



A Systematic Model to Model Transformation for Knowledge-Based Planning Generation Problems

Liwen Zhang, Franck Fontanili, Elyes Lamine, Christophe Bortolaso,
Mustapha Derras, Hervé Pingaud

► To cite this version:

Liwen Zhang, Franck Fontanili, Elyes Lamine, Christophe Bortolaso, Mustapha Derras, et al.. A Systematic Model to Model Transformation for Knowledge-Based Planning Generation Problems. IEA/AIE 2020 - The 33th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, Sep 2020, Kitakyushu, Japan. pp.140-152, 10.1007/978-3-030-55789-8_13. hal-02935179

HAL Id: hal-02935179

<https://imt-mines-albi.hal.science/hal-02935179>

Submitted on 10 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Systematic Model to Model Transformation for Knowledge-Based Planning Generation Problems

Liwen Zhang^{1,2}, Franck Fontanili², Elyes Lamine², Christophe Bortolaso¹, Mustapha Derras¹, Hervé Pingaud³

¹ Berger-Levrault, Jean Rostand, 64, 31670 Labège, France
{first name.last name}@berger-levrault.com

² CGI, University of Toulouse-IMT Mines Albi, Campus Jarlard, 81000 Albi, France
{first name.last name}@mines-albi.fr

³ CNRS LGC, University of Toulouse-INU Champollion, Place de Verdun, 81012 Albi, France
herve.pingaud@univ-jfc.fr

Abstract. The generation of an optimum planning problem and search for its solution are very complex regarding different business contexts. Generally, the problem is addressed by an optimization formulation and an optimizer is used to find a solution. This is a classical approach for the Operations Research (OR) community. However, business experts often need to express specific requirements and planning goals mathematically on a case by case basis. They also need to compute a planning result within various business constraints. In this paper, we try to support these experts during this preliminary problem design phase using a model driven engineering framework. An OR model could be generated from the knowledge included in a business conceptual model. A model to model transformation is described to support this goal. The Traveling Salesman Problem is used as a simple case study that allows explanation of our model transformation rules. We implemented our approach in the ADOxx meta-modelling Platform.

Keywords: Planning problem, Conceptual modeling, Operations Research, Model Driven Engineering, Model to Model transformation, ADOxx, CPLEX

1 Introduction

Currently, in our connected world, there is a very wide demand from many organizations to regularly route and schedule people and goods. However, depending on the management culture of a given organization, planning will not be approached with the same spirit. A decision maker will certainly appreciate using a customized Human Machine Interface (HMI) in an efficient and comprehensive manner to obtain correct business-oriented knowledge to implement its own target planning needs.

Knowing this context, under the assumption of varying demands and specificities of service deliveries of each organization, it is necessary to embrace a broad set of planning problems including a variety of mathematical formulations. An adapted mathematical formulation is a preliminary condition to call an optimizer with an em-

bedded mathematical modelling language (i.e. CPLEX, Gurobi, LocalSolver), that will compute a solution with respect to an objective function and a set of constraints. For sake of efficiency as well as agility, a decision maker will appreciate the use of a strong decision-making support system in the complex combination of business constraints with which he must operate. Furthermore, the intrinsic nature of the demands for planning within a special application area (i.e. healthcare management, supply chain management, cooperation management), will tend to evolve considering the better control of system states obtained from the intensive use of information and communication facilities. This will lead to new pressure on decision makers as they face more complex situations and strive to more frequently formulate effective responses. It will pave the way to achieve business objectives while respecting the constraints inherent to the reality of an operational field that is ever changing. However, we think there is a lack of a strong decision support system which better captures knowledge using models about the system under study, whatever the specificity of the planning problem.

Let us illustrate this, making reference to the traditional planning optimization process that is often addressed as a mathematical problem in optimizing the planning of staff. The subject has been widely studied by the Operations Research (OR) community. It is often titled in the literature under the heading VRP (Vehicle Routing Problem) [1]. The mathematical formulation is two part, either mono-criteria or multi-criteria. In addition, the planning model can also be derived from a large number of variants depending on the different business constraints to be satisfied (i.e. means of transportation, geographical features, user preferences). As an illustration, a rise of activity on the operational field will inevitably reveal new characteristics to be taken into account and enforce a revision of the problem to be addressed. Facing this diversity in terms of operational planning model, the decision maker in the organization is not always able to react and change the model expeditiously. Decision makers do not necessarily have the expertise to consider the best or impact of the revised planning model. They are, however, business experts with sufficient knowledge to master business-oriented processes and content in order to apply them in the required business application. So, we must master a transition from their knowledge and business expertise to an ability to perform planning generations. We translate this need into a rationale of interactions between a real business-oriented model on the one hand, and the OR decision-making support model on the other. To do so, we use the best available knowledge management methods as part of a model driven engineering (MDE) approach. Our research aims to build a decision support system platform for helping these businessmen make ad-hoc planning decisions individually.

In this paper, based on a very simplified VRP problem, the Travelling Salesman Problem (TSP) [2], we show how the whole model transformations are put into practice following a three step generation process. i) Two models are considered at the two parts of the transformation. A given business-oriented conceptual model is a starting point, the source of transformation into a graph-based OR model that is a target. The knowledge transformation has been performed within the ADOxx meta-modeling platform developed by the OMILAB community [3]. ii) The graph-based OR model will provide the basis to parameterize a mathematical model predefined in the optimizer CPLEX. iii) CPLEX will compute the TSP result.

The following sections are divided into 6 parts. The overview of our approach is il-

illustrated in section 2. In the section 3, we review the related work on MDE as well as MTL (Model Transformation Language) which will support an automatic model to model transformation. The section 4 details the two meta-models of step i) of our generation process, as well as the transformation rules. Next, we explain the other remaining steps using the simple TSP case study. We make the assumption here that these final steps are not at the center of our concerns and we have effectively simplified our approach in consequence. The last section gives some conclusions and draws perspectives for future progress.

2 Approach based on Model Driven Engineering (MDE)

Our MDE decision support platform is based on a series of three modules forming a consistent architecture. Together, they will deliver a generative solution for the TSP. **Fig. 1** shows an overview of the linear process that goes through the three modules. We firstly focus on the interaction between the business-oriented conceptual model (BCM) and the graph-based operation research model (GORM). Then, the GORM built from the BCM allow us to parameterize a predefined TSP solution-oriented mathematical model written in the OPL language and by doing so, prepare it to be submitted to the IBM CPLEX optimizer to generate a result.

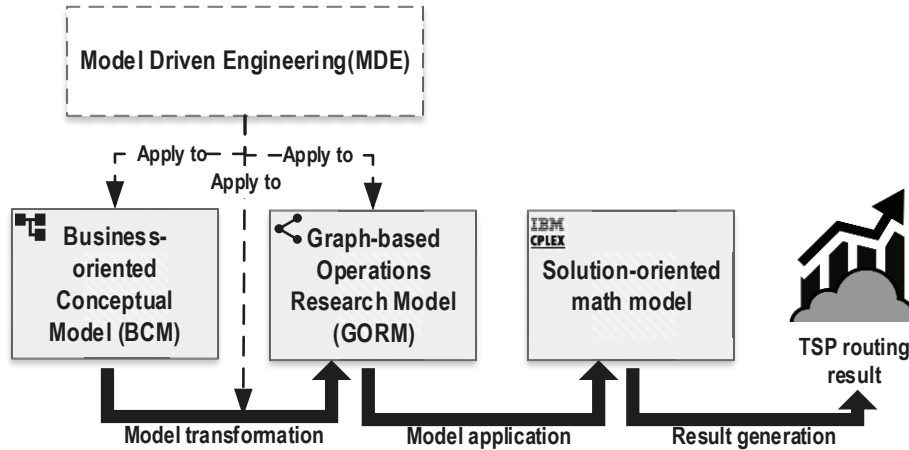


Fig. 1. Overview of our approach based on Model Driven Engineering

The first module (BCM) consists of embedding a domain specific modeling language (DSML) for defining the key concepts in the TSP, based on a business-oriented vocabulary. This conceptual modeling [4] uses a syntax that is meaningful for the business expert and shall provide him the means to easily specify the requirements of her/his TSP. It allows us to give a clear and specific definition of :

- Human support in the analysis of organizational and technical relationships.
- Creation of models from graphs or diagrams.
- Facultative definition of formal semantics.
- Improvement of human understanding and model mechanical processing.

BCM allows the user to characterize the knowledge required in the TSP formulation phase, and to identify the concepts associated with it such as the objective function. It could be the minimization of total salesman travel distances.

In the second module (GORM), as mentioned previously, OR knowledge is made available in order to translate business requirements into a mathematical formulation [5]. For the TSP case study, it will mainly consist in setting up a graph based representation for which the nodes and the edges will be symbols associated with variables and associations between them [6], matching ultimately the TSP's mathematical formulation that IBM CPLEX OPL model is waiting for.

The set of 2 first modules structure the model-driven engineering approach [7]. For MDE, models are considered as primary artifacts flowing throughout an engineering life cycle, during which it ensures model conformity to languages as well as ensuring model to model transformation (M to M). In order to do so, the keys required are a source meta-model and a target meta-model, which specify the concepts, and therefore the primitives of the languages related to their modelling. **Fig. 2** shows the relationship between the real world, the model and the meta-model. The M to M transformation is realized by respecting the mapping rules defined in the two meta-models. In our case, BCM is as the source model and conforms to a DSML meta-model. The GORM is the target model and conforms to the OR graph language meta-model. In the following, we will call them respectively BCMM and GORMM.

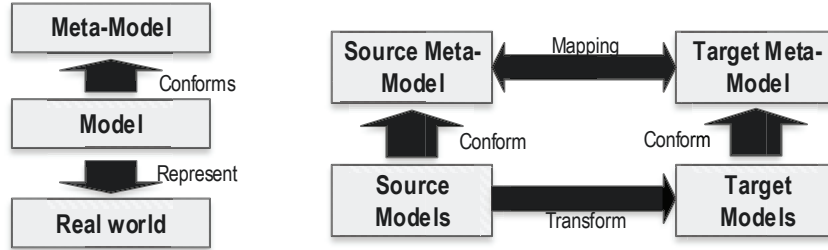


Fig. 2. Relationships between real world, model, meta-model and meta-models based M to M transformation adapted from Wang et al.[8]

When the GORM is derived correctly from the source BCM, all the necessary parameters have been mandatorily translated into mathematical elements in the graph model.

3 Related work

In recent years, model transformation methods have been extensively studied. Czarnecki, K., Helsen, S [9] gave us a clear view about the classification of general existing approaches in model transformation. For a simple use case based on UML/MOF class diagram [10] in conjunction with OCL constraint [11], the model transformation is addressed by an ordinary and basic model, from the entity-relationship model to the relational model. Moreover, as model validation is an essential phase after model transformation, a specific constraint and query language (i.e. OCL) is needed to control the distinct properties during the transformation [12]. In terms of application in

the industry, interoperability problems linked to collaboration network configurations have been largely addressed. For example, we cite the mediation information system engineering (MISE) approach based on MDE to design a mediation information system (MIS), with the aim of improving interoperability in emerging collaborative situations [13]. Before achieving this goal, the design of the collaboration network could also be based on MDE using added knowledge given by use of ontologies [14]. Facing the improvement of the cooperation capabilities, model-to-model mapping is applied to detect the meanings and relations between different core business enterprise models automatically [8]. For another applicative domain such as healthcare system, a prototype based on MDE is introduced for realizing the mobile device interface code generation from the visual care plan modeling [15]. In regard to our work, some approaches are presented to perform the graph transformation through a simple visual model [16]. However, we have not found related research referring to the application of MDE for model transformation between a business context model and graph-OR model.

As mentioned above, the main ingredient for supporting model transformation is Model Transformation Language (MTL). ATL is largely used in the related research works [17]. Furthermore, the recent research of Burgueño, L et al. shows a survey about the MTL in practice [18], as well as the current and future state of MTL against the evolution of artificial intelligence. One objective of our work is to describe the transformation process from BCM to GORM. We will do it not only in the backend of our support system, as a black box, but also in the frontend as an HMI facility during our experiment in order to display our source and target model during operation. For fulfilling these two goals, we implement our framework through ADOxx platform embedded in an MTL named AdoScript, which is largely used to build a Proof of Concept (PoC) in OMILAB community [3].

4 Meta-Models specification

4.1 Meta-Model: BCMM

The knowledge (business objects and their relationships) conveyed by TSP must be based on a meta-model whose content will set requirements for structuring data. The main concepts of a TSP consist of:

- **Salesman**: the person who will execute the city tour, characterized by her/his *name*.
- **City**: the town entity visited by the salesman, characterized by an *identifier (id)*.
- **Travelling**: the routing itinerary between each pair of cities, characterized by distance length in kilometers. The distance of a two-way journey for a pair of cities might be different.

In terms of business constraints (**BuConst**), the constraint concept must guarantee that the salesman visits each city only once and finishes the tour in the starting city (the journey is a cycle). An enumeration list characterizes this concept within the “*type*” attribute, which includes the three types of constraints assimilated to options:

- Each city reached from exactly one other city by the salesman.
- From each city, there is only a departure to another city.
- There is only one tour covering all the cities to be visited.

Finally, the **Goal** concept is introduced to specify our objective function, the minimization of the salesman's total routing distance during the chosen visits. It is also explained by a "type" attribute. **Fig. 3** illustrates the BCMM, the goal is to provide the user with a platform for collecting the necessary data in a user-friendly way, though conceptual knowledge modeling covering all dimensions of the TSP.

4.2 Meta-Model: GORMM

In OR, TSP is regarded as a typical combinatorial optimization problem in NP-optimization. This problem has always been represented using a directed-graph, with the form $G = (V, E)$ where V is the set of nodes and E is the set of edges [6]. Moreover, as the meta-model for the graph structure has yet to be addressed by Enriquez et al. [19], we extend this proposal to fit our problem. The explicit concepts in our GORMM are the followings :

- **CityNode**: the city node inherited the meta-class **Node** that makes up a graph. A graph can contain zero or more city nodes, and a city node is part of a graph. **CityNode** characterized by *name* and *identifier (id)*.
- **RoutingEdge**: the routing edge inherited the meta-class **Edge** of a graph. A graph can contain zero or more routing edges and a routing edge is part of a graph. Two types of edge exist: income or outcome edge in respect of each city node, due to the intrinsic directed-graph structure. A **RoutingEdge** is characterized by its *name*, *start node id* (i.e. 3), *end node id* (i.e. 1), *start-end tuple* (i.e. <3,1>) and *edge length* associated to a distance in a given unit (i.e. 8). The *start-end tuple* attribute must fit with the required input in the CPLEX environment.
- **CityNodeSet**: the set of city nodes in a graph. This is the key concept in the combinatorial optimization, because the decision variable will be formally designated in the field of concept attributes, which will be defined in accordance with the constraint or objective functions. The set of nodes is a class whose attributes will become parameters (as opposed to variables) of the mathematical model afterwards (i.e. number of cities). The **CityNodeSet** is characterized by its *name*, and consequently by the *total number of city node*.
- **RoutingEdgeSet**: the set of routing edges in a graph. Similar rules are applied to this concept as for the CityNodeSet, including the decision variables definition and parameters integration. The **RoutingEdgeSet** is characterized by its *name*, the *total edge tuples* and the *distance matrix*.

Besides these mains graph-oriented concepts, we have to indicate the content of the constraints (**Const**) and the Objective Function (**OF**) in our GORMM. Meanwhile these contents are related to the decision variable gathered in one of the two "Set" mentioned before. For example, the constraint named "No subtour" is linked to the binary decision variable named "*dvarNoSubTour*" built in **CityNodeSet** concept. **Table 1** is an overview about the relevant mechanisms to set the stage of the TSP.

Fig. 3 illustrates the GORMM. The goal of this meta-model is to represent the graph-oriented knowledge in 3 aspects:

- Visualization of the structuring graph-oriented knowledge.
- Generation of the input data for our mathematical IBM CPLEX optimizer.
- Definition of the decision variable based on the constraint content.

Table 1. overview about the constraints' content and the related decision variable under TSP

| Constraint content | Constraint explication in business context | Decision variable type | Decision variable nomination |
|--------------------|---|------------------------|------------------------------|
| Flow in | Each city is arrived at from exactly one other city | Boolean | dvarAssignmentnt |
| Flow out | From each city there is a departure to exactly one other city | | |
| No subtour | Only one tour covering all the cities to be visited covering all cities | Positive integer | dvarNoSubTour |

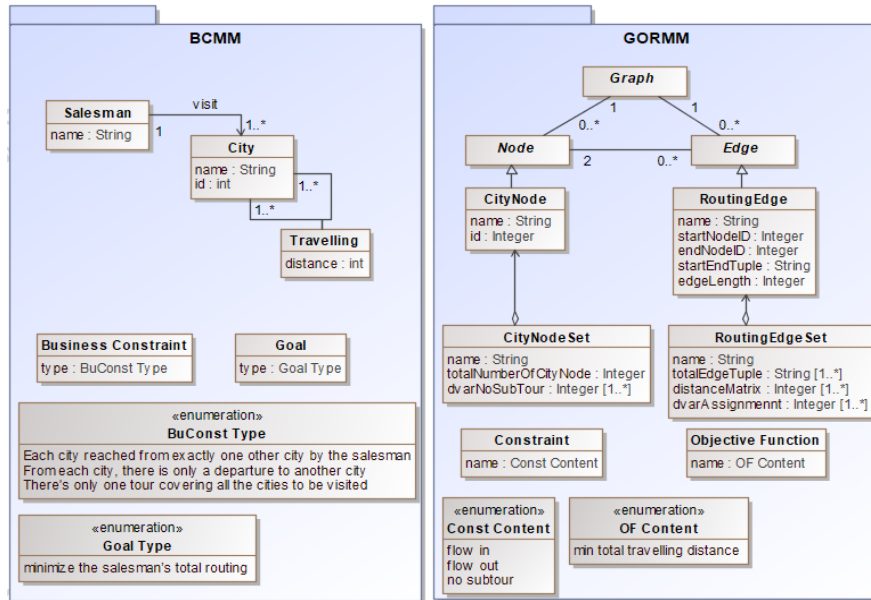


Fig. 3. Illustration of BCMM and GORMM

5 Implementation

In this section, we introduce the methodology for processing the Model to Model transformation from BCM (source model) to GORM (target model).

The translation from one model to the other is divided into two main parts: class (concept) mapping and attribute mapping, which are performed simultaneously. The transformation process operates progressively and can be described in 4 steps. **Fig. 4** shows what happens at the model level on both sides subject to inference of the class

mapping rules at the meta level.

1. At first, we execute the mapping from **City** to **CityNode**, the corresponding attribute value such as identifier in city class will be assigned directly in the identification of **CityNode**.
2. The second step pertains to city interrelation building. The **Travelling** class (an arrow represents the relation in BCM) specifies the distance between each pair of cities. In the GORM, the relation class will be endowed a new pattern [**flow in** (relation) – **RoutingEdge** (class) – **flow out** (relation)]. This is for explaining the constraint named “flow in” and “flow out” regarding each **CityNode** in the result generation phase. As for the attributes, we focus not only on distance, but also on the identification of the incoming and outgoing **CityNode**, referring to each routing edge.
3. Next, the business constraints and goal are translated. Through this operation, on the mathematical side, we provide an indicator to indicate the relative equations in mathematic model.
4. Finally, the **Set** class generation is done. It is not a translation, but it depends on the result of steps 1 and 2. The purpose is to determine the defined set parameter value, which will be the input data of our IBM CPLEX model. To progress, the operation must consist of managing data in a new structure gathering together the attribute value in homogeneous classes, such as the whole set of distances defined in a routing edge matrix. This is realized by “browsing” functions which collect a singular value (i.e. a distance) as input and store it in a collection in output (i.e. distance matrix).

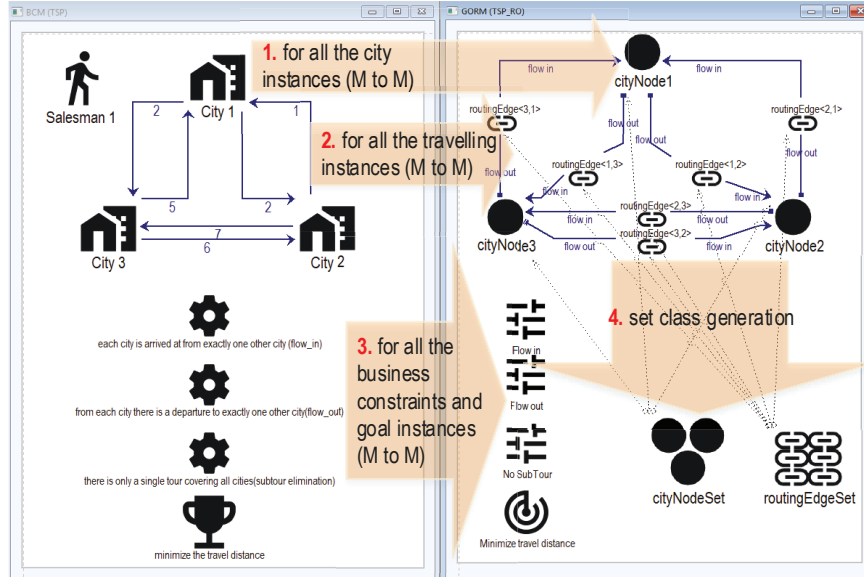


Fig. 4. M to M transformation steps in model level by one TSP Use Case: 3 cities to be visited

Table 2 details the mapping rules applied to support the model transformation process, based on AdoScript MTL.

Table 2. Mapping rules between BCMM and GORMM

| Source | | Mapping rules base on AdoScript | Target | |
|------------|-----------------|--|-----------------------|----------------|
| Concept | Attribute | | Attribute | Concept |
| City | <i>idOfCity</i> | For srObj in srObjs { CC "Core" getClass (City) CC "Core" getAttribut (srObj)of(City) SET aVal = " <i>idOfCity</i> " (srObj)of(City) CC "Core" getClass (CityNode) CC "Core" createObj (tarObj=srObj)of(CityNode) CC "Core" set(aVal)in" <i>id</i> " (tarObj)of(CityNode) } | <i>id</i> | CityNode |
| City | <i>idOfCity</i> | For srObj in srObjs { SET conns = getConnectors(srObj)of(City) SET aValStart = " <i>idOfCity</i> " (srObj)of(City) For conn in conns { CC "Core" getAttribut (conn)of(Travelling) SET aValDist= " <i>distance</i> " (conn)of(Travelling) SET srEndP = getEndPoint(conn) CC "Core" getAttribut (srEndP)of(City) SET aValEnd = " <i>idOfCity</i> " (srEndP)of(City) CC "Core" getClass (RoutingEdge) CC "Core" createObj (rTarObj)of(RoutingEdge) CC "Core" createConnector From (tarObj= srObj) To (rTarObj) CC "Core" createConnector From (rTarObj) To (tarEndP =srEndP) CC "Core" set(aValStart)in" <i>startNodeId</i> " (rTarObj)of(RoutingEdge) CC "Core" set(aValEnd)in" <i>endNodeId</i> " (rTarObj)of(RoutingEdge) CC "Core" set(<aValStart ,aValEnd>)in" <i>startEndTuple</i> " (rTarObj)of(RoutingEdge) CC "Core" set(aValDist)in" <i>edgeLength</i> " (rTarObj)of(RoutingEdge) } } | <i>startNodeId</i> | RoutingEdge |
| | | | <i>endNodeId</i> | |
| | | | <i>startEndTuple</i> | |
| | | | <i>edgeLength</i> | |
| Travelling | <i>distance</i> | | | |
| BuConst | <i>type</i> | For srObj in srObjs { CC "Core" getClass (BuConst) CC "Core" getAttribut (srObj)of(BuConst) SET aValConst = " <i>type</i> " (srObj)of(BuConst) CC "Core" getClass (Goal) CC "Core" getAttribut (srObj)of(Goal) SET aValGoal = " <i>type</i> " (srObj)of(Goal) CC "Core" getClass (Const) CC "Core" createObj (tarObj=srObj)of(Const) CC "Core" set(aValConst)in" <i>name</i> " (tarObj)of(Const) CC "Core" getClass (OF) CC "Core" createObj (tarObj=srObj)of(OF) CC "Core" set(aValGoal)in" <i>name</i> " (tarObj)of(OF) } | <i>name</i> | Const |
| Goal | <i>type</i> | | <i>name</i> | OF |
| | | CC "Core" getClass (CityNode) CC "Core" getAllObjs(tarObjs)of(CityNode) SET aNum = countTheNumber. (tarObjs)of(CityNode) CC "Core" createObj(tarObj)of(CityNodeSet) CC "Core" set(aNum)in" <i>numOfCityNode</i> " (tarObj)of(CityNodeSet) | <i>numOfCityNode</i> | CityNodeSet |
| | | CC "Core" getClass (RoutingEdge) CC "Core" getAllObjs(tarObjs)of(RoutingEdge) SET aEdgeTuple = append." <i>startEndTuple</i> " (tarObjs)of(RoutingEdge) SET aDistMat = append." <i>edgeLength</i> " (tarObjs)of(RoutingEdge) CC "Core" createObj(tarObj)of(RoutingEdgeSet) CC "Core" set(aEdgeTuple)in" <i>edgeTuple</i> " (tarObj)of(RoutingEdgeSet) CC "Core" set(aDistMat)in" <i>distanceMatrix</i> " (tarObj)of(RoutingEdgeSet) | <i>edgeTuple</i> | RoutingEdgeSet |
| | | | <i>distanceMatrix</i> | |

5.1 Application in IBM CPLEX for TSP result generation

Once the GORM is generated by applying the explicit rules enumerated in Table 2, this GORM provides an architecture that allows us to manipulate a predefined mathematic formulation in the CPLEX environment, adapted to the TSP being addressed. A CPLEX oriented mathematical model is composed of two parts : model (.mod) and

data (.dat). The model component is embedded a modeling language Optimization Programming Language (OPL) [20] that allows to compute linear (or quadratic) programs, through an expression closed to the mathematical formulation among the literature of OR community. OPL is promoting well-defined syntaxes and operators within CPLEX environment, to express the combinatorial optimization problem (i.e. TSP in our case) mathematically by specifying relevant data (parameter and decision variable), equation constraints and OF, and then applying an exact algorithm (i.e. branch and bound [21]) to resolve such a complex problem. The data component is aimed to provide the initial values to the structuring parameters which are defined in the model component.

Having in mind this architecture of the CPLEX math model, the GORM serves as a source for a new translation going to the TSP expression into a mathematical optimization model. **Fig. 5** shows the overview of the roadmap concerning the application of GORM in CPLEX model. With respect to this figure, the GORM is processing the following knowledge :

1. Indication of decision variable. The attribute with prefix “dvar” of “set” object in GORM indicates the needed decision variables in respect of TSP.
2. Indication of data declaration. Some objects’ attributes in GORM can be considered as a pointer, in order to identify the type’s declaration of relative data in CPLEX math model.
3. Input data generation. After declaring data types, this stage allows us to assign the value to the related parameter with a specific format: parameter name – value in data part (.dat) of math model in CPLEX.
4. Constraint and objective indication. The value of “name” attribute in constraint and OF object in GORM, draws a link to the respective equation in the CPLEX model. This mapping enables the computation module to process the algorithm and find the optimum value of the indicated decision variables.
5. The last part is a configuration for providing the decision-making support to the user, by properly returning the results from the optimizer, ideally in the BCM. These values announce the best routing result for TSP regarding the conceptual modelling. We emphasize the flexibility that comes with the modification of the TSP’s dimension, such as the increase of the number of cities to be visited on the modelling component of BCM.

6 Conclusions and perspectives

This research work addresses a framework to solve the classic combinatorial optimization problem of TSP based on MDE. We emphasized the M to M transformation between BCM and GORM by adopting the MTL AdoScript as a component of the ADOxx platform for supporting this transformation, as well as the visualization of the two models. Furthermore, we specified the knowledge to be handled during the main steps for processing the wanted transformation: 2 meta-models and the explicit mapping rules between meta-models at an upper level. During the framework design and experimentation, we decided the final target model to be an IBM CPLEX solution-oriented one, which facilitates the generation of a TSP solution for a given set of data.

We are not sure at this time we have reached the best transformation model level,

so we cannot affirm that we have captured all the existing knowledge about how the experts resolve the planning problem. But we think that our MDE framework could be a basis for the continuous improvement in the future on that point, which leads to a better optimized model to model transformation

In the future, at first, it will be interesting to make progress at the end of the MDE chain. We intend to develop a meta-model adapted to OPL in CPLEX, for supporting the M to M transformation from GORM to this math model in CPLEX. Then we wish to extend this approach not only for TSP, but also to other Vehicle Routing Problems (VRP), and test its added value for a decision maker facing more and more complex OR problems. Starting from this prototype, the following question is to demonstrate how we could handle a broad variety of planning problems with less effort. Considering the TSP experimentation, for example, we work on the assessment of our tool's effectiveness when many problem configurations are considered. It means that the OF or some constraints have to be systematically modified through model transformation, subject to a change in specifications made at the business level. This is the price to be paid for quality proof of our innovation.

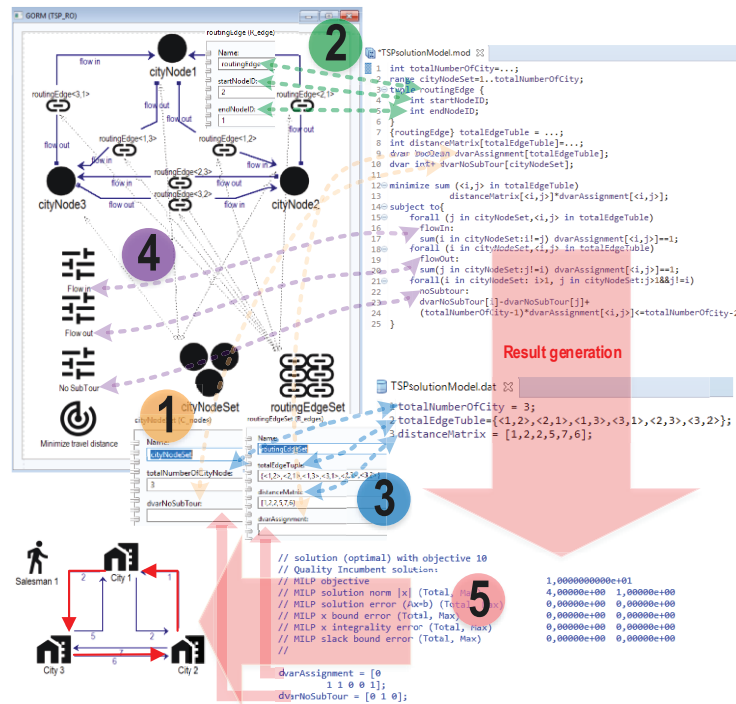


Fig. 5. Overview of the roadmap concerning the application of GORM in CPLEX model

References

1. Toth, P., Vigo, D.: The vehicle routing problem. SIAM (2002).
2. Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem. Management Science. 6, 80–91 (1959).

3. Fill, H.-G., Karagiannis, D.: On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*. 8, 4–25 (2013). <https://doi.org/10.18417/emisa.8.1.1>.
4. Fill, H.-G.: Semantic Evaluation of Business Processes Using SeMFIS. In: Karagiannis, D., Mayr, H.C., and Mylopoulos, J. (eds.) *Domain-Specific Conceptual Modeling: Concepts, Methods and Tools*. pp. 149–170. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-39417-6_7.
5. Hastings, K.J.: *Introduction to the mathematics of operations research with mathematica®*. CRC Press (2018).
6. Bondy, J.A., Murty, U.S.R.: *Graph theory with applications*. (1976).
7. MOF, O.: *OMG Meta Object Facility (MOF) Specification v1. 4*. OMG Document formal/02-04-03 [Online]. Available: <http://www.omg.org/cgi-bin/apps/doc> (2002).
8. Wang, T., Truptil, S., Benaben, F.: An automatic model-to-model mapping and transformation methodology to serve model-based systems engineering. *Inf Syst E-Bus Manage*. 15, 323–376 (2017). <https://doi.org/10.1007/s10257-016-0321-z>.
9. Czarnecki, K., Helsen, S.: Classification of model transformation approaches. In: *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*. pp. 1–17. USA (2003).
10. Loecher, S., Ocke, S.: A Metamodel-Based OCL-Compiler for UML and MOF. *Electronic Notes in Theoretical Computer Science*. 102, 43–61 (2004). <https://doi.org/10.1016/j.entcs.2003.09.003>.
11. Bézivin, J., Büttner, F., Gogolla, M., Jouault, F., Kurtev, I., Lindow, A.: Model Transformations? Transformation Models! In: Nierstrasz, O., Whittle, J., Harel, D., and Reggio, G. (eds.) *Model Driven Engineering Languages and Systems*. pp. 440–453. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11880240_31.
12. Lengyel, L., Levendovszky, T., Mezei, G., Forstner, B., Charaf, H.: Metamodel-Based Model Transformation with Aspect-Oriented Constraints. *Electronic Notes in Theoretical Computer Science*. 152, 111–123 (2006). <https://doi.org/10.1016/j.entcs.2005.10.020>.
13. Mu, W., Benaben, F., Boissel-Dallier, N., Pingaud, H.: Collaborative Knowledge Framework for Mediation Information System Engineering. *Scientific Programming*. 2017, 1–18 (2017). <https://doi.org/10.1155/2017/9026387>.
14. Wang, T., Montarnal, A., Truptil, S., Benaben, F., Luras, M., Lamothe, J.: A Semantic-checking based Model-driven Approach to Serve Multi-organization Collaboration. *Procedia Computer Science*. 126, 136–145 (2018). <https://doi.org/10.1016/j.procs.2018.07.217>.
15. Khambati, A., Grundy, J., Warren, J., Hosking, J.: Model-Driven Development of Mobile Personal Health Care Applications. In: *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering*. pp. 467–470. IEEE Computer Society, Washington, DC, USA (2008). <https://doi.org/10.1109/ASE.2008.75>.
16. Taentzer, G., Ehrig, K., Guerra, E., de Lara, J., Levendovszky, T., Prange, U., Varro, D.: *Model Transformation by Graph Transformation: A Comparative Study*. 48.
17. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I.: ATL: A model transformation tool. *Science of Computer Programming*. 72, 31–39 (2008). <https://doi.org/10.1016/j.scico.2007.08.002>.
18. Burgueño, L., Cabot, J., Gérard, S.: The Future of Model Transformation Languages: An Open Community Discussion. *JOT*. 18, 7:1 (2019). <https://doi.org/10.5381/jot.2019.18.3.a7>.
19. Enríquez, J., Domínguez-Mayo, F., Escalona, M., García, J.G., Lee, V., Goto, M.: Entity Identity Reconciliation based Big Data Federation-A MDE approach. *International Conference on Information Systems Development (ISD)*. (2015).
20. Kordafahar, N.M., Rafeh, R.: On the optimization of cplex models. *International Research Journal of Applied and Basic Sciences*. 4, 2810–2816 (2013).
21. Lawler, E.L., Wood, D.E.: Branch-And-Bound Methods: A Survey. *Operations Research*. 14, 699–719 (1966).