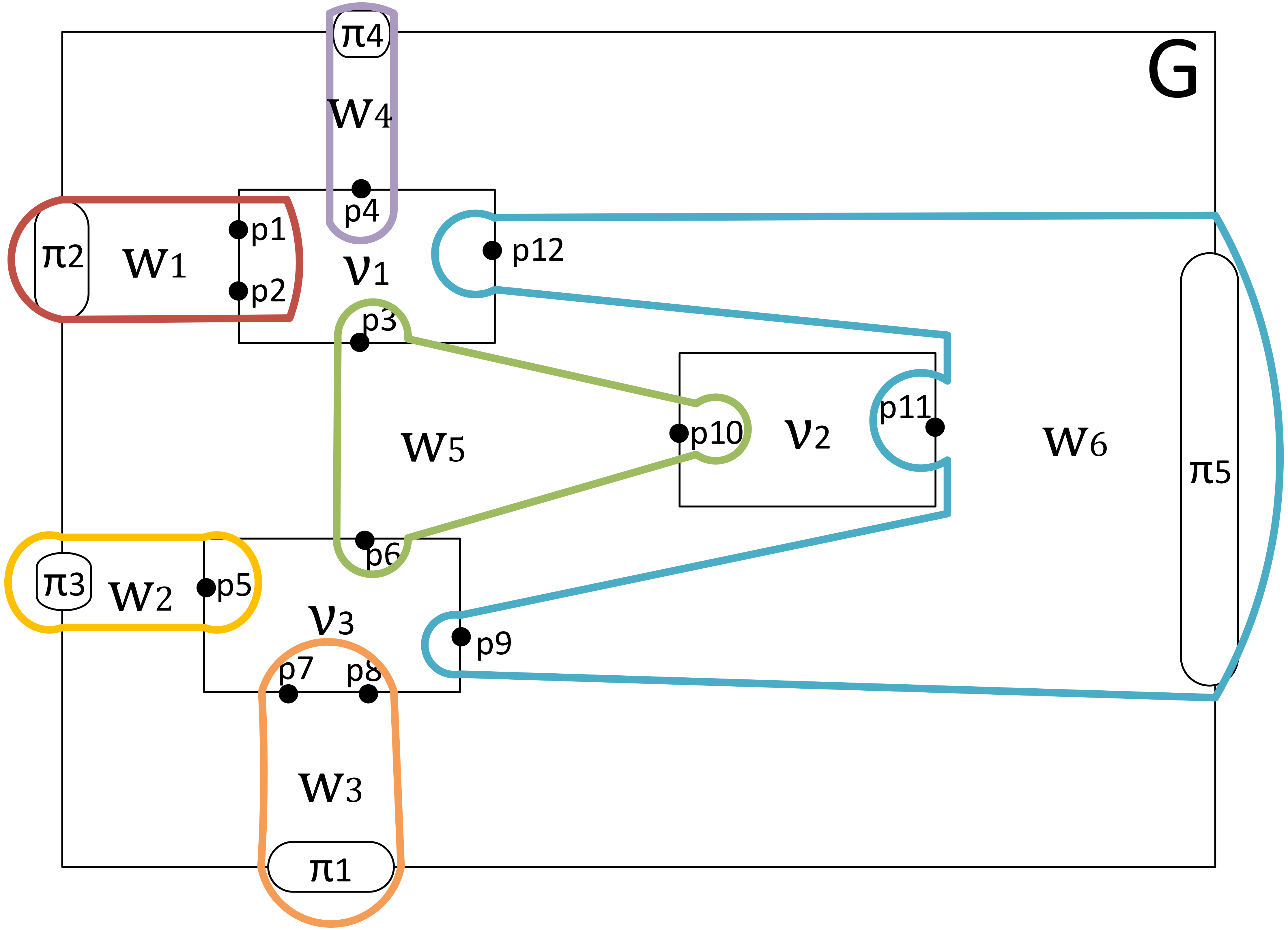
DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

$V = \{v_1,\ldots,v_m\}$ is a non-empty set of mutually disjoint vertices, consisting of internal poles

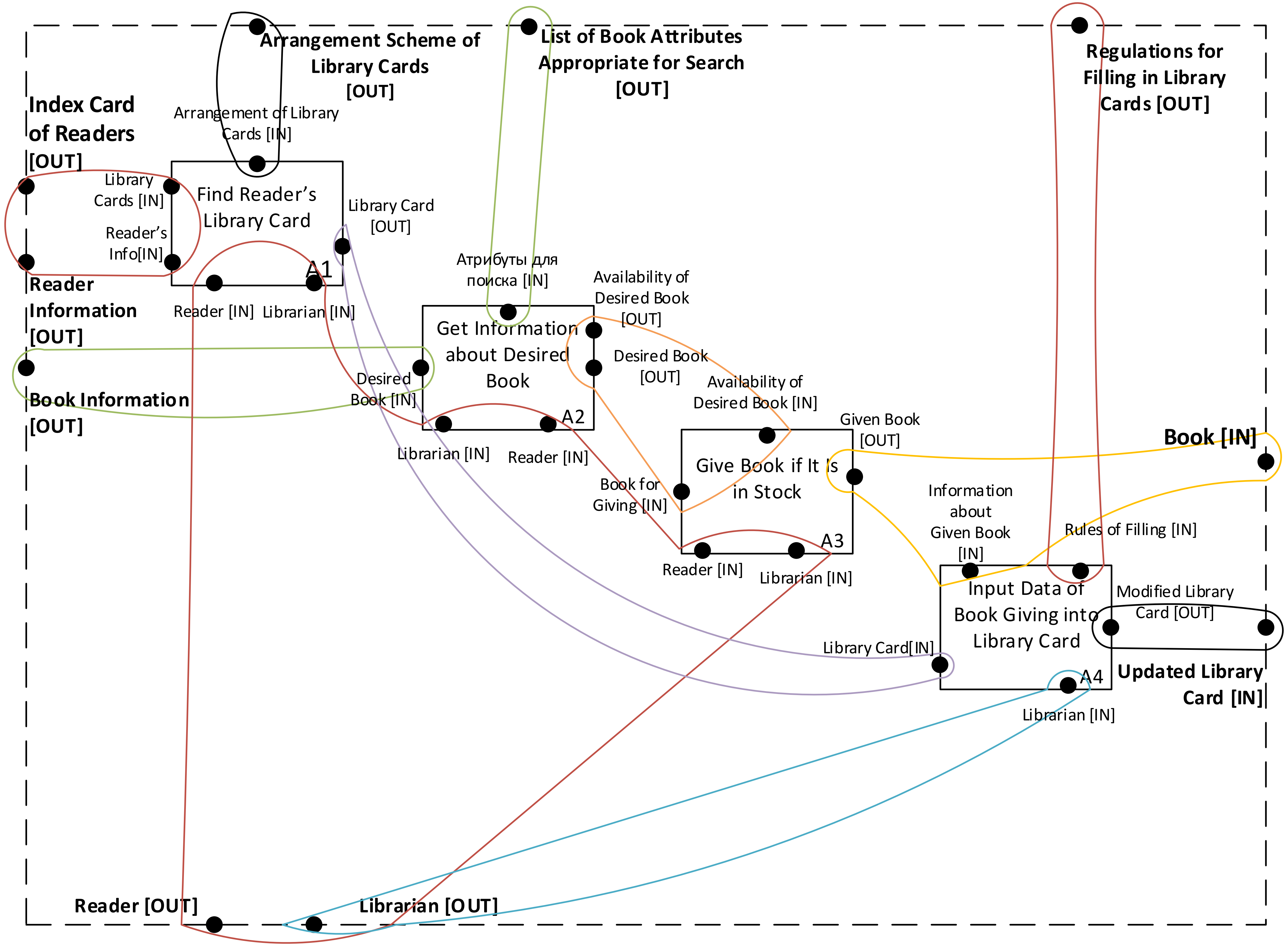


DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

***W* = {*w*$_1$,…,*w*$_l$}** is a set of hyperedges, consisting of poles



DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT



DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

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

DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

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 $\forall w' \in W'$: $[\lvert w' \rvert = 2]$

DSL TOOLS: METAMODEL DEVELOPMENT

Metalanguage comparison: GOPPRR appears to be the most expressive language

Metalanguage	Object	Relationship	Role	Port	Model	Property
EMOF	Class	Property, Association	–	–	–	Property
Ecore	EClass	EReference	–	–	–	EAttribute
GOPPRR	Object	Relationship	Role	Port	Graph	Property
ArkM3	Class	Association, Composition	–	–	–	Property
WebGME MML	FCO	Pointer, Set, Connection, Containment, Inheritance, Mixin	Connection Role	–	FCO	Attribute

DSL TOOLS: METAMODEL DEVELOPMENT

Metalanguage comparison: **GOPPRR appears to be the most expressive language**

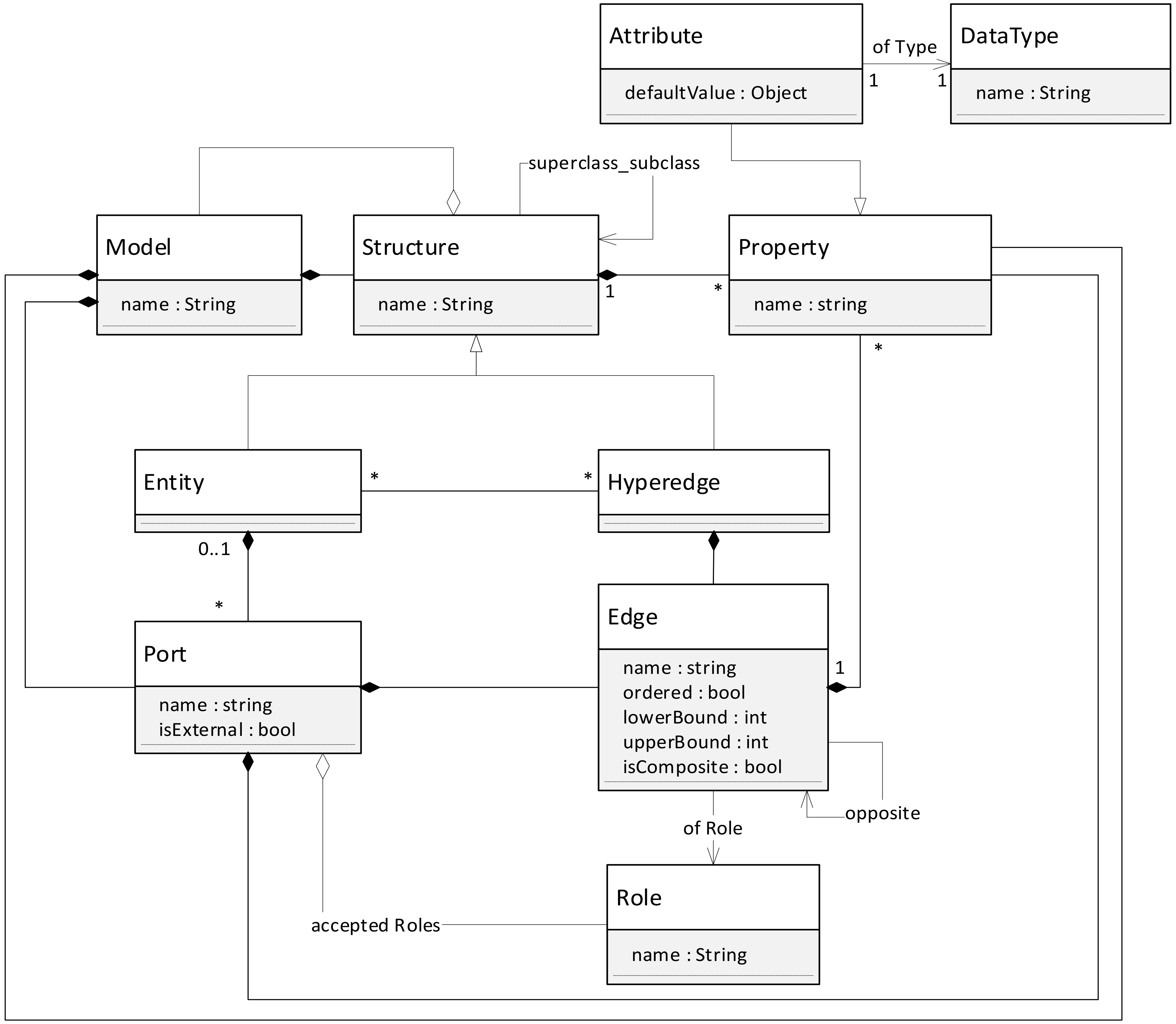
Notation	EMOF	Ecore	MS DSL Tools MML	GOPPRR	ArkM3	WebGME MML
SADT	67%	67%	67%	100%	67%	83%
DFD	80%	80%	80%	100%	80%	100%
ERD	100%	100%	100%	100%	100%	100%
Use Case Diagram	100%	100%	100%	100%	100%	80%
Activity Diagram	80%	80%	80%	80%	80%	80%
Sequence Diagram	80%	80%	80%	90%	80%	80%
Class Diagram	100%	100%	100%	100%	100%	86%
Component Diagram	75%	75%	75%	100%	75%	75%
Average:	**85%**	**85%**	**85%**	**96%**	**85%**	**86%**

DSL TOOLS: METAMODEL DEVELOPMENT

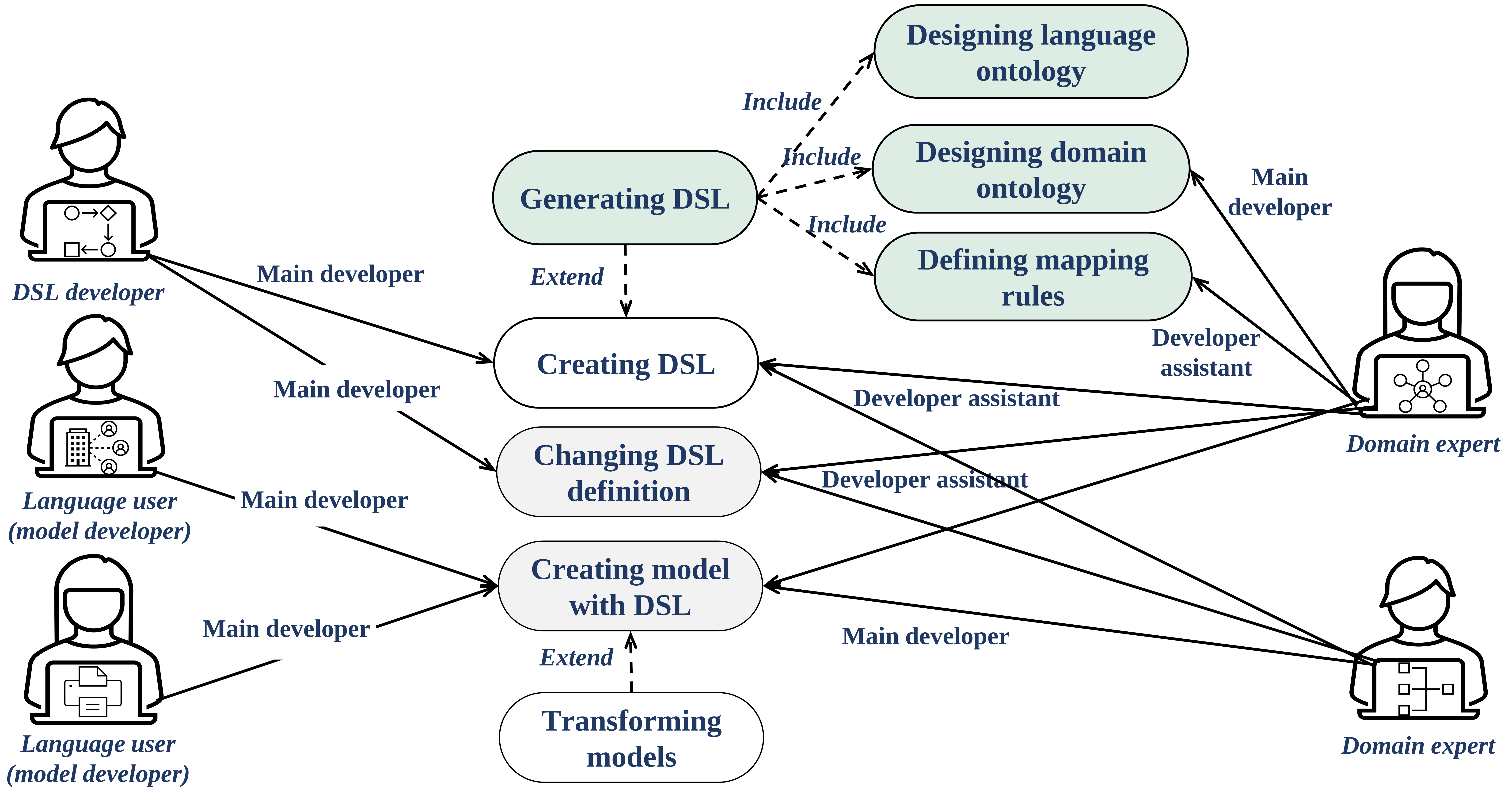
Metalanguage comparison: GOPPRR appears to be the most expressive language

Concept	GOPPRR	System Metalanguage
Model	Graph	Model
Object	Object	Entity
Port	Port	Port
Property	Property	Property
Role	Role	Role
Relationship	Relationship	Edge
Hyperedge	—	Hyperedge

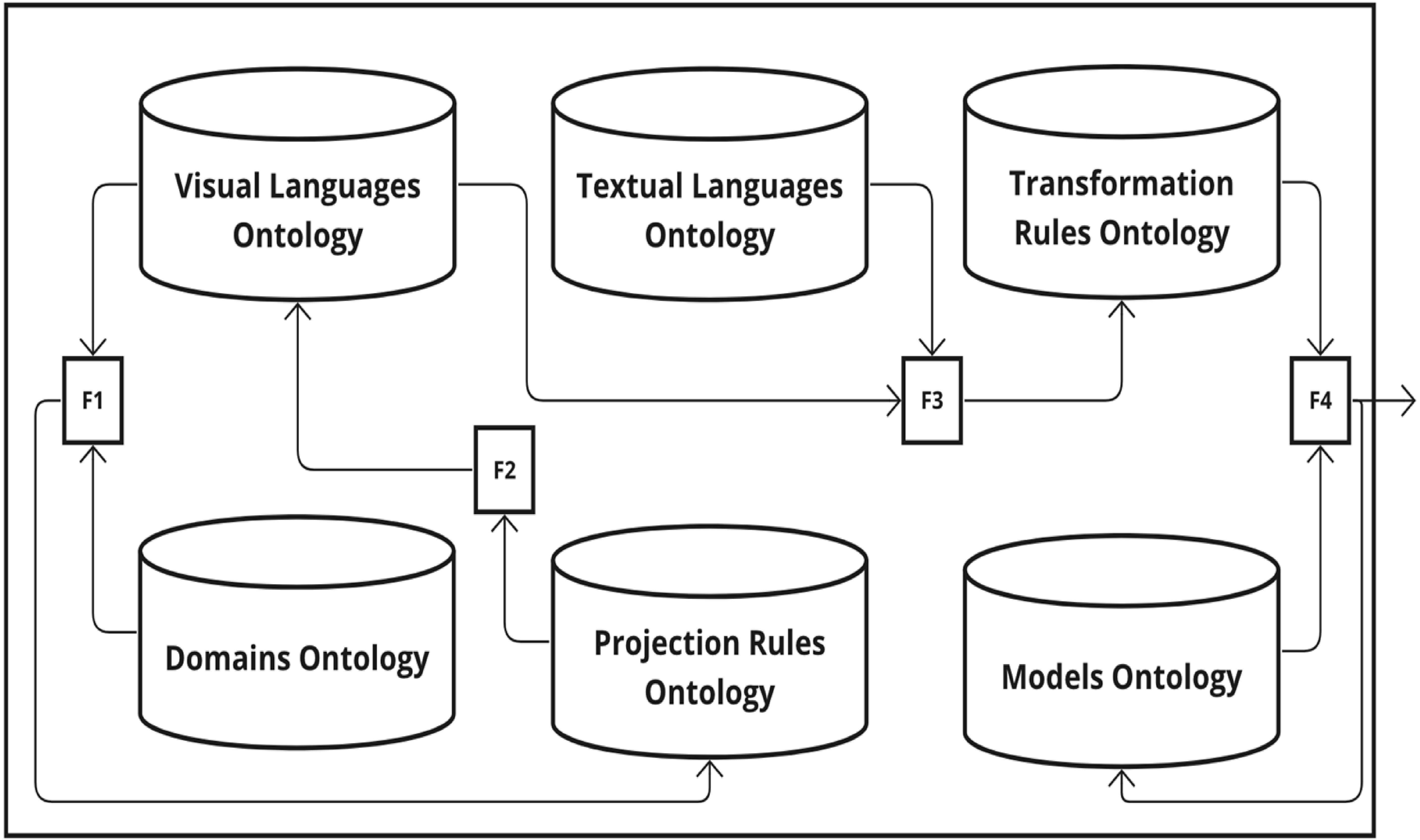
DSL TOOLS: METAMODEL DEVELOPMENT



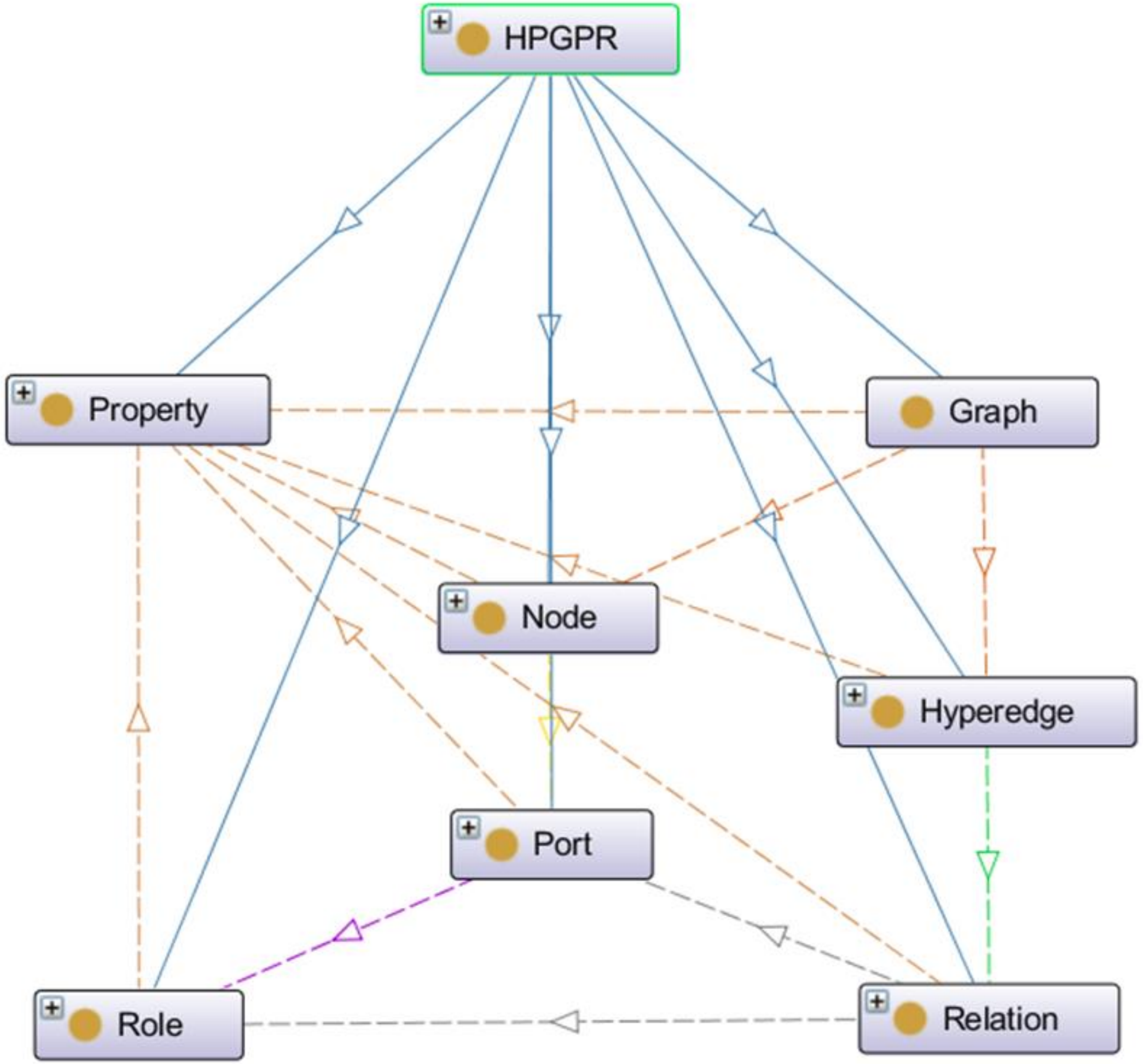
DSL TOOLS: ONTOLOGY-DRIVEN APPROACH



DSL TOOLS: ONTOLOGY OF LANGUAGES



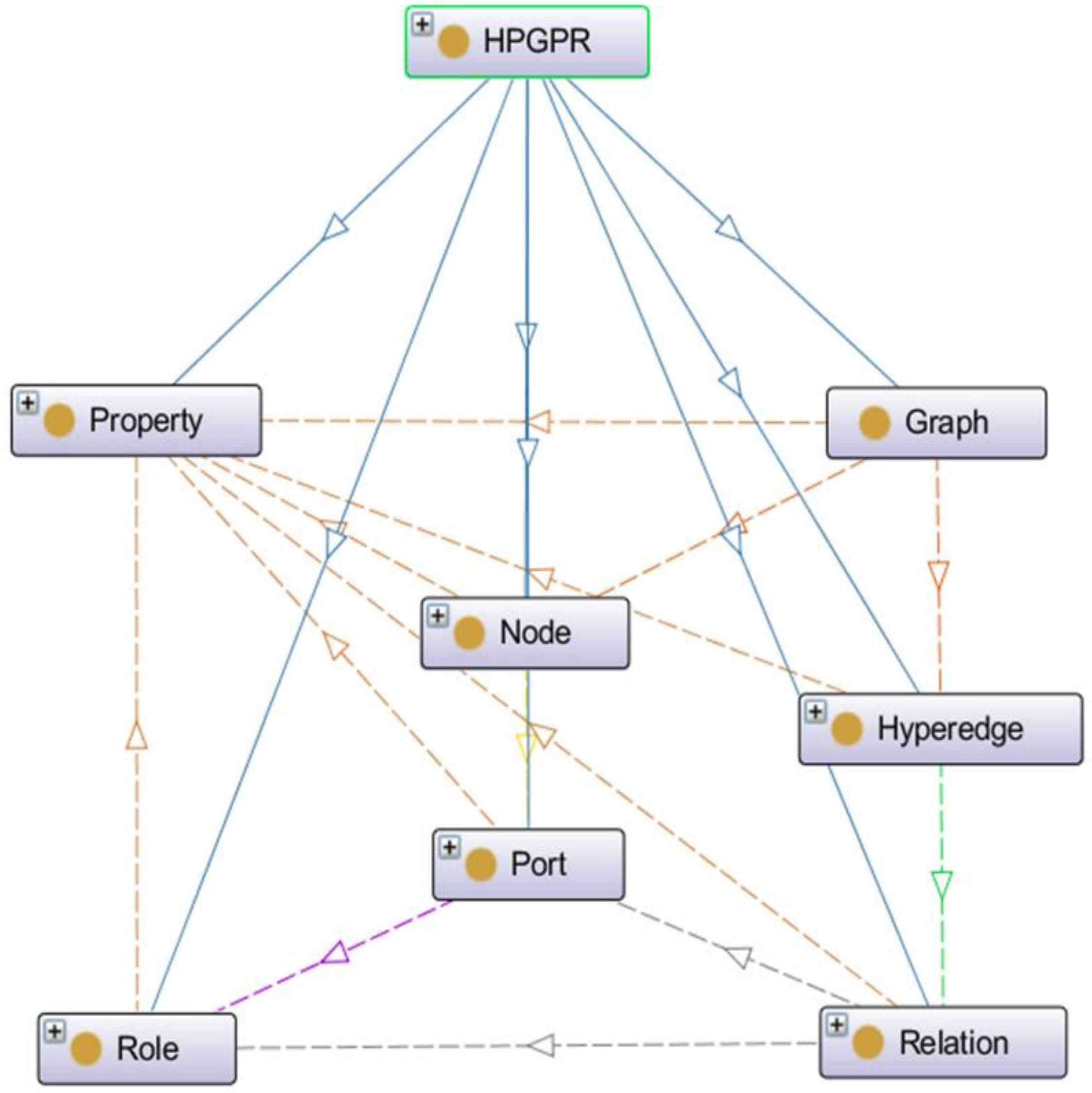
DSL TOOLS: ONTOLOGY OF LANGUAGES – HPGPR METALANGUAGE



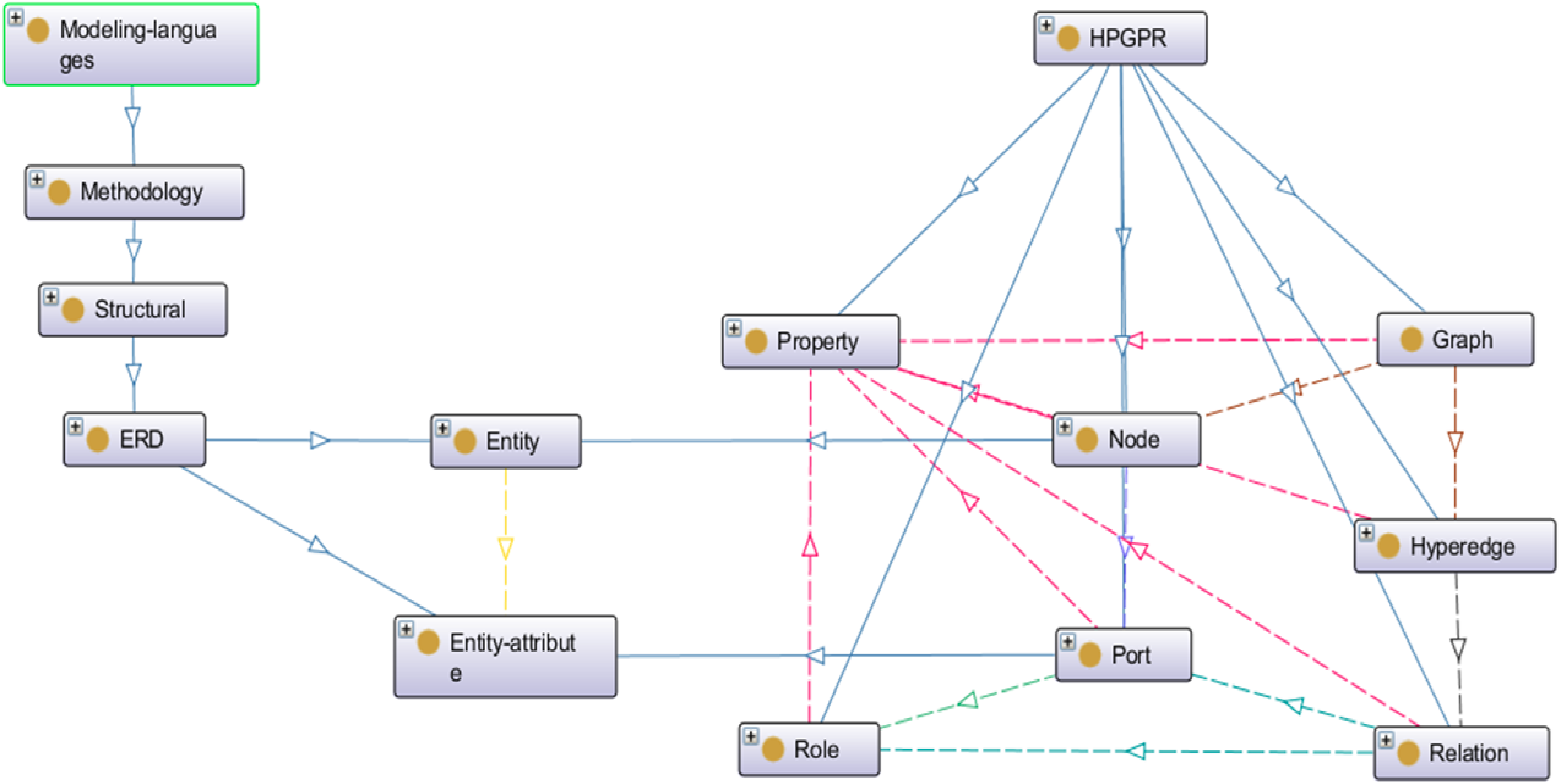
DSL TOOLS: ONTOLOGY OF LANGUAGES – HPGPR METALANGUAGE

Seven types of connections are used in metamodels:

1. The relationship between the graph and the elements that it includes (Graph–Bindings).
2. The relationships between hyperedges and ordinary edges (Hyperedge–Relation–Bindings).
3. The relationship between the node and the port (Node–Port–Bindings).
4. The relationship between all objects of the language and the properties (Objects–Properties–Bindings).
5. The relationship between the port and the roles (Port–Roles–Bindings).
6. The relationship between a property and its data type (Property–Data Type–Bindings).
7. Relationship between communication, port, and role (Relationship–Bindings).



DSL TOOLS: METAMODEL REPRESENTATION IN THE ONTOLOGY OF LANGUAGES



DSL TOOLS: LANGUAGE TOOLKITS IMPLEMENTATION

Based on these ontologies, the functions of the language toolkit are implemented:

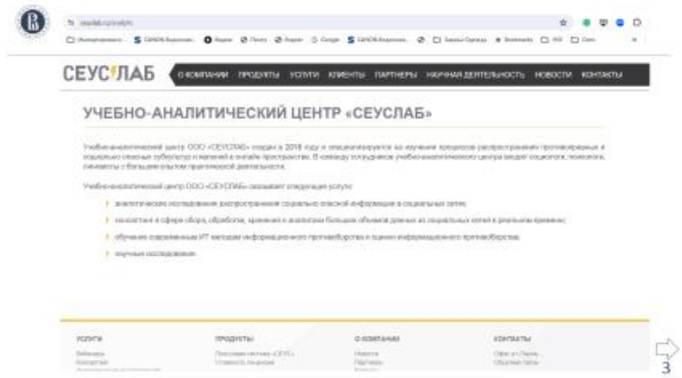
- F1 defines DSL generation rules as a mapping of the domain ontology onto the metamodel of the basic modeling language.
- F2 applies DSL generation rules to create a new DSL.
- F3 matches the elements of the visual language metamodels with the elements of textual language grammars when creating transformation rules (code generation rules).
- F4 applies transformation rules to generate textual artifacts.

THANKS FOR ATTENTION!

Presenter: ***Lyudmila N. Lyadova***

Project participants:

- *Zayakin V. S.*
- *Suvorov N. M.*
- *Ermakov I.D.*
- *Proskuryakov K.A.*
- *Lanin V.V.*
- *Zamyatina E.B.*
- *Rabchevskiy E.A.*



PROJECT GOAL AND TASKS

The *goal* of the project is to design an ontology-driven analytical platform for data-intensive domains.

Tasks must be solved:

1. Requirements analysis and clarification for analytical platforms for research in domains with intensive using data.
2. Creating formal model for knowledge base design and developing basic structure of knowledge base.
3. Developing knowledge-based approach to information integration.
4. Designing core architecture of the knowledge-based analytical platform.
5. Developing design patterns for creation of analytical platforms based on the core architecture.
6. Developing language toolkits.
7. Designing domain specific languages to describe user's data and research scenarios.

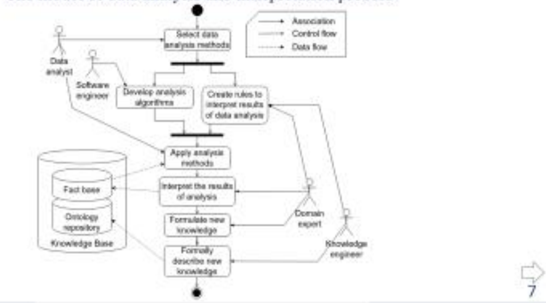
BASIC REQUIREMENTS

Basic requirements:

1. Providing models extensibility to allow description of new information sources (data sources, domain models).
2. Ensuring domain data integration with results of their analysis when algorithms of automated data processing.
3. Delivering tools for independent interpretation of data and results of their processing according to different domain models for end-users.
4. Ensuring traceability of changes in metamodels and models to support the relevance of semantic annotations of stored information.
5. Providing tools for declarative specification of data processing (methods and research scenarios) as well as data structures used to extract data from specified data sources for processing at the research scenarios execution or to write data (results of analysis) into data stores after executing data processing algorithms.
6. Ensuring tools for the data and knowledge models integration with software components implementing data processing algorithms.

AN APPROACH TO ANALYTICAL PLATFORM DEVELOPMENT

The model of data analysis and interpretation process



FORMAL MODEL OF KNOWLEDGE BASE DESIGN

Formally, let's define the ontology as a

$$O = < C, R, P, D, A, U >,$$

where C is the set of concepts (classes), R is the set of relations (object properties), P is the set of attributes of concepts (data type properties), D is the set of data types, A is the set of axioms, U is the set of instances (class objects).

The set of relations R is defined as the set of binary relations between concepts

$$\forall R_1 \in R\ \exists C_1, C_2 \in C:\ R_1 \subseteq C_1 \times C_2.$$

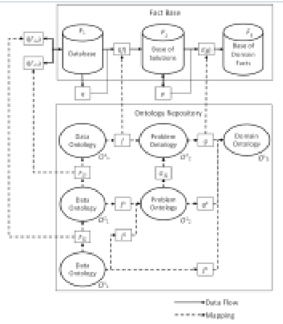
A set of attributes P is defined as a set of binary relationships between concepts and data types

$$\forall P_1 \in P\ \exists C_1 \in C, D_1 \in D:\ P_1 \subseteq C_1 \times D_1.$$

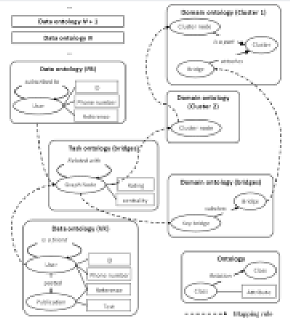
Instances of ontologies U are defined as some subset of objects, each of which corresponds to at least one ontology concept

$$U \subseteq \cup_{C_i \in C} C_i, \forall a \in U\ \exists C_1 \in C : a \in C_1.$$

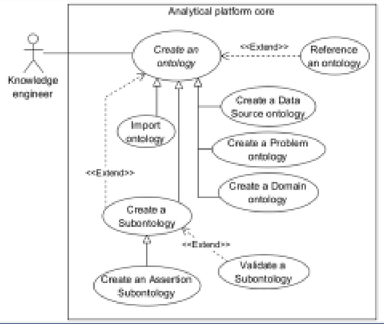
THE BASIC STRUCTURE OF KNOWLEDGE BASE



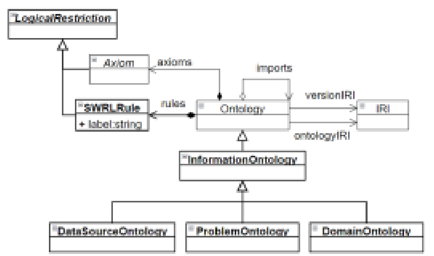
AN EXAMPLE OF ONTOLOGY INTEGRATION SCHEME



THE FUNCTIONALITY OF THE TOOLS FOR MANAGING THE REPOSITORY OF ONTOLOGIES



PATTERN FOR INTEGRATING ONTOLOGIES



DSL TOOLS: LANGUAGE TOOLKITS COMPARING

Requirements	MS DSL Tools	Eclipse Sirius	MetaEdit+	MS Visio	QReal
Ability to define modeling languages for most subject areas	+	+	+	+	+
Ability to dynamically change the modeling language	–	–	+	–	–
Ability to alienate the created language from the system	–	+	–	–	–
Ability to modify the visual model	+	+	+	+	+
Ability to perform a horizontal transformation	–	+	–	–	–

DSL TOOLS: THEORETICAL BASIS FOR DEVELOPMENT

Purpose:

Development of a new graph formalism that can be used as a basis for a DSM platform and provide a possibility to perform multi-level and multi-aspect modeling, meeting the mentioned above requirements.

Graph model Definition:

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

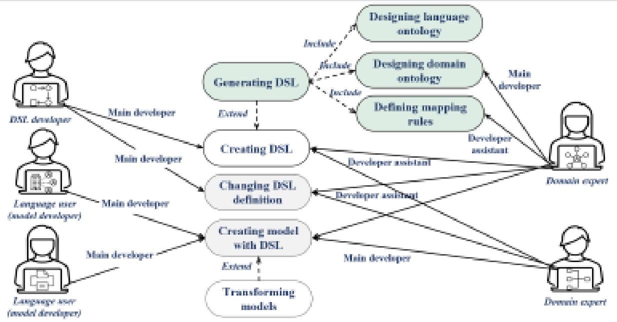
- $P = \{\pi_1,\ldots,\pi_n\}$ is a set of external poles of the graph.
- $V = \{v_1,\ldots,v_m\}$ is a non-empty set of mutually disjoint vertices, consisting of internal poles.
- $W = \{w_1,\ldots,w_l\}$ is a set of hyperedges, consisting of poles.
- *Pol* is a set of all poles of the graph

DSL TOOLS: METAMODEL DEVELOPMENT

Metalanguage comparison: GOPPRR appears to be the most expressive language

Metalanguage	Object	Relationship	Role	Port	Model	Property
EMOF	Class	Property, Association	–	–	–	Property
Ecore	EClass	EReference	–	–	–	EAttribute
GOPPRR	Object	Relationship	Role	Port	Graph	Property
ArkM3	Class	Association, Composition	–	–	–	Property
WebGME MML	FCO	Pointer, Set, Connection, Containment, Inheritance, Mixin	Connection Role	–	FCO	Attribute

DSL TOOLS: ONTOLOGY-DRIVEN APPROACH



THANKS FOR ATTENTION!

Presenter: ***Lyudmila N. Lyadova***

Project participants:

- *Zayakin V. S.*
- *Suvorov N. M.*
- *Ermakov I.D.*
- *Proskuryakov K.A.*
- *Lanin V.V.*
- *Zamyatina E.B.*
- *Rabchevskiy E.A.*