Model Transformation by Example with Statistical Machine Translation

Karima Berramla[1,3][a], El Abbassia Deba[1][b], Jiechen Wu[2][c], Houari Sahraoui[2][d] and Abou El Hassen Benyamina[1][e]

[1]*LAPECI Laboratory, Oran 1 University Ahmed Ben Bella, Algeria*

[2]*DIRO, Université de Montréal, Canada*

[3]*Ain Temouchent University Center, Algeria*

berramla.karima@edu.univ-oran1.dz, abbassia.deba@univ-oran1.dz, jiechen.wu@umontreal.ca, sahraouh@iro.umontreal.ca, benyanabou@yahoo.fr

Keywords: Model Transformation, Model Transformation by Example, Learning System, SMT System, IBM1 Model.

Abstract: In the last decade, Model-Driven Engineering (MDE) has experienced rapid growth in the software development community. In this context, model transformation occupies an important place that automates the transitions between development steps during the application production. To implement this transformation process, we require mastering languages and tools, but more importantly the semantic equivalence between the involved input and output metamodels. This knowledge is in general difficult to acquire, which makes transformation writing complex, time-consuming, and error-prone. In this paper, we propose a new model transformation by example approach to simplify model transformations, using Statistical Machine Translation (SMT). Our approach exploits the power of SMT by converting models in natural language texts and by processing them using models trained with IBM1 model.

1 INTRODUCTION

Model transformation is not a new activity. It has been recognized as an essential component since the appearance of application development techniques, but it has essentially been implemented using general purpose programming languages. The emergence of the Model Driven architecture (MDA) (Kleppe et al., 2003) paradigm has given another perspective to model transformation by the introduction of dedicated languages and tools.

Although dedicated languages and tools advanced considerably the state of the practice, defining model transformations still requires a lot of effort and time. The current challenge in MDE is not only (i) how to choose the best technique through which we define and generate models and metamodels in a simple way but also (ii) how to automate transformation process, to alleviate the burden on developers who has to deal with a variety of domain specific languages in which models are expressed.

In this paper, we propose a new approach based on a statistical machine translation (SMT) system for model transformation by example. This approach has several advantages. Firstly, it allows transforming models without using a transformation language that requires mastering its syntax and the semantic between source and target metamodels. Secondly, it reuses SMT, a technique with a proven track record in natural language translation. Although SMT is mainly used for natural language translation, this technique was already used to solve some domain-specific transformation problems, but in an ad-hoc manner. An interesting example of such a usage is the work by Alghawanmeh et al. (Al-Ghawanmeh and Smaïli, 2017), which translates Arab vocal improvisation to instrumental melodic accompaniment. We evaluated our approach on varous transformation examples. The results show that SMT-based model transformation can be used successfully with a reasonable number of examples.

The remainder of this paper is organized as follows. Section 2 defines the crucial problem of model transformations. The proposed approach based on a statistical machine translation system is presented in Section 3 and a case study illustrates it in the sec-

[a] https://orcid.org/0000-0002-2847-4895
[b] https://orcid.org/0000-0003-2948-2093
[c] https://orcid.org/0000-0002-4011-0859
[d] https://orcid.org/0000-0001-6304-9926
[e] https://orcid.org/0000-0003-4778-0123

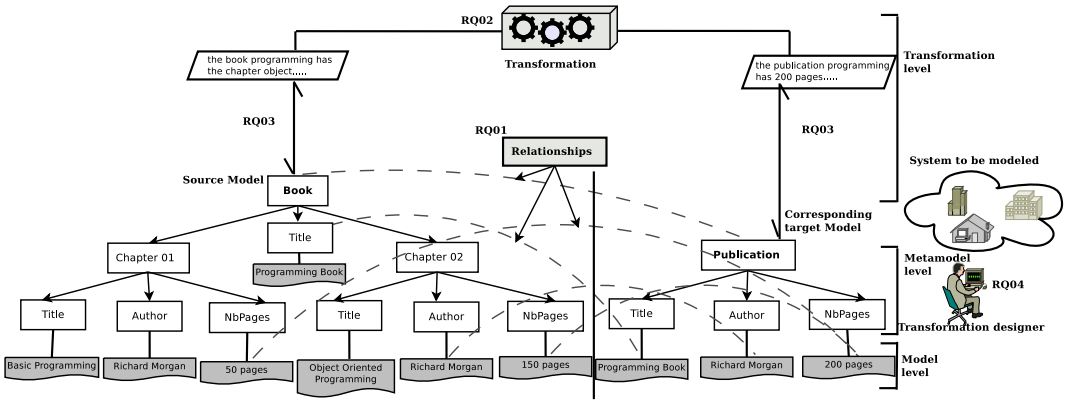

Figure 1: Model transformation problems.

tion 4. Section 5 describes the evaluation of the approach. In Section 6 the related work is analyzed and discussed. Section 7 concludes the work by focusing on the advantages and limitations of the proposed approach.

2 PROBLEM STATEMENT

In MDE context, one founding assumption is that automated model transformation reduces the cost and time of development and facilitates the development through code generation. Model transformations also allow providing intermediate bridges between the different development phases of an application. Usually, transformation programs express the automatic translation of a model to another model according to a set of transformation rules. Writing these rules requires on the one hand, to have a good knowledge about the semantics of each input and output metamodels and on the other hand, to have a good knowledge of a specific language in order to implement these transformation rules. In this paper, we are interested in the challenge of defining transformations with a minimum of domain knowledge. To address this challenge, we formulate the following research questions, illustrated through the figure 1:

RQ1: *Can we define the relationships between input and output models using natural language?*

RQ2: *Can we generate the output models automatically without using transformation rules that require to use of a specific language?*

RQ3: *Can we use the natural language to define our models or/and metamodels without using a specific language in MDE context?*

RQ4: *Is there an optimal technique to model systems without mastering a specific language or a modeling tool?*

When the model transformation is based on the specification of the correspondence rules between the elements of the source and the target metamodels, the domain experts have to implement it manually. To circumvent this burden, different research contributions investigate the idea of deriving specification or concrete transformation from example through ad-hoc or machine learning algorithms. The proposed solutions are generally known as Model Transformation By Example (MTBE). MTBE approaches use with different techniques such as genetic programming (Baki and Sahraoui, 2016) and Ad-hoc Algorithm (Varró, 2006).All these approaches attempt to generalize the correspondence between source and target model elements in the form of transformation rules written in a specification or implementation language. Learning complex structure such as rules is, however, a complex problem.

Another perspective on transformation automation is to reuse the large body of knowledge on natural language translation, which is a very similar problem. In particular, statistical machine translation can be adapted to MTBE. This technique is based on the statistical models whose parameters are derived using a parallel (or bilingual) corpus in the learning phase. Generally, another corpus is also used, that is known as monolingual corpus, to define the word resemblance in target language (Koehn, 2009).

We believe that SMT is suitable to our problem because the translation is implemented through the statistical training using a set of examples without requiring transformation languages.

3 MTBE USING SMT

Our objective is to define a transformation mechanism that is based on a corpus of transformation examples(TE). Our approach can be divided into three-steps: (1) data preparation to map transformation ex-

amples into training data, (2) SMT training and (3) the use of the trained SMT to actually transform models. In the next, we explain these steps in detail.

3.1 Data Preparation

This step is a crucial part of our approach since it allows to prepare the data that will be used to train the statistical machine translation models. Data preparation is done into two sub-steps: (1) transforming source and target models into textual representations, i.e., natural language sentences, and (2) aligning source and target sentences that describe semantically-equivalent source and target elements. Figure 2 summarizes the data preparation step.

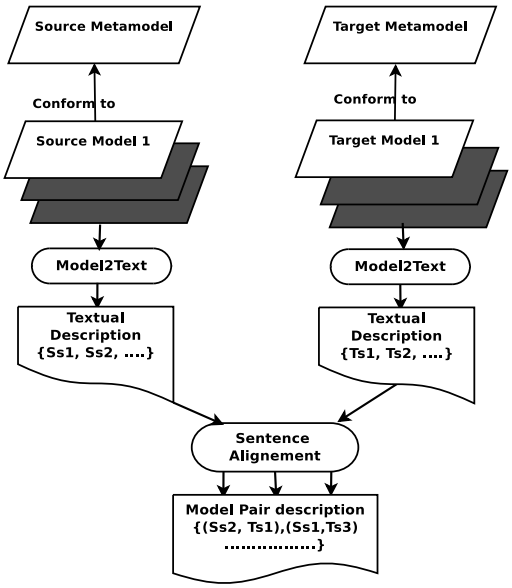

Figure 2: Corpus preparation process.

As in MDE, a model is an instance of its metamodel, it can be decomposed into elements according to the metamodel structure and constraints. Let us consider the transformation problem of UML class diagrams to relational schemas. For example, in a given UML class diagram, one element can be a class "Student" with its attribute "name". For each element, we generate one or more sentences in a natural language. Source models are translated into French, e.g., "la classe Student possède attribut name", which means "class student has attribute name". Target models are translated into English. For example, in a relational schema corresponding to the considered class diagram, we can find a table "Student" with a column "name". This model element will be translated into English as "Table Student has column name". The translation of model elements into natural language sentences can be done manually, as it can be automated as a simple model-to-text transformation.

Observation 1: Ambiguity Problem

To solve the problem of ambiguity, we have proposed a parallel corpus for each transformation example by considering the source metamodel as the source language and the target metamodel as also the target language.

Once the source and the target models are translated into natural language sentences, the next step consists in aligning source and target sentences that describe semantically-equivalent elements. Here again, the alignment can be automatic if the input and the output models are created simultaneously and automatically (in our case, this process is done by using Acceleo language for following example). Otherwise, it can be done manually.

3.2 SMT Training

After the data preparation, the set of all aligned sentences of all the pair examples define the parallel corpus, which will be used later to train the translation model. Additionally, the sentences of all the target model examples define the monolingual corpus that will be used to derive the language model.

SMT training is based on the parallel and monolingual corpora of the translation system to be used. In the following, we explain the basic concepts of SMT system and we present the translation and the language models that are used in this step. Figure 4 describes the training process of our SMT system. The first model to train is the translation model. According to (Brown et al., 1993), a sentence f in a source language has a possible translation to e in a target language according to the probability $P(e|f)$. Starting from the theorem of Bayes on the pair of sentences (s,t), the probability $P(e|f)$ is computed according to the following equation 1.

$$argmaxP(e|f) = argmax\frac{P(f|e).P(e)}{P(f)}. \quad (1)$$

The main objective of statistical translation is to find the best translation $\hat{e}$, i.e. the value that maximizes $P(e|f)$. In addition, the probability $P(f)$ of the source sentence f has no influence on the result of this equation, because it is fixed. Sequentially, SMT translation process is defined by this simplified equation 2 that maximizes the probability $P(e|f)$ according to realized observations in a parallel corpus.

$$\hat{e} = argmaxP(e|f) = argmaxP(f|e).P(e). \quad (2)$$

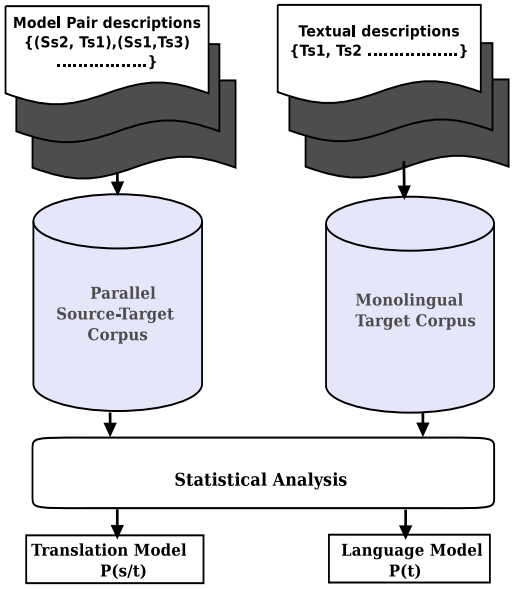

Figure 3: Training process.

$P(f|e)$ defines a translation model based on interlingual probabilities and $P(e)$ explains a language model to evaluate the probability of a word-sequence. This equation and the following ones are explained through case study in Section 4. In the following, we detail the translation and language models.

3.2.1 Translation Model

The translation model defines the probability that a word or word-sequence in a source language will be translated into one or more words in a target language. This model compute the probability of a target sentence $e_1^I = e_1 ... \ e_I$ to be associated to a sentence from the source language $f_1^J = f_1 ... f_J$. In our work, we experimented with IBM1 model. Its translation model is defined according to the following equation (Brown et al., 1993).

$$P(f_1^J|e_1^I) = \frac{\varepsilon}{(I+1)^J} \sum_{a_1=0}^{I} ... \sum_{a_J=0}^{I} \prod_{j=1}^{J} t(f_j|e_{aj}). \quad (3)$$

Where $\sum_{a_1=0}^{I} .. \sum_{a_J=0}^{I}$ defines the possible alignment and $\prod_{j=1}^{J} t(f_j|e_{aj})$ explains the translation probability of f_j to e_{aj} apply to all words in the source sentence.

3.2.2 Language Model

The objective of SMT is not only to produce a word-by-word translation sequence as output, but also to guarantee the grammatical reasonableness of the results in the target language. To achieve this objective, many researchers propose to adapt language model for machine translation system. This model is defined by the following equation (Brown et al., 1993).

$$P(e_1^I) \approx \prod_{i=1}^{I} P(e_i|e_{i-n+1}, ...e_{i-1}). \quad (4)$$

3.3 Model Transformation with SMT

In this sub-section, we describe our model transformation process using SMT system. Figure 4 gives an overview of this process. Firstly, a model to be transformed is translated into French sentences as described in the data preparation process. Then, we use the trained translation and language models to produce an equivalent English set of sentences describing the target model. Finally, this textual description is mapper to a fully fledged target model according to the associated metamodel.

The SMT translation is performed in two steps: a decoding phase and a sentence and a model generation phase.

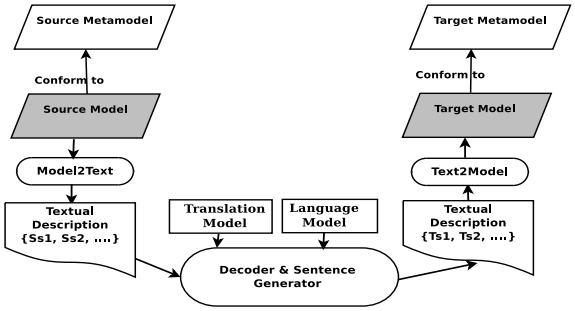

Figure 4: Model transformation process with SMT system.

3.3.1 Decoder

The decoder is the main component of the SMT system. It consists in providing an output text from a source text based on translation and language models. This treatment is defined by the following equation (Brown et al., 1993) and illustrated in Section 4:

$$\hat{e} = argmax P(f_1^J|e_1^I)P(e_1^I) = P(f_1^J, \hat{a}_1^J|f_1^I)P(e_1^I). \quad (5)$$

$P(f_1^J, \hat{a}_1^J|f_1^I)$ defines the probability of the best translation that is based on the highest alignment probability (in equation 5 this alignment is presented by $\hat{a}_1^J|f_1^I$).

3.3.2 Sentence and Model Generation

After the decoding using word-by-word translation, the sentence generator assemble the translated words into sentences according to the language model. From these sentences, we generate the target model according to the data preparation process.

4 ILLUSTRATIVE CASE STUDY

To better illustrate the proposed approach, we discuss UML class diagram to relational schema. This case has been used in several papers in order to evaluate model transformation approaches (see for example (Baki and Sahraoui, 2016)).

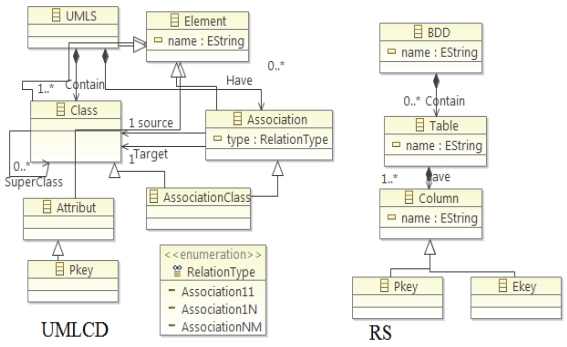

Figure 5: Source and target metamodels.

4.1 Data Preparation

In this step, we are interested to define the parallel and monolingual corpora in an automatic way by executing a model-to-text transformations written in Acceleo language. The generated sentences are apired according to the paired model elements. Figure 6 shows an example of source-target model pair together with an excerpt of the paired sentences set.

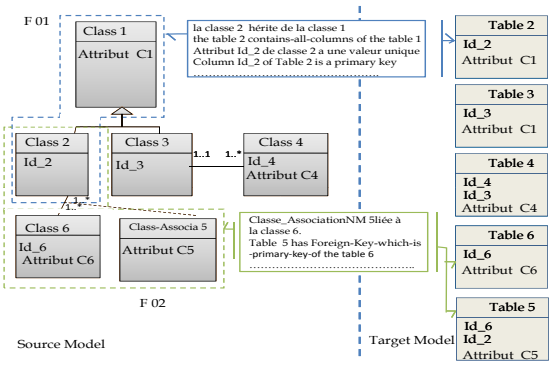

Figure 6: Data preparation process for UMLCD2RS.

4.2 SMT Training

In this step, we describe the computation process of both translation and language models.

Translation Model. In general, the translation model gives the probability of the word translation from a source language to another word in a target language. Based on this probability, we generate the target words that will be used then as a data to create the target model. In the training phase, we calculate the parameters of Equation 1 that will be useful in the test phase in order to compute the probability of each word translation. The calculation of these IBM1 model parameters is based on the Expectation-Maximization (EM) algorithm. This algorithm allows to maximize the reasonableness of the parameters to be learned (Dempster et al., 1977). The following table describes some values of translation table that defines IBM1 parameters.

Table 1: Translation table values.

P(f|e)	value	P(f|e)	value
P(attribut|column)	0.999	P(est|column)	0.007
P(association1n |foreign-key)	0.499	P(possède |foreign-key)	1e-12
P(classe |has)	0.015	P(possède |has)	0.999
P(valeur |primary)	0.497	P(unique |key)	0.497
P(classe|table)	0.836	P(unique|table)	1e-12
P(la|has)	0.103	P(la|the)	0.999

Language Model. In SMT systems, the language model is used to compute the probability of P(e) for a word-sequence t in a target language that $e = m_1, m_2, m_3 .... m_n$. In our case, we used a bi-gram language model that follows this probability (Brunning, 2010):

$$P(m_i|m_{i-1}) = \frac{c(m_{i-1}m_i)}{c(m_{i-1})} \quad (6)$$

Where $c(m_{i-1}m_i)$ defines the number of this word sequence $m_{i-1}m_i$ and $c(m_{i-1})$ explains the number of m_{i-1} word. The following table describes the language model values of UMLCD2RS example.

Table 2: Language model values.

$P(m_i \mid m_{i-1})$	value	$P(m_i \mid m_{i-1})$	value
P(table|the)	0.992	P(there|the)	0.00
P(y|table)	0.349	P(column|table)	0.00
P(primary|a)	0.833	P(key|a)	0.00
P(has|x)	0.342	P(column|x)	0.00
P(has|primary)	0.000	P(key|primary)	0.97
P(table|the)	0.992	P(another-defined|the)	0.00

4.3 Transformation with SMT

To illustrate our translation, we select a set of sentences written in natural language that define class3, class4 and relationship between them in the source model of figure 6.

Table 3: Comparative study of model transformation techniques.

Authors	Input Elements	Meta-modeling Tools	Used Method	Output Elements
Our Paper	Sentences/Models	Natural language	SMT system	Sentences/Models
(Hammoudi et al., 2014)	ST-Metamodels	Ecore Language	Matching tools	mappings
(Berramla et al., 2017b)	ST-Metamodels	Ecore Language	Matching Alg	Tr rules(A/C) [2]
(Berramla et al., 2017a)	ST-Models	Ecore Language	FCA-Matching	Tr rules (A/C)[2]
(Kessentini et al., 2012)	Bloks	///	PSO and SA	Models

Elt1: *attribut Id_3 de classe 3 a une valeur unique.*
Elt2: *attribut Id_4 de classe 4 a une valeur unique.*
Elt3: *la classe 4 possède attribut C4.*
Elt4: *la classe 4 possède association1n avec la classe 3.*

Decoder. In this step, the decoder uses the parameters of IBM1 model[1] to find the best translation of source text from bilingual and monolingual corpora. The calculated parameters are shown in table 1 and 2. Once previous tables (1 and 2) are created, the decoder can generate the target sentences from source sentences based on the following equation.

$$\hat{e} = P(f_1^J, \hat{a}_1^J | f_1^I) P(e_1^I) = \prod_{j=1}^{J} \hat{t}(f_j | e_{aj}) \prod_{i=1}^{I} P(e_i | e_{i-1}). \quad (7)$$

IBM1 model is not based on the alignment where $P(f_1^J, \hat{a}_1^J | f_1^I) = \hat{t}(f_j | e_{aj})$ that explains the best translation value which is selected from the previous table.

Sentence and Model Generation. Once the decoding phase is done, the extraction of target sentences can be executed based on this calculated probability $argmax P(f_1^J | e_1^I) P(e_1^I)$. The output sentences of the input sentences (see Figure 6 and its definition in natural language in this sub-section) are defined as follows:
Translation of Elt1: *column Id_3 of table 3 is a primary key.*
Translation of Elt2: *column Id_4 of table 4 is a primary key.*
Translation of Elt3: *the table 4 has column C4.*
Translation of Elt4: *the table 3 has foreign-key that-is-defined-as-primary-key-in the table 3.* To have useful MDE artifacts, it is necessary to map the sentences generated by SMT system into models conform to relational schema metamodel. In our case, we create the models manually from their correspondence sentences.

[1]https://www.nltk.org/_modules/nltk/translate/ibm1.html

5 EXPERIMENTATION AND DISCUSSION

The objective of this section is to evaluate the proposed approach with six transformation examples from various domains. For each transformation we collected/defined a set of pairs of model examples. The models were translated into natural language texts. Table 4 shows the vocabulary volume (sentences and words) for parallel corpus of each transformation example (in our case, the monolingual corpus size generally presents the half (1/2) of parallel corpus for each example).

Table 4: Information about used transformation examples.

Transformation Examples		Parallel Corpus	
Example	NO_{Exples}	NO_{Sents}	NO_{Words}
UMLCD2RS	10	154	1073
Book2Publication	10	66	688
Family2Person	10	88	1016
Ecore2Coq	10	392	3071
AADL2TA	05	118	1217
UMLSM2PetriNet	15	388	3899

We evaluate the precision and recall metrics as adapted to model transformation in (Hammoudi et al., 2014). Both metrics are calculated in terms of words found by the automated SMT translation (W_{auto}) w.r.t. the words expected (W_{expert}), i.e, defined by the expert who produced the model example pair. We define the set $W_{positive}$=$W_{auto} \cap W_{expert}$, as the set of words that are obtained by SMT translation and are also included in the expert's words.

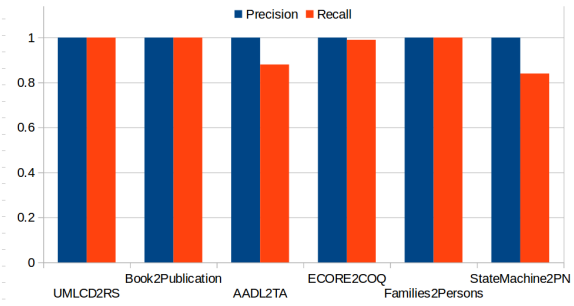

Figure 7: Precision and Recall of transformation examples.

In this context, we applied the trained translation model on the examples used for training. Figure 7 summarizes the results obtained for the precision and recall. Our SMT system based on IBM1 model gives perfect precision 100% for all transformation examples. For the recall our SMT system exhibits also perfect values 100% for all transformation examples except for AADL2TA, which are considered as a very less score in our SMT system (these values are defined between 88% and 93%).

These values reflect the difficulty of fully handling transformation with complex semantic gaps between source and target models such as AADL2TA example, but with well structured semantics such as UMLCD2RS transformation we have obtained the best values.

In our case, we generate the target models (in our case, from target sentences are created the target models) without using a set of transformation rules but only with a set of source and target model pairs. This latter is used to specify the parallel and monolingual corpora for our SMT system. In this case, we have processed the following problems: ***RQ3*** and ***RQ4*** that are presented in section 2 have been solved by specifying the models in natural language using manual or automatic preparation).

In model transformation context, all proposed works are based on optimization technique or matching tool in their main step, however,our approach aims to ease the transformation by treating the challenge of ***RQ2*** by using exactly SMT system to generate the output models without using a specific program.

Observation 2: Using NL in MDE Context

The use of natural language facilitates metamodeling and model transformation processes. These latter are done automatically without manipulating specific languages or tools.

Our approach makes it possible to solve the problem ***RQ1*** focusing on learning system based on parallel and monolingual corpora using IBM1 model. In this case, parallel and monolingual corpora are proposed as a definition of transformation rules and the execution of IBM1 is defined as the executable of transformation rules.

6 RELATED WORK

The model transformation challenge is still how to automate the model transformation process. Several studies suggested to interpret this challenge into two principal classes according to the type of input element: the first class is based on metamodels and the second one manipulates models to automate or semi-automate transformation process.

First class is focused on the detection of relationships between the input and the output metamodels automatically. It is important to notify that some works are focused on specifying semi-automatically a transformation rules by using matching techniques. For instance, this work (Hammoudi et al., 2014) describes the use of various matching tools and techniques to automate model transformations but these matching techniques are based also on metamodel pretreatment and require understanding the basic information about the ontologies. Another example that illustrates this idea is Berramla's paper (Berramla et al., 2017b) that is focused on matching algorithm but it compares all metamodel elements without considering their types and uses metamodels as input data without changing their structure. Both abstract and concrete transformation rules are generated from simple mapping elements, however, the intervention of an expert is obligatory for complex transformation examples to validate their generated mappings.

Second class is founded on the use of models as input elements to automate model transformation. This category contains also several proposed works using different techniques. In what follows, we briefly discuss on these prpoposed works. Some works are based on MTBE using ad-hoc algorithms. One among the first researches in model transformation by example is the Varró 's work (Varró, 2006) that uses as input elements a set of source and target model pairs and the prototype mapping between each model pair in order to help the transformation designer the creation of model transformation semi-automatically. Other studies use artificial intelligence (AI) techniques. In (Baki and Sahraoui, 2016) is based on genetic programming to learn operational rules from source and target model pairs and a set of transformation rules by adding also the control of each rule during the their generation. In (Kessentini et al., 2012) describe a new approach to define how to integrate Particle Swarm Optimization (PSO) and Simulated Annealing (SA) in order to generate the output models automatically without using a transformation program. The rest of these studies integrate the mathematical methods to define the model transformation by examples. For instance, in (Berramla et al., 2017a) a new approach is proposed defining the hybridization between the model-matching and the formal concept analysis. The first one is used to create the correspondences from

[2]Transformation rules written in abstract level(or and in concrete levels).

source and target models and the second one is applied to reduce the redundancy of transformation rule creation.

From the table 3, we note that all proposed approaches are based on metamodeling tools or languages to have a best representation of their input elements which require a little times and efforts to interpret their input elements but with our approach, we use only the natural language to describe the models to be translated.

The main challenge of the most proposed works in this context is how to create transformation rules semi-automatically. This paper is focused on how to translate easily a model to another model without using tools or languages in both metamodeling and transformation levels.

7 CONCLUSION AND FUTURE WORK

During recent years, model transformation by example has seen in several works such as (Baki and Sahraoui, 2016; Varró, 2006) to define the model transformation in a semi-automatic way but applying the proposed techniques are limited to simple transformation examples. In this paper, we proposed an approach to define the model transformation automatically using only a set of models and SMT system more specifically IBM1 model in order to reduce the costs and the time of software development.

The most important objective of this work is not only to automate transformation process but also to facilitate the modeling phase by using natural language without basing on specific tools or/and languages that require good knowledge about them. Once the modeling phase is executed, a set of sentences written in two languages are builded from the source and the target model pairs. From these sentences, is established a parallel corpus which is useful in the training phase to calculate a set of parameters that has been also used in the test phase in order to obtain a good translation with this system. Finally the translated sentences permit to create automatically the target models through the use of their metamodels. This process changes from one transformation example to an another according to the structure of target metamodel.

Improvements can be planned as perspectives for this work. First of all, to propose a generalization of the transformation process that will make it possible to translate any model into a set of sentences written in natural language. Also, we can use other translation system such as phrase-based translation system.

ACKNOWLEDGEMENTS

This work has been funded in part by the europian project PRIMA WaterMed 4.0, "Efficient use and management of conventional and non-conventional water resources through smart technologies applied to improve the quality and safety of Mediterranean agriculture in semi-arid areas" and by MESRS, "Ministère de l'enseignement supèrieur et de la recherche scientifique".

REFERENCES

Al-Ghawanmeh, F. and Smaïli, K. (2017). Statistical machine translation from arab vocal improvisation to instrumental melodic accompaniment. *Journal of the International Science and General Applications*, 1(1):11–17.

Baki, I. and Sahraoui, H. (2016). Multi-step learning and adaptive search for learning complex model transformations from examples. *ACM Trans. Softw. Eng. Methodol.*, 25(3):20:1–20:37.

Berramla, K., Deba, E. A., Benyamina, A., Touam, R., Brahimi, Y., and Benhamamouch, D. (2017a). Formal concept analysis for specification of model transformations. In *Embedded & Distributed Systems (EDiS)*, pages 1–6. IEEE.

Berramla, K., Deba, E. A., Benyamina, A. E. H., and Benhamamouch, D. (2017b). A Contribution to the Specification of Model Transformations with Metamodel Matching Approach. *IJISMD*, 8(3):1–23.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Brunning, J. J. J. (2010). *Alignment models and algorithms for statistical machine translation*. PhD thesis, University of Cambridge.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Hammoudi, S., Feki, J., and Lafi, L. (2014). Metamodel matching techniques: Review, comparison and evaluation. *International Journal of Information System Modeling and Design*, pages 1–27.

Kessentini, M., Sahraoui, H., Boukadoum, M., and Omar, O. B. (2012). Search-based model transformation by example. *Software & Systems Modeling*, 11(2):209–226.

Kleppe, A. G., Warmer, J., Warmer, J. B., and Bast, W. (2003). *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Professional.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Varró, D. (2006). Model transformation by example. In *International Conference on Model Driven Engineering Languages and Systems*, pages 410–424. Springer.