

## Research Article

# A Configurable Semantic-Based Transformation Method towards Conceptual Models

Tiexin Wang <sup>1,2</sup>, Jingwen Cao,<sup>1</sup> Chuanqi Tao,<sup>1</sup> Zhibin Yang,<sup>1,2</sup> Yi Wu,<sup>1</sup> and Bohan Li<sup>1</sup>

<sup>1</sup>College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

<sup>2</sup>Key Laboratory of Safety-Critical Software (Nanjing University of Aeronautics and Astronautics), Ministry of Industry and Information Technology, Nanjing 211106, China

Correspondence should be addressed to Tiexin Wang; [tiexin.wang@nuaa.edu.cn](mailto:tiexin.wang@nuaa.edu.cn)

Received 12 May 2020; Accepted 16 June 2020; Published 27 September 2020

Guest Editor: Jianxin Li

Copyright © 2020 Tiexin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Conceptual models are built to depict and analyze complex systems. They are made of concepts and relationships among these concepts. In a particular domain, conceptual models are helpful for different stakeholders to reach a clear and unified view of domain problems. However, the process of building conceptual models is time-consuming, tedious, and expertise required. To improve the efficiency of the building process, this paper proposes a configurable semantic-based (semi-) automatic conceptual model transformation methodology (SbACMT) that tries to reuse existing conceptual models to generate new models. SbACMT contains three parts: (i) a configurable semantic relatedness computing method building on the structured linguistic knowledge base “ConceptNet” (SRCM-CNet), (ii) a specific meta-model, which follows the Ecore standard, defines the rules of applying SRCM-CNet to different conceptual models to automatically detect transformation mappings, and (iii) a multistep matching and transformation process that employs SRCM-CNet. A case study is carried out to detail the working mechanism of SbACMT. Furthermore, through a systematically analysis of this case study, we validate the performance of SbACMT. We prove that SbACMT can support the automatic transformation process of conceptual models (e.g., class diagrams). The scalability of SbACMT can be improved by adapting the meta-model and predefined syntax transformation rules.

## 1. Introduction

Conceptual models, which are mainly built to represent the static characteristics of a system, can be used to refer to models formed after a conceptualization or generalization process [1]. A conceptual model represents concepts (entities) and relationships between them; it helps different stakeholders reach a clear and unified overview of a particular system or domain being modeled. In artificial intelligence field, a typical usage of conceptual models is to build expert systems and knowledge-based systems.

To achieve different kinds of purposes, several kinds of conceptual models (with modeling languages) have been proposed, such as “class diagrams,” “data flow models,” “entity-relationship models,” and “business process models.” Considering diverse research fields and practical

cases, conceptual modeling theories are defined and relevant modeling tools are employed.

In practice, conceptual models have been built for various purposes, such as simulating groundwater system [1] and improving environmental science curriculum [2]. In the context of computer science, especially in the software engineering field, a typical kind of conceptual models is “class diagrams.”

Class diagrams are built in the requirement analysis phase and being refined in the system design phase; they are defined with the UML (Unified Modeling Language) notations. Based on the object-oriented theories, a class diagram is a complete mapping of the subjects and their relations of a particular domain being analyzed.

Domain-specific conceptual models can be used to generate domain ontologies and knowledge graph (since

ontologies can be regarded as the schema of knowledge graph). The creation of semantic web can also benefit from qualified domain-specific (or domain-cross) conceptual models.

However, conceptual modeling is a time-consuming and iteratively evolve process, which is expertise required. Since conceptual models are useful to help solve complex problems, more and more specific conceptual models (in different domains, with diverse modeling languages and for different purposes) have to be built. Consequently, how to effectively build conceptual models attracts attention from both academics and industrial.

An intuitive idea is to merge/integrate the existing qualified conceptual models to get new ones. To realize this idea, model transformation methods shall be adopted. However, there are three main challenges in adopting model transformation methods to solve this problem:

- (1) Ch1: conceptual models may have different structures and formats. Furthermore, they are defined on different abstract levels. How to deal with the heterogeneous problem?
- (2) Ch2: the semantic content conveyed in conceptual models is complex and may be ambiguous. How to find the same or similarity concepts automatically?
- (3) Ch3: how to ensure and validate the correctness of the model transformation process and the new generated model?

Considering the three challenges, we propose a configurable semantic-based (semi-) automatic conceptual model transformation method (SbACMT). The main contributions of this work are as follows:

- (1) Con1: focusing on generating new class diagrams (a kind of conceptual models), we define a meta-model on a higher abstract level to unify the structure and format (syntax).
- (2) Con2: considering the semantic conveyed in class diagrams, we propose a semantic relatedness computing method to automatically detect potential transformation mappings.
- (3) Con3: we propose an integrated transformation process that combines the meta-model-based transformation theory and the semantic relatedness computing method.

SbACMT is a general-purpose model transformation methodology. It aims at serving to the transformations among different kinds of conceptual models, instead of focusing on two specific kinds of conceptual models.

This paper is organized in 7 sections. Section 2 presents the preliminaries of this work. Section 3 introduces the meta-model defined in SbACMT and illustrates the semantic relatedness computing method "SRCM-CNet." Section 4 details the integrated transformation process. Section 5 shows a case study and evaluates the performance of SbACMT. Section 6 gives the related work, while Section 7 draws a conclusion.

## 2. Preliminaries

*2.1. Model and Model Transformation.* Model and model transformation are two pilots of model-driven approaches, such as model-driven engineering (MDE), model-driven architecture (MDA), and model-based testing (MBT). These approaches have been adopted by various domains (e.g., enterprise engineering [3–5] and software engineering [6–8]).

Focusing on a specific viewpoint, a model provides abstractions of a system that allow people to have a better understanding of and to reason about it [9–11].

Models have been divided into different categories. Depending on existing forms, there are mathematical models, graph models, text models, etc. In the context of MDA [12], models are categorized as four abstraction levels: meta-meta-model, meta-model, model, and system (subject). Depending on precision, models are divided into three levels, namely: conceptual models, specification models, and implementation models [13]. Also, in MDA, a similar classification is as follows: computation-independent model (CIM), platform-independent model (PIM), and platform-specific model (PSM).

As another pilot of model-driven approaches, model transformation is a process which contains a sequence of activities operating on models. The aim of model transformation is to generate target models based on source models [14–16].

Two systematic classifications of model transformation methods are given in [17, 18]. According to the content conveyed in models, model transformations can be divided into two groups: model-to-model and model-to-text.

Depending on the abstraction level of source and target models, model transformation can be divided as follows: horizontal transformation (source and target models belong to the same abstraction level, e.g., CIM and CIM) and vertical transformation (e.g., transform CIMs to PIMs).

Model transformation methods are defined by leveraging certain model transformation technologies, such as XSLT [19], VIATRA [20], QVT [21], and ATL [22]. Table 1 illustrates briefly the above four model transformation technologies.

As stated in [22], "a vast number of model transformations are being developed and organized in complex patterns." On the one hand, some model transformation technologies provide a wide range of functions but require certain effort to learn to use them properly. On the other hand, some other transformation technologies are capable to be executed automatically; however, the precondition of applying them is strict and the transformation practices built on them have low reusability.

Besides the above model transformation technologies, some other technologies, such as Kermeta (extension of EMOF) [31], MOFM2T (model-to-text) [32], JTL (EMF-based tool) [33], Tefkat (special for MOF models) [34], and MOMENT [35] (model merging), are also being used in practice.

TABLE 1: Model transformation languages and technologies.

Languages	Characteristic	Relevant standards & technologies	Instances	Note
XSLT	Expressed as well-formed XML document; associating patterns with templates	XML, XMI [23]	[27]	Text based, text-to-text
VIATRA2	based on graph transformation techniques	VPM meta-modeling approach [24]	[28]	Graph models
QVT	Meta-models shall be conformed to MOF standards	MOF 2.0 [21]; OCL [25]	[29]	Not well tool supported
ATL	Supported by mature software tools	KM3 [26]	[30]	Requiring manual effort

SbACMT focuses on conceptual model transformation. To serve to different kinds of conceptual models, SbACMT is designed to be configurable. It belongs to the horizontal classification. SbACMT aims at finding automatically the shared and same (similar) or relevant concepts (entities) between different conceptual models.

Considering the model transformation theories, SbACMT is meta-model based and employs semantic relatedness computing method as the detecting technology.

*2.2. Semantic Relatedness Computing Methods.* Semantic relatedness is a form of measurement that quantitatively identifies the relationship between words or concepts based on the similarity or closeness of their meaning [36].

Semantic relatedness computing methods have been widely adopted. For instance, in natural language processing field, they are used to do semantic information retrieval, keyword extraction, etc.

To develop semantic relatedness computing methods, knowledge resource such as semantic thesaurus (e.g., WordNet and Wikipedia) or corpus (e.g., classics and speech drafts) is needed. Furthermore, depending on different kinds of knowledge resources, various computing methods, such as graph based and information content based, have been proposed.

A systematically classification about the knowledge resources is given in [37]. Figure 1 shows a general idea of this classification.

Comparing with the “web-based” resources, both the “linguistically constructed” and “collaboratively constructed” resources provide more precise and credible content.

Considering the development of semantic relatedness computing methods, the linguistically constructed knowledge resources are commonly used in practice.

As stated in [38], the computing methods can be divided into three groups, namely, “graph based,” “context based,” and “temporal.” Table 2 shows a general illustration of this classification.

Many research works employ WordNet as the semantic thesaurus. However, WordNet is no longer continuously updated and maintained. As an evolved semantic thesaurus, ConceptNet is built partially upon WordNet.

Considering the precision issue, this paper employs ConceptNet as the knowledge resource and adopts a graph-

based computing method to do semantic relatedness computing method to do semantic relatedness computing.

### 3. Foundations

*3.1. SbACMT\_MM.* A meta-model is a model that defines modeling rules. It is built by leveraging the meta-modeling languages. Currently, three mature meta-modeling languages are as follows: MOF [21], KM3 [26], and Ecore [43].

Ecore is a lightweight version of MOF, and it is well tool supported. Comparing with KM3, Ecore is more widely used. Focusing on the transformation of UML class diagrams, we define a meta-model “SbACMT\_MM” conforming to the Ecore standard.

Different kinds of conceptual models have different structures and formats. The structures and formats of conceptual models concern on the syntax aspect, which is easy to handle (comparing to the semantic aspect). The purpose of defining a meta-model is to unify structure and format, which may affect the transformation accuracy, and is to apply the semantic checking rules in a general way. The risk of adjusting syntax transformation rules is lower than adapting semantic checking rules.

As shown in Figure 2, SbACMT\_MM contains three packages: class diagram, semantic relatedness, and mapping rule. The “class diagram” package shows the composition of a class diagram in model transformation context. The “semantic relatedness” package defines how to do semantic relatedness computing between a pair of concepts. The “mapping rule” package defines the mechanism of selecting concept mapping pairs based on semantic relatedness computing results.

A class diagram can be defined as a six-tuple.

*3.1.1. CD= $\langle$ Domain, Class, Attribute, Relationship, Type, Object $\rangle$ .* “Domain” represents the problem domain that a class diagram built for; it defines the context of a class diagram. “Class” corresponds to a group of the same or similar objects within a domain. “Attribute” can be regarded as additional information to clearly explain the “Class.” “Relationship” stands for the relations between classes, which has four instances “generalization,” “association,” “aggregation,” and “dependency.” “Attribute” owns “Type,” which stands for a predefined set of types (e.g., integer and string). “Object” stands for the instances of “Class.” The names of

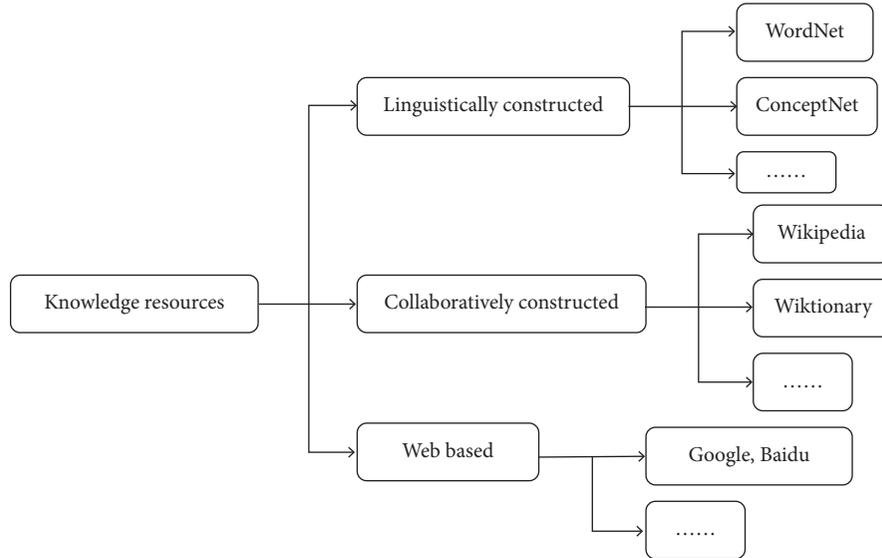


FIGURE 1: Knowledge resource classification (adapted from [37]).

TABLE 2: Semantic relatedness computing method.

Method	Instances	Research work	Note
Graph based	Path based	[37]	Shortest
	Random walk	[38]	Probability
	Co-occurrence based	[39]	WordNet glosses
Context based	Vector based	[40]	WordNet glosses
	Information theoretic	[41]	WordNet
Temporal		[42]	Time series

“Domain,” “Class,” “Attribute,” and “Relationship” correspond to “Concept” defined in the “Semantic Relatedness” package.

In the “Semantic Relatedness” package, there are four core modeling elements defined as a four-tuple.

3.1.2.  $SR = \langle \text{Concept}, \text{SemanticRelation}, \text{Type}, \text{CalculationRules} \rangle$ . “Concept” relates to the names (attribute) used in “class diagram” package. “Semantic relation” stands for a set of semantic relations (e.g., hyponym and part of) between concepts. Semantic relations are divided into two groups: direct semantic relations and iterative semantic relations. Each of the direct semantic relations has an enumeration type, whereas iterative semantic relations are combinations of the direct semantic relations. A concrete value or a specific computing rule, which is represented as “calculation rules,” is assigned to each of the semantic relations.

In the “mapping rule” package, there are two modeling elements: “MappingPair” and “Threshold.” “MappingPair” stands for all the potential mapping concept pairs, i.e., class-to-class, class-to-attribute, and attribute-to-attribute. For each kind of the mapping pairs, two thresholds (as instances

of “Threshold”) are defined to distinguish potential mapping levels, i.e., acceptable, considerable, and impossible.

3.2. *Semantic Relatedness Computing in SbACMT.* ConceptNet is a semantic network based on information from the OMCS (Open Mind Common Sense) database. It is represented as a directed graph whose nodes are concepts and edges are common sense assertions about these concepts. The concepts represent a closely related set of natural language phrases, e.g., noun phrases, verb phrases, adjective phrases, and clauses.

Currently, the latest version of ConceptNet is 5.7, which contains 304 kinds of languages and maintains 34 kinds of semantic relations among concepts.

Figure 3 is a simple illustration of the structure of ConceptNet. As shown in Figure 3, each arrow indicates a start node and an end node of an edge. In the middle of an edge, the relationship and weight between two nodes are shown. The weight means the credibility of the assertion. Between two nodes (concepts), there may exist several edges (semantic relations).

In the SbACMT context, we select 15 (out of 34) semantic relations from ConceptNet to be used to detect the potential mappings (the same or similar concepts).

As shown in Table 3, a value is assigned to each of the relations.

The selection of these fifteen semantic relations and value assigning are completed through a three-step process. First, we invited seven researchers to vote independently and to choose the relations that they thought are appropriate for judging similar and relevant concepts. The semantic relationships that have no fewer than 4 votes are finally selected. Next, we asked the seven researchers to assign a corresponding value to each of these relations based on their experience. Finally, we carried out tests based on the gold standard “RG-65” [40] to adjust these values.

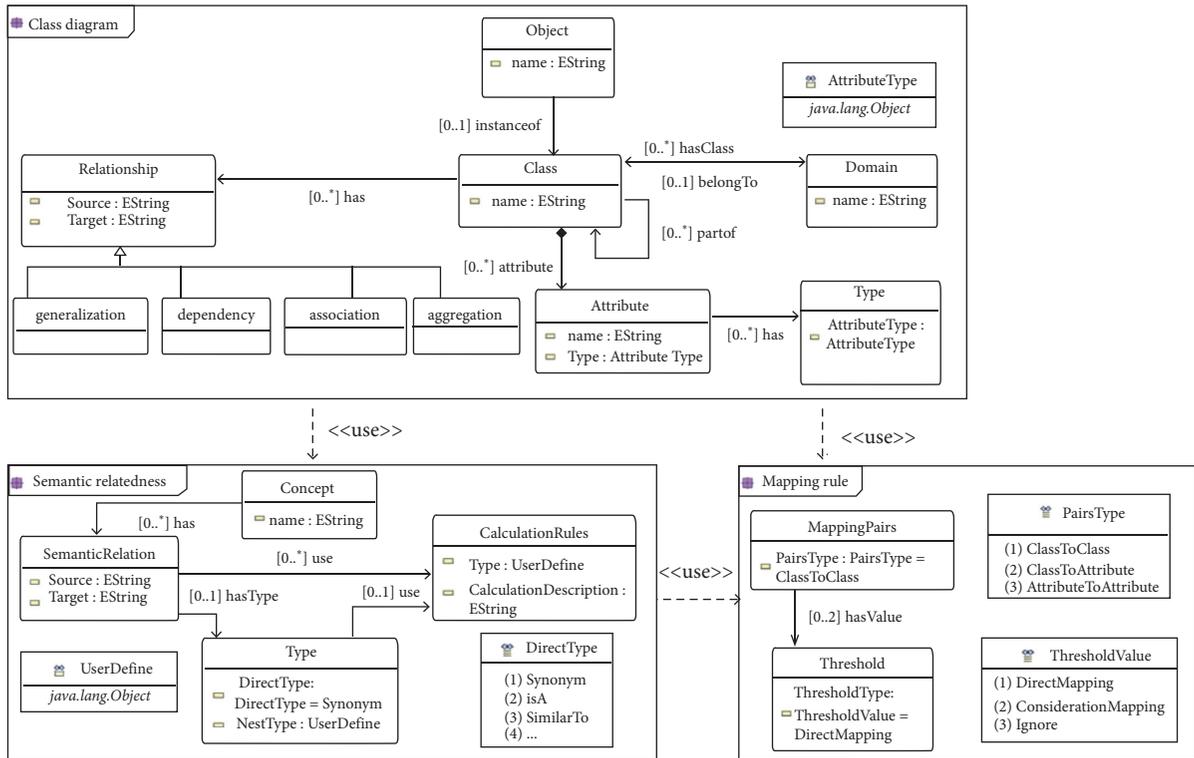


FIGURE 2: A brief illustration of SbACMT\_MM.

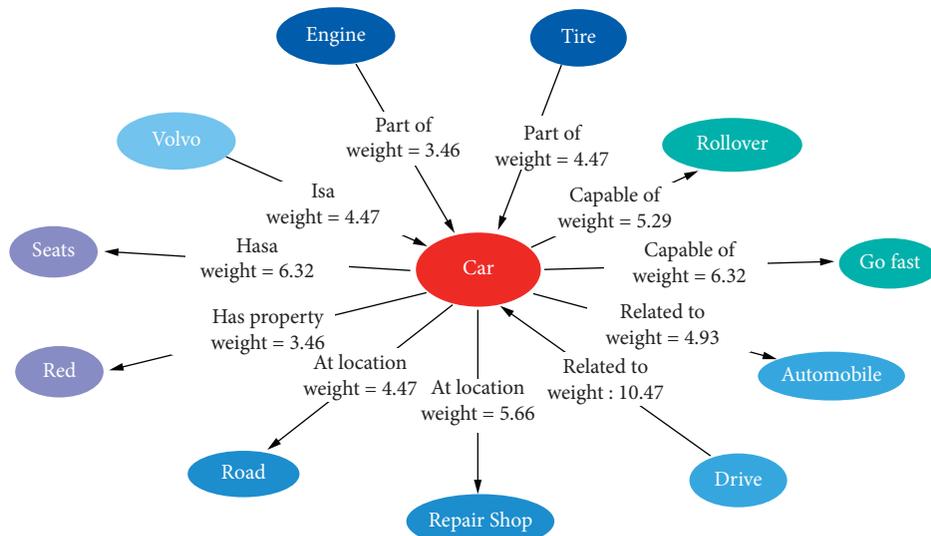


FIGURE 3: A simple illustration of ConceptNet.

For each of the comparing concept pairs, we define two algorithms to compute the semantic relatedness. Algorithm 1 detects if a direct semantic relation exists, whereas Algorithm 2 detects if an iterative semantic relation exists. The pseudocode of the two algorithms is shown as follows.

If a comparing concept pair cannot be located, a lexical analysis function will be executed first. This function aims to transform the input concept (word) to its origin form, and it deals with two main situations: words with delimiters and anagram (e.g., plural and tense).

“SR\_V” is short for semantic relatedness value, which is computed between a pair of comparing concepts (string A and string B).

#### 4. Selecting Potential Mapping Pairs

“SR\_V” ranges from 0 to 1. Thresholds on “SR\_V” are defined to distinguish potential mapping pairs. Table 4 shows three sets of these thresholds.

TABLE 3: The selected semantic relations.

Semantics relation	Sev	Example
Synonym	0.95	Sunlight $\longleftrightarrow$ sunshine
Similar to	0.9	Mixer $\longleftrightarrow$ food processor
Is a	0.7	Car $\longrightarrow$ vehicle
Related to	0.6	Car $\longleftrightarrow$ motor
Defined as	0.6	Peace $\longrightarrow$ absence of war
Distinct from	0.5	Red $\longleftrightarrow$ blue
Derived from	0.45	Pocketbook $\longrightarrow$ book
Part of	0.45	Tire $\longrightarrow$ car
Made of	0.45	Bottle $\longrightarrow$ plastic
Created by	0.45	Tree $\longrightarrow$ fruit
Etymologically related to	0.4	Folkmusiikki $\longleftrightarrow$ folk music
Has a	0.4	Car $\longrightarrow$ seat
Has context	0.4	Astern $\longrightarrow$ ship
At location	0.4	Car $\longrightarrow$ road
Located near	0.4	Table $\longleftrightarrow$ chair

The thresholds are adjusted and assigned through a similar process as the one of the selecting semantic relations from ConceptNet.

For a comparing concept pair, if their SR\_V is bigger than “\* \* -t1,” we assume the two concepts can be matched directly (acceptable). If the SR\_V is between “\* \* -t1” and “\* \* -t2,” the users or domain experts shall be asked to make the matching judgment (considerable). If the SR\_V is less than “\* \* -t2,” we assume no mapping shall be built. All the considerable mapping pairs will be listed out as potential mapping. Without manual judgments, the potential mappings will be used as transformation rules.

**4.1. Matching Process.** In SbACMT, model transformation process is divided into two phases: building potential mappings and executing transformations.

Figure 4 illustrates briefly the transformation process. After taking in two models, SbACMT extracts and classifies two groups of “Class” and “Attribute.” The first phase “building potential mappings” contains three main steps: building matching pairs among classes, building matching pairs between classes and attributes, and building matching pairs among attributes. The three steps are executed in sequence. After getting the potential mappings, a manual verification step can be added optionally. Then, the transformation phase, which concerns on those mappings, is to be executed.

**4.2. Class-to-Class Mapping Step.** The name of a class is regarded as a specific domain concept. A class may own a set of attributes. Classes are connected with each other by relations, which can be regarded as class’s attributes. While comparing two classes, their names are taken into account.

Equation (1) is defined to compute the semantic relatedness between two classes:

$$C2C\_SRV_{(i,j)} = SR\_V(C_i, C_j). \quad (1)$$

In equation (1),  $C2C\_SRV_{(i,j)}$  stands for the SR\_V between  $Class_i (C_i)$  and  $Class_j (C_j)$ , whereas  $Class_i$  comes from

the source conceptual model and  $Class_j$  comes from the target conceptual model.  $C2C\_SRV_{(i,j)}$  equals to  $SR\_V(C_i, C_j)$ , the SR\_V between two class names. All the  $C2C\_SRV_{(i,j)}$  values (calculated automatically) will be compared with thresholds “c2c-t1” and “c2c-t2” to generate potential mappings.

**4.3. Class-to-Attribute Mapping Step.** After executing the first matching step, some classes may be left unmatched. This step aims to build potential cross-level (granularity) mappings between classes and attributes. Equation (2) is used:

$$C2C\_SRV_{(i,j)} = SR\_V(C_i, A_j), \text{ or } SR\_V(A_k, C_j). \quad (2)$$

Take one class (or attribute) from the source conceptual model and one attribute (or class) from the target conceptual model, and then apply semantic relatedness computing algorithm between their names.  $C2C\_SRV_{(i,j)}$  stands for the cross-level comparing result between a specific pair of Concept  $C_i$  (or  $C_j$ ) and attribute  $A_j$  (or  $A_i$ ).

**4.4. Attribute-to-Attribute Mapping Step.** After executing the two former matching steps, some classes and attributes may still be left unmatched. For the unmatched classes, we regard them as specific parts that cannot be transformed or merged. For the unmatched attributes, equation (3) is used to build potential mappings among them:

$$A2A\_SRV = SR\_V(A_i, A_j). \quad (3)$$

The potential attribute matching pairs are only built between those attributes that belong to the potential matched classes. In equation (3), assuming  $A_i$  is an attribute of  $Class_n (C_n)$  and  $A_j$  is an attribute of  $Class_m (C_m)$ , the matching algorithm only needs to be executed if there is a potential mapping between  $C_n$  and  $C_m$ .

**4.5. Verification of Potential Mappings.** To generate transformation rules, the potential mappings have to be verified. In SbACMT, we propose two ways “verification based on relations” and “manual verification.”

The relations defined in the source conceptual models (class diagrams) are used to verify the automatically generated potential mappings:

- (1) For the relation “generalization/specialization (is a or is a kind of)”, if the father class is matched then all its children classes should be matched to the same class (or its children classes) in the target class diagram.
- (2) For the “aggregation/composition (has a or part of)” relation, the classes owning this relation can be matched to the same class in the target class diagram.
- (3) For the “dependency” and “association” relations, we regard them as “user-defined” type. They are not used as verification measurements.

For instance, if in the source class diagram, two classes  $C_i$  and  $C_n$  have a generalization relation, and a potential mapping is built between  $C_i$  and  $C_j$  (from the target class

```

/* initialize corresponding values of semantic relations and values*/
Semantic_Relation_Values = {(synonym, 0.95), (similarto, 0.9), (), (), ...}
/* Algorithm 1 detect direct semantic relations*/
Function detecting direct Semantic Relation (String A, String B){
    int SR_V=0;
    for (int i=0; i < Semantic_Relation_Values.size (); i++){
        if (Semantic_Relation_Value.get (i).get Relation (Concept Net (A, B))){
            if (Semantic_Relation_Value.get (i).get Value ()>SR_V)
                SR_V= Semantic_Relation_Value.get(i).get Value();
        }
    }
    } return SR_V;
}

```

ALGORITHM 1: The algorithm of detecting direct semantic relations.

```

/* Algorithm 2 detect iterative semantic relations*/
Function detecting iterative Semantic Relation (String A, String B){
    int temp1, temp2, SR_V=0;
    string [] inter Strings = Concept Net (A);
    for (int i=0; i < inter Strings.size (); i++){
        for(j=0; j < Semantic_Relation_Values.size (); j++){
            if (Semantic_Relation_Value.get (j).get Relation (inter Strings.get (i), B))){
                temp1 = Semantic_Relation_Value.get (j).get Value ();
                temp2 = Semantic_Relation_Value.get (i).get Value();
                if (temp1*temp2> SR_V)
                    SR_V = temp1*temp2;
            }
        }
    }
    } return SR_V;
}

```

ALGORITHM 2: The algorithm of detecting iterative semantic relations.

TABLE 4: Thresholds for judging mapping pairs.

Group	Threshold	Value
Class-to-class	c2c-t1	0.75
	c2c-t2	0.50
Class-to-attribute	c2a-t1	0.70
	c2a-t2	0.50
Attribute-to-attribute	a2a-t1	0.80
	a2a-t2	0.60

diagram), we will verify that if a mapping between  $C_n$  and  $C_j$  is automatically generated. This verification step is executed automatically.

For the “manual verification,” a visual interface is provided to the users (or experts) to show the automatically generated mappings. These mappings are divided as two groups: “acceptable” and “considerable.” If there is no manual judgment, all these mappings will be used directly as transformation rules.

## 5. Use Case

**5.1. Use Case Illustration.** We evaluate the performance of SbACMT with the following use case. In the use case, two

conceptual models (UML class diagrams) “vehicle management” and “traffic management” conforming to Ecore standard are involved. Figure 5 shows the two class diagrams.

The class diagram “vehicle management” contains three classes “person” (with seven attributes), “driving license” (with six attributes) and “vehicle” (with five attributes). The (association) relations between them are as follows: “person” possesses “vehicle” and “driving license.”

The class diagram of “traffic management” contains three classes “automobile” (with four attributes), “ticket” (with five attributes), and “driver” (with four attributes). The relationships are as follows: “ticket” concerns “automobile” and “driver” pays for the “ticket” and drives “automobile.”

SbACMT takes the two class diagrams as inputs and generates the potential mappings (pairs) between them automatically.

**5.2. Transforming Process.** The first matching step focuses on classes. A compound word (the class name) will be split into single words and then compared with potential target, respectively. For instance “driving license” is split as “driving” and “license”.

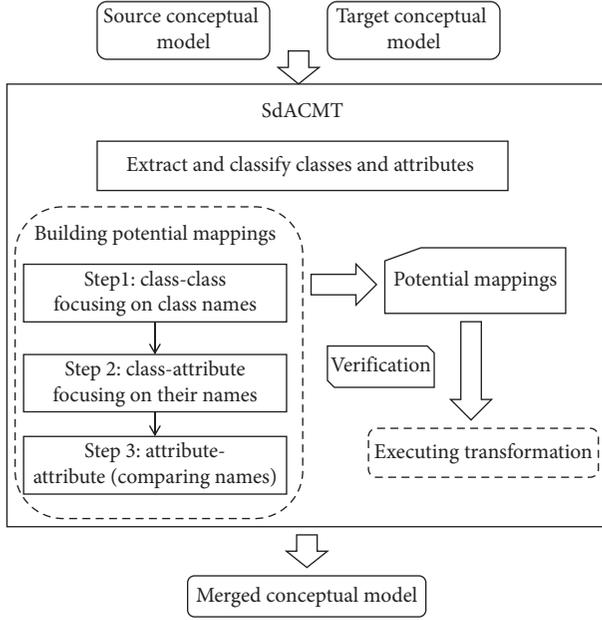


FIGURE 4: An overview of the matching process.

Table 5 shows the computing results. All three classes “person,” “driving license,” and “vehicle” are mapped with classes of the target class diagram. Since no classes left unmatched, the second matching step is skipped.

The third matching step focuses on attributes. In this step, all the attributes belonging to the already matched classes are taken into consideration. Tables 6–9 show the computing results.

## 6. Evaluation

The evaluation of SbACMT contains two dimensions: (1) focusing on the semantic relatedness computing method (parameters and constraints) and (2) focusing on the correctness of the automatically generated mappings.

*6.1. Adjusting SRCM-CNet with Gold Standard RG-65.* We use the “RG-65 gold standard” data set to test SRCM-CNet. The testing results are measured with Spearman’s rank correlation coefficient.

As stated in [36], a semantic relatedness computing method is considered reliable if an inter-rater agreement (computing results) of it is over 75%.

Considering Spearman’s rank correlation coefficient, we compare SRCM-CNet with three other semantic relatedness computing methods. Table 10 shows the testing results. SRCM-CNet gets Spearman’s rank correlation coefficient “0.9,” which is better than the other methods.

*6.2. Correctness Evaluation.* We evaluate the correctness of SbACMT on basis of the use case presented above.

The quantitative evaluation focuses on class-class and attribute-attribute mappings. The retrieved mappings and expected mappings are shown in Tables 11 and 12, respectively.

As shown in Table 12, according to the computing results, the attributes “id” and “name” (both from class “person”) match with the attribute “id” (from class “driver”). However, we only build the mapping between attributes “id” and “id,” since they have a higher SR\_V.

Inspired by the work stated in [45], performance measures are based on three indicators: precision, recall, and F-measure:

- (1) The precision (P) evaluates the quality of the automatic built transformation mappings.
- (2) The recall (R) evaluates the sufficiency of the automatic built mappings.
- (3) The F-measure (F) is the weighted harmonic average of P and R. A higher value of F is preferred.

Equation (4) shows the computing rules of “precision,” “recall,” and “F-measure”:

$$P = \frac{|\text{retrieved} \cap \text{expected}|}{|\text{retrieved}|},$$

$$R = \frac{|\text{retrieved} \cap \text{expected}|}{|\text{expected}|}, \quad (4)$$

$$F = \frac{2 \times P \times R}{P + R}.$$

The evaluation results of SbACMT are shown in Figure 6.

The performance evaluation has been performed on a computer with 2.4 GHz, i5-6200U CPU, and 8 G RAM (with Windows 10 OS and Java 9 JDK).

Generally, the precision rate and the recall rate affect each other. Considering the attribute mapping pairs, the quality of the automatic built mappings is high. The precision of automatically built class mappings is mediocre (the number of classes involved in the use case is small).

## 7. Related Works

This section presents the related works in three aspects: the usage of meta-models in practice, semantic relatedness computing methods adopted in practice, and our previous relevant work [47].

*7.1. Meta-Model Usage.* Meta-models, which are a special kind of models, define the rules of building models and can be used to verify the correctness of the models conforming to them.

Meta-models have been defined in various domains. In [48], a meta-model is defined to adopt organizational concepts to help analyze and design multiagent systems. In [49], a meta-model, which is made of four smaller meta-models, is defined to formalize software protection from man-at-the-end attacks. In [50], an accident process meta-model is defined to describe the hazard-accident process by fault propagation. In [51], a meta-model is defined to argue about compliance of requirements models and to keep track of compliance decision through the requirement modeling.

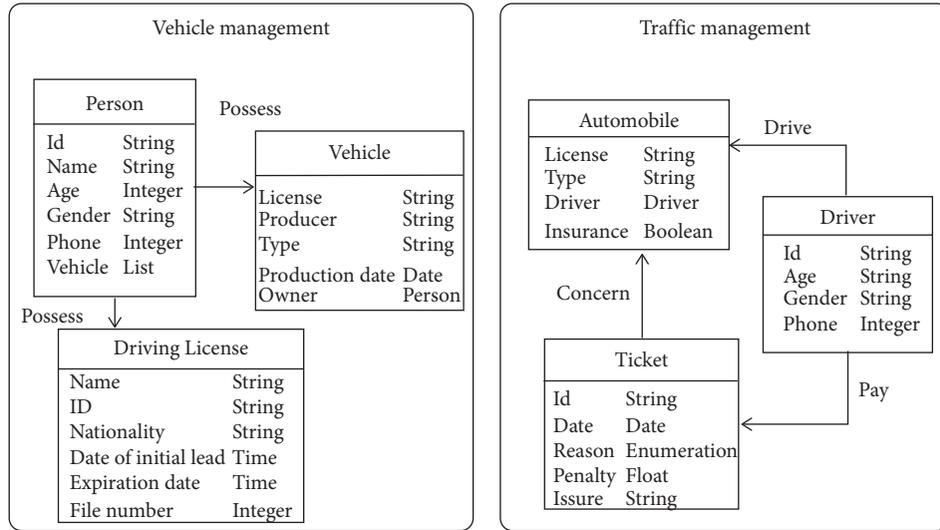


FIGURE 5: An illustration of the use case.

TABLE 5: Matching between concepts.

Vehicle\traffic	Automobile	Driver	Ticket
Person	0.0	0.6	0.36
Vehicle	0.90	0.57	0.57
Driving license	0.6	0.6	0.57

TABLE 6: Matching between attributes, Set 1.

Vehicle\automobile	License	Type	Driver	Insurance
License	1	—	—	—
Producer	0.49	0.2	0	0
Type	0.49	1	—	—
Production date	0.14	0	0.1	0
Owner	0	0	0.49	0

TABLE 7: Matching between attributes, Set 2.

Person\driver	Id	Age	Gender	Phone
Id	1	—	—	—
Name	0.92	—	—	—
Age	0.38	0.8	—	—
Gender	0.2	0.29	1	—
Phone	0	0.18	0	1
Address	0.2	0.29	0.49	0.58
Vehicle	0	0	0	0.23

TABLE 8: Matching between attributes, Set 3.

Driving license\automobile	License	Type	Driver	Insurance
Name	0.49	0.66	0	0.23
Id	0.2	0.66	0	0
Nationality	0	0.42	0	0.36
Date of initial lead	0.36	0.36	0.36	0
Expiration date	0.36	0	0	0
File number	0.36	0.57	0.36	0.36

TABLE 9: Matching between attributes, Set 4.

Driving license\driver	Id	Age	Gender	Phone
Name	0.92	—	—	—
Id	1	—	—	—
Nationality	0	0.42	0.36	0.42
Date of initial lead	0.57	0.6	0.36	0.36
Expiration date	0.36	0	0	0
File number	0.57	0.6	0.36	0.6

TABLE 10: Spearman’s rank correlation coefficient with RG-65.

Relatedness measure	RG-65
SRCM-CNet	0.90
Extended gloss overlaps [44]	0.8
Resnik [45]	0.72
Lin [46]	0.72

In [52], a meta-model is defined to help effectively build interenterprise collaborations.

The majority of current existing meta-models are defined without strict semantics or syntax. They are presented as class-based graphs or activity diagrams, and natural language is used to describe the context and the constraints, etc. Researchers and engineers focus on the content conveyed by meta-models more than meta-modeling theories and languages, which leads to the low reusability and poor readability of these meta-models. Table 13 presents the comparing results of the above meta-models.

We define SbACMT\_MM conforming to the Ecore standard, which ensures the unified semantics and syntax. In the “semantic-based conceptual model transformation” context, SbACMT\_MM sets the theoretical foundation.

*7.2. Semantic-Based Matching Methods.* Employing semantic-based analysis to determine similarity between concepts (entities) is a common research topic in various fields, such as ontology matching and data schemas matching. A certain number of semantic relatedness computing theories and methods are proposed.

In [53], semantic analysis is used to lift meta-models into ontologies to integrate eventually modeling languages. In [54], towards data integration, an automatic schema matching method built on probabilistic lexical annotation technique is proposed. In [55], a new semantic matching method is adopted to find the corresponding entities via similarity matrices. In [56], a two-step method built on state-of-the-art methods is proposed to improve ontology matching by enriching ontology mappings.

Table 14 presents the above research works and shows a comparison between them.

To support general-purpose usage, a certain number of semantic reasoners are proposed and are used in practice. In Table 15, five semantic reasoners are listed and analyzed.

Agreement maker uses different matching algorithms to aggregate their matching results, which is usually called parallel combination.

STROMA uses linguistic, structural techniques to deal with ontology matching. It adopts five matching strategies and weights the results. The system is generic as it can be used for different domains.

ASMOV utilizes lexicographic, structural, and instance-based techniques and WordNet. It incorporates a semantic validation process, which increases the accuracy of the system.

S-Match represents the ontology entity as a logical formula; the matching problem is reduced to the proposition validity problem. The input of the S-Match system is two graph structure patterns. The output is the logical relation between the nodes in the graph.

Compared with existing computing methods and semantic reasoners, we adapted ConceptNet as semantic knowledge base (with rich semantic relations and assertions). Furthermore, focusing on conceptual model (class diagrams) transformation, we tailor the semantic relations defined in ConceptNet and quantify the adopted relations with the RG-65 gold standard. The rules of applying SRCM-CNet are defined in SbACMT\_MM.

*7.3. Relevant Previous Works.* Compared with our previous work [47], the main novelty of this paper is reflected in three aspects.

First, it is the new meta-model “SbACMT\_MM”. Towards conceptual model transformation, we define this meta-model by employing the Ecore standard. SbACMT\_MM consists of three parts “class diagram,” “semantic relatedness,” and “mapping pair”; it builds the foundation of combing SRCM-CNet into automatic model transformation process.

Second, it is the adopted semantic knowledge base. In our previous work, WordNet is adapted as the basic semantic resources. In this paper, we adapted ConceptNet as the semantic knowledge base and built a local semantic thesaurus (with Neo4J) based on it. Comparing to WordNet, ConceptNet contains more content (e.g., words and semantic relations), being constantly updated and supports multilanguages.

Third, it is the evaluation part. In previous work, we evaluated the performance of the automatic generated transformation mappings only based on manual judgments. In this paper, we adopted the RG-65gold standard and Spearman’s rank correlation coefficient, which is more persuasive.

TABLE 11: Mapping built between classes.

Potential entity mappings	Retrieved mappings	Expected mappings
Person-driver	Yes	Yes
Vehicle-automobile	Yes	Yes
Driving license-driver	Yes	No

TABLE 12: Mapping built between attributes.

Potential entity mappings	Retrieved mappings	Expected mappings
Person: id-driver: id	Yes	No
Person: age-driver: age	Yes	Yes
Person: gender-driver: gender	Yes	Yes
Person: phone-driver: phone	Yes	Yes
Vehicle: license-automobile: license	Yes	Yes
Vehicle: type-automobile: type	Yes	Yes
Vehicle: owner-automobile: owner	No	Yes
Driving license: id-driver: id	Yes	No

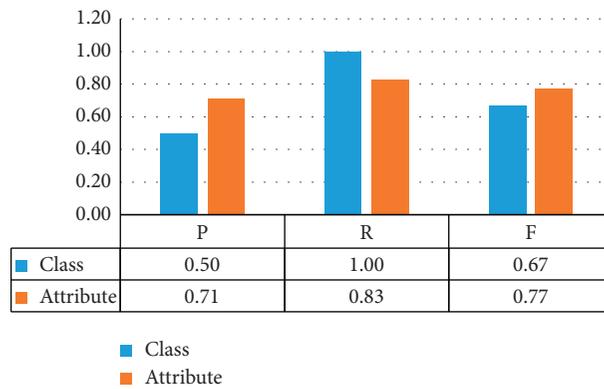


FIGURE 6: “P,” “R,” and “F” evaluation results.

TABLE 13: Meta-models defined in practice.

Ref.	Serving domain	Meta-modeling language adopted	Existing form
[48]	Software engineering	No	Graph & nature and mathematical languages description
[49]	Software protection	No	Class diagram & nature language description
[50]	System engineering	SysML from implementation view	Graphs + natural language description
[51]	Requirements engineering	No	Class-based description
[52]	Enterprise engineering	No	Class-based diagram and natural language description

TABLE 14: Research works concerning semantics.

Ref.	Serving filed	Semantics resource & adopted methods	Note
[53]	Software development	Manual	Ecore to OWL
[54]	Data integration	Word net	Combing lexical annotation
[55]	Ontology matching	Concept net	Syntax & semantic
[56]	Ontology matching	Adopted mature semantic reasoner	Two-step approach

In social network field, as presented in [57–59], semantic checking technologies are also adapted to build connections between entities (e.g., events, persons, and comments).

TABLE 15: Existing semantic reasoners.

Name	Semantic relations	Semantic source	Note
Agreement maker [60]	Is a	WordNet	Hybrid pattern matching system
STROMA [56]	Equal, is a, part of, related	WordNet, UMLS, Open Thesaurus	Two steps (1) initial matching (2) semantic enrichment
ASMOV [61]	Equal, is a	WordNet	One step Linguistic, structural, instance-based techniques
S-Match [62]	Equal, is a, related	WordNet	One step Pattern layer based matching system

## 8. Conclusion

There are many kinds of conceptual models defined and used in diverse domains. Reusing and merging existing conceptual models to generate new conceptual models are always required in practice.

To accelerate the building process of conceptual models, this paper proposes a general configurable (semi-) automatic conceptual model (class diagrams) transformation method. A semantic relatedness computing method, which is mainly built on the structured knowledge base ConceptNet, is employed as the potential mapping-detecting technology.

SbACMT follows the meta-model based model transform theory. A specific meta-model “SbACMT-MM” defines the rules of applying SRCM-Net to generate mappings between conceptual models. Considering the granularity of conceptual models, we also propose an integrated mapping and transformation process in which three matching steps are divided. User effort can be optionally involved in this process at certain points, such as judging the intermediate potential mapping pairs.

SbACMT is configurable in two aspects. First, by adapting the “class diagram” part of SbACMT-MM, SbACMT can support other kinds (besides class diagrams) of conceptual model transformations. Second, by adapting the thresholds assigned in Table 4, SbACMT can serve for customized transformation. Moreover, the values assigned to semantic relationships can also be adjusted by users.

SbACMT aims to serve to transformations of different kinds of conceptual models. It can accelerate the process of building new conceptual models in general.

SbACMT is well tool supported. As shown with the evaluation of the use case, SbACMT can achieve automatic conceptual model transformation with effective performance. A potential research direction of SbACMT concerns about contextual information. While calculating the semantic relatedness between concept pairs, the contextual information shall be taken into account. Two possible solutions are as follows: making use of the assertions (on the edges) of ConceptNet and combining other semantic resources together with ConceptNet.

## Data Availability

The data set RG-65 was used to support the findings of this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the “Fundamental Research Funds for the Central Universities (no. 3082018NS2018057).”

## References

- [1] D. Haynes, J. Brodie, J. Waterhouse, Z. Bainbridge, D. Bass, and B. Hart, “Assessment of the water quality and ecosystem health of the great barrier reef (Australia): conceptual models,” *Environmental Management*, vol. 40, no. 6, pp. 993–1003, 2007.
- [2] K. P. J. Fortuin, C. S. A. Van Koppen, and R. Leemans, “The value of conceptual models in coping with complexity and interdisciplinarity in environmental sciences education,” *Bioscience*, vol. 61, no. 10, pp. 802–814, 2011.
- [3] D. Chen, G. Doumeings, and F. Vernadat, “Architectures for enterprise integration and interoperability: past, present and future,” *Computers in Industry*, vol. 59, no. 7, pp. 647–659, 2008.
- [4] K. Kosanke, F. Vernadat, and M. Zelm, “CIMOSA: enterprise engineering and integration,” *Computers in Industry*, vol. 40, no. 2-3, pp. 83–97, 1999.
- [5] F. Benaben, W. Mu, N. Boissel-Dallier et al., “Supporting interoperability of collaborative networks through engineering of a service-based Mediation information system (MISE 2.0),” *Enterprise Information Systems*, vol. 9, no. 5-6, pp. 556–582, 2015.
- [6] T. Yue, L. C. Briand, and Y. Labiche, “Facilitating the transition from use case models to analysis models,” *ACM Transactions on Software Engineering and Methodology*, vol. 22, no. 1, pp. 1–38, 2013.
- [7] F. Rademacher, J. Sorgalla, and S. Sachweh, “Challenges of domain-driven microservice design: a model-driven perspective,” *IEEE Software*, vol. 35, no. 3, pp. 36–43, 2018.
- [8] C. Vögele, A. van Hoorn, E. Schulz, W. Hasselbring, and H. Krcmar, “WESSBAS: extraction of probabilistic workload specifications for load testing and performance prediction—a model-driven approach for session-based application systems,” *Software & Systems Modeling*, vol. 17, no. 2, pp. 443–477, 2018.
- [9] A. W. Brown, “Model driven architecture: principles and practice,” *Software and Systems Modeling*, vol. 3, no. 4, pp. 314–327, 2004.
- [10] T. Kühne, “Matters of (meta-) modeling,” *Software & Systems Modeling*, vol. 5, no. 4, pp. 369–385, 2006.

- [11] J. Bézivin, "Model driven engineering: an emerging technical space," *International Summer School on Generative and Transformational Techniques in Software Engineering*, pp. 36–64, Springer, Berlin, Germany, 2005.
- [12] S. J. Mellor, K. Scott, A. Uhl et al., *MDA Distilled: Principles of Model-Driven Architecture*, Addison-Wesley Professional, Boston, MA, USA, 2004.
- [13] M. Fowler, K. Scott, and G. Booch, "UML distilled, object oriented series," 1999.
- [14] M. Belaunde, C. Casanave, D. DSouza et al., "MDA guide version 1.0," 2003.
- [15] A. G. Kleppe, J. Warmer, J. B. Warmer et al., *MDA Explained: The Model Driven Architecture: Practice and Promise*, Addison-Wesley Professional, Boston, MA, USA, 2003.
- [16] L. Tratt, "Model transformations and tool integration," *Software & Systems Modeling*, vol. 4, no. 2, pp. 112–122, 2005.
- [17] K. Czarnecki and S. Helsen, "Classification of model transformation approaches," *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, vol. 45, no. 3, pp. 1–17, 2003.
- [18] T. Mens and P. Van Gorp, "A taxonomy of model transformation," *Electronic Notes in Theoretical Computer Science*, vol. 152, pp. 125–142, 2006.
- [19] Objects X S L F, "W3C recommendation," 1999.
- [20] D. Varró and A. Balogh, "The model transformation language of the VIATRA2 framework," *Science of Computer Programming*, vol. 68, no. 3, pp. 214–234, 2007.
- [21] Q. V. T. Omg, "Meta object facility (mof) 2.0 query/view/transformation specification," 2008.
- [22] F. Jouault, F. Allilaire, J. Bézivin et al., "ATL: a model transformation tool," *Science of Computer Programming*, vol. 72, no. 1–2, pp. 31–39, 2008.
- [23] OMG X M L, "Metadata interchange (XMI) specification," version," 2000.
- [24] D. Varró and A. Pataricza, "VPM: a visual, precise and multilevel metamodeling framework for describing mathematical domains and UML (the mathematics of metamodeling is metamodeling mathematics)," *Software & Systems Modeling*, vol. 2, no. 3, pp. 187–210, 2003.
- [25] J. Cabot and M. Gogolla, "Object constraint language (OCL): a definitive guide," *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pp. 58–90, Springer, Berlin, Germany, 2012.
- [26] F. Jouault and J. Bézivin, "KM3: a DSL for metamodel specification," *International Conference on Formal Methods for Open Object-Based Distributed Systems*, pp. 171–185, Springer, Berlin, Germany, 2006.
- [27] J. Kovse and T. Härder, "Generic XMI-based UML model transformations," *International Conference on Object-Oriented Information Systems*, pp. 192–198, Springer, Berlin, Germany, 2002.
- [28] D. Varró and A. Pataricza, "Generic and meta-transformations for model transformation engineering," *International Conference on the Unified Modeling Language*, pp. 290–304, Springer, Berlin, Germany, 2004.
- [29] V. De Castro, E. Marcos, and J. M. Vara, "Applying CIM-to-PIM model transformations for the service-oriented development of information systems," *Information and Software Technology*, vol. 53, no. 1, pp. 87–105, 2011.
- [30] M. Tisi, J. Cabot, and F. Jouault, "Improving higher-order transformations support in ATL," *International Conference on Theory and Practice of Model Transformations*, pp. 215–229, Springer, Berlin, Germany, 2010.
- [31] J. M. Jézéquel, O. Barais, and F. Fleurey, "Model driven language engineering with kermeta," *International Summer School on Generative and Transformational Techniques in Software Engineering*, pp. 201–221, Springer, Berlin, Germany, 2009.
- [32] J. Oldevik, T. Neple, R. Grønmo et al., "Toward standardised model to text transformations," *European Conference on Model Driven Architecture-Foundations and Applications*, pp. 239–253, Springer, Berlin, Germany, 2005.
- [33] A. Cicchetti, D. Di Ruscio, R. Eramo et al., "JTL: a bidirectional and change propagating transformation language," *International Conference on Software Language Engineering*, pp. 183–202, Springer, Berlin, Germany, 2010.
- [34] M. Lawley and J. Steel, "Practical declarative model transformation with Tefkat," in *Satellite Events at the MoDELS 2005 Conference, Lecture Notes in Computer Science*, J. M. Bruel, Ed., Vol. 3844, Springer, Berlin, Germany, 2006.
- [35] A. Boronat, J. Carsí, and I. Ramos, "Exogenous model merging by means of model management operators," *Electronic Communication of the European Association*, vol. 20, 2006.
- [36] Y. Feng, E. Bagheri, F. Ensan et al., "The state of the art in semantic relatedness: a framework for comparison," *The Knowledge Engineering Review*, vol. 32, 2017.
- [37] M. Jarmasz and S. Szpakowicz, "Roget's thesaurus and semantic similarity," in *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pp. 212–219, Borovets, Bulgaria, September 2003.
- [38] T. Hughes and D. Ramage, "Lexical semantic relatedness with random graph walks," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 581–589, Prague, Czechia, June 2007.
- [39] R. L. Cilibrasi and P. M. B. Vitanyi, "The google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370–383, 2007.
- [40] T. Pedersen, "Duluth: measuring degrees of relational similarity with the gloss vector measure of semantic relatedness," in *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 497–501, Montréal, Canada, June 2012.
- [41] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in WordNet," in *Proceedings of the European Conference on Artificial Intelligence*, Trento, Italy, August 2004.
- [42] K. Radinsky, E. Agichtein, E. Gabrilovich et al., "A word at a time: computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th international conference on World wide web*, pp. 337–346, Hyderabad, India, April 2011.
- [43] D. Steinberg, F. Budinsky, E. Merks et al., *EMF: Eclipse Modeling Framework*, Pearson Education, London, UK, 2008.
- [44] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *Proceedings of the International Joint Conferences on Artificial Intelligence*, pp. 805–810, Acapulco, Mexico, August 2003.
- [45] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," 1995, <http://arxiv.org/abs/9511007>.
- [46] D. Lin, "An information-theoretic definition of similarity," *International Conference on Machine Learning*, vol. 98, pp. 296–304, 1998.
- [47] T. Wang, S. Truptil, F. Benaben et al., "A meta-model based automatic conceptual model-to-model transformation methodology," in *Proceedings of the 6th International*

- Conference of MODELSWARD*, pp. 586–593, Funchal, Portugal, January 2018.
- [48] J. Ferber and O. Gutknecht, “A meta-model for the analysis and design of organizations in multi-agent systems,” in *Proceedings International Conference on Multi Agent Systems (Cat. No. 98EX160)*, pp. 128–135, IEEE, Paris, France, July 1998.
- [49] C. Basile, D. Canavese, L. Regano, P. Falcarin, and B. D. Sutter, “A meta-model for software protections and reverse engineering attacks,” *Journal of Systems and Software*, vol. 150, pp. 3–21, 2019.
- [50] Q. Wei, J. Jiao, S. Zhou et al., “Research on accident process meta-modeling based on SysML,” in *Proceedings of the 2015 First International Conference on Reliability Systems Engineering (ICRSE)*, IEEE, HoChiMinh City, Vietnam, pp. 1–6, October 2015.
- [51] A. Siena, A. Perini, A. Susi et al., “A meta-model for modelling law-compliant requirements,” in *Proceedings of the 2009 Second International Workshop on Requirements Engineering and Law*, IEEE, Atlanta, GA, USA, pp. 45–51, September 2009.
- [52] A. Montarnal, W. Mu, F. Benaben, J. Lamothe, M. Laurus, and N. Salatge, “Automated deduction of cross-organizational collaborative business processes,” *Information Sciences*, vol. 453, pp. 30–49, 2018.
- [53] G. Kappel, E. Kapsammer, H. Kargl et al., “Lifting metamodels to ontologies: a step to the semantic integration of modeling languages,” *International Conference on Model Driven Engineering Languages and Systems*, pp. 528–542, Springer, Berlin, Germany, 2006.
- [54] L. Po and S. Sorrentino, “Automatic generation of probabilistic relationships for improving schema matching,” *Information Systems*, vol. 36, no. 2, pp. 192–208, 2011.
- [55] M. Keshavarz and Y. H. Lee, “Ontology matching by using ConceptNet,” in *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference*, pp. 1917–1925, Phuket, Thailand, December 2012.
- [56] P. Arnold and E. Rahm, “Enriching ontology mappings with semantic relations,” *Data & Knowledge Engineering*, vol. 93, pp. 1–18, 2014.
- [57] J. Li, C. Liu, and M. Islam, “Keyword-based correlated network computation over large social media,” in *Proceedings of the 30th IEEE International Conference on Data Engineering (ICDE)*, pp. 268–279, Chicago, IL, USA, April 2014.
- [58] J. Li, C. Liu, J. X. Yu, Y. Chen, T. Sellis, and J. S. Culpepper, “Personalized influential topic search via social network summarization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1820–1834, 2016.
- [59] J. Li, T. Cai, K. Deng et al., “Community-diversified influence maximization in social networks,” *Information Systems*, vol. 92, pp. 1–12, 2020.
- [60] I. F. Cruz, F. P. Antonelli, and C. Stroe, “Agreement maker,” *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1586–1589, 2009.
- [61] Y. Jean-Mary and M. Kabuka, “Asmov: Ontology alignment with semantic validation,” in *Proceedings of the Joint SWDB-ODBIS Workshop*, pp. 15–20, Vienna, Austria, September 2007.
- [62] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, “S-Match: an algorithm and an implementation of semantic matching,” *European Semantic Web Symposium*, pp. 61–75, Springer, Berlin, Germany, 2004.