



# A Model Driven Approach to Transform Business Vision-Oriented Decision-Making Requirement into Solution-Oriented Optimization Model

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

## ► To cite this version:

Liwen Zhang, Hervé Pingaud, Elyes Lamine, Franck Fontanili, Christophe Bortolaso, et al.. A Model Driven Approach to Transform Business Vision-Oriented Decision-Making Requirement into Solution-Oriented Optimization Model. IEA/AIE 2022 - 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Jul 2022, Kitakyushu, Japan. pp.211-225, 10.1007/978-3-031-08530-7\_18 . hal-03768919

**HAL Id: hal-03768919**

**<https://imt-mines-albi.hal.science/hal-03768919>**

Submitted on 12 Sep 2022

**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 Model Driven Approach to Transform Business Vision-Oriented Decision-Making Requirement into Solution-Oriented Optimization Model

Liwen Zhang<sup>1,2(✉)</sup>, Hervé Pingaud<sup>3</sup>, Elyes Lamine<sup>2</sup>, Franck Fontanili<sup>2</sup>,  
Christophe Bortolaso<sup>1</sup>, and Mustapha Derras<sup>1</sup>

<sup>1</sup> Berger-Levrault, Jean Rostand. 64, 31670 Labège, France  
{liwen.zhang, christophe.bortolaso,  
mustapha.derras}@berger-levrault.com

<sup>2</sup> CGI, University of Toulouse-IMT Mines Albi, Campus Jarlard, 81000 Albi, France  
{liwen.zhang, elyes.lamine, franck.fontanili}@mines-albi.fr

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

**Abstract.** Currently in our highly connected society, there is a strong requirement for decision-makers in organizations to coordinate and schedule their activities. Frequently, there are various uncertain factors, multiple objectives, many business knowledge and requirements, which heavily increase the difficulty of decision-making process regarding these issues. Therefore, a decision-maker will appreciate having control over the formulation of decision-making models and being able to adapt to highly dynamic situation. In this paper, we study a Model Driven Engineering (MDE) approach to link the business requirement defined by a model with solution-oriented logical models, which are codes that could be submitted to a combinatorial optimization solver. The design of our proposal follows the principles of three-levels Model Driven Architecture (MDA) and is based on a cognitive process for decision-making systems. Then, several transformation rules between models are explained to realize automatic Model to Model Transformation (M2M) with a special emphasis on the Platform Independent Model (PIM) to Platform Specific Model (PSM) part. To make a proof of our model transformation chain efficiency, a classical Travelling Salesman Problem (TSP) is chosen as a use case.

**Keywords:** Model driven architecture · Model to model transformation · Decision-making support · Scheduling problem

## 1 Introduction

Currently in a connected society with worldwide relationships, there is a strong incentive on productive organizations to regularly coordinate their activities within their environment when managing their products and service flows. For them, if it is necessary to achieve the right objectives (e.g., performance control and risk mitigation), such

objectives are often evolving and varying facing dynamics of the industrial and market contexts. Such modifications can occasionally obstruct autonomous decision-making abilities.

Due to the strategic nature of such problems, a regular debate about management culture within the company has emerged in recent years. Especially about the ability to fulfill the demands of multiple networks into which the organization is involved focusing on supply chain management approaches. Thus, when continuous improvement permeates the quality process, it is not only limited to operational and support processes, but also impacts those of management. It raises concerns about methods of thinking, decision-making activities, and decision-making aids.

The industrial engineering community has worked intensively and for some while to promote progress on such decision-making support systems. Planning and scheduling, for example, are not only strategic functions, but must be upgraded to strategic means. The challenge became to write “personalized” formulations of the relevant problems, considering local specificities in a given organization as real opportunities. Doing so, the trend is to include more and more business information in the decision-making process. Saving time being of prime interest, the short-term demands that company must manage are rising. Not only should the supply chain aim for economic and social performance goals, but it should also be able to deal with these unexpected changes in demand and know exactly how to meet them. As the importance of these uncertainties grows in the decision-making process, the utilization of decision support technologies becomes more critical. For all these reasons, a decision-maker will appreciate having control over the formulation of decision-making models and being able to adapt them when the situation is highly dynamic.

Mathematical solvers are tools (e.g., CPLEX, OptaPlanner) that have been widely disseminated. These digital technologies have historically been commonly used by experts in the decision-making support domain, including the Operations Research (OR) community. For tackling large scale constrained optimization problems, most solvers create with libraries of mathematical algorithms (e.g., branch and bound for CPLEX or simulated annealing for OptaPlanner, for example). After carefully implementing them using a descriptive language (e.g., respectively Optimization Programming Language-OPL for CPLEX, and Drools for OptaPlanner), users should be able to get a personalized solution quite effortlessly. Our research tries to facilitate the knowledge management for solving decision support problems using models. The objective is to investigate how to give the decision-maker an environment where the ability to fluently specify his/her problems online will be effective and where any change management would require minimal mathematical expertise.

This method of thinking prompts a reconsideration of modelling decision-making in business models. A transition must change from a high-level business knowledge to the knowledge required by a solver to perform calculations in order to optimize decisions. We discuss and develop our decision support environment with this goal in mind. The approach for making the transition concrete is based on Model Driven Engineering (MDE). We apply an MDE process by mapping (1) business knowledge and expertise available at the Computer Independent Model (CIM) level of the Model

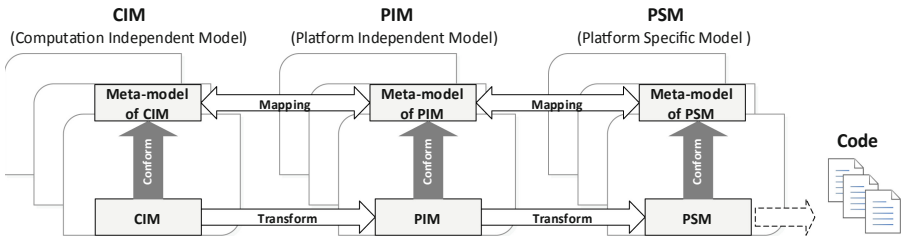
Driven Architecture recommended by OMG [1] (2) formal addressing of a decision-making problem in mathematical transcription at the Platform Independent Model (PIM) level (3) ability to make decisions by a solver at the Platform Specific Model (PSM) level. Hereafter, we will further study these conceptual links between mathematical decision support models and logical solution-oriented models.

The following content is divided into five sections. In the section two, we briefly remind the foundation of MDE, its basic concepts and what may be a lifecycle of an MDE process. Then, we proceed to a literature review about past references on the same subject. In section three, an overview of our MDE approach is explained. Then, we put the emphasis on the PIM to PSM part, which extends some of our previous results [2]: the transformation from a Graph-based Operations Research Model (GORM) to a Solution-oriented mathematical Model (SM). The two meta-models and the transformation rules defined between them to achieve the Model-to-Model transformation (M2M) are discussed in the next section. We have chosen the Travelling Salesman Problem (TSP) as a use case to illustrate our study throughout the paper. Finally, we deliver conclusions about the study and draw some perspectives for future research on this field.

## 2 Past Related Studies

### 2.1 Theoretical Foundation of MDE

MDE is motivated by the idea of valuing models for a certain purpose [3]. OMG (Object Management Group) is a well-known nonprofit international organization who promoted this usage more than twenty years ago. One of their recommendation for MDE is a standard named Model Driven Architecture (MDA) which was published at the end of 2001 [1]. This standard has been a reference framework for a long time. This initiative delivers two concrete learning outcomes. Firstly, it clarifies the relationships between a model and a meta-model located at two different levels into the MDA pyramid. A meta-model is a model that defines the language used to express a model [4]. Second, the MDA requires that a software production design system be divided into three components, resulting in a two-stage engineering life cycle. As a direct result, three different models are identified, one for each component of the life cycle: Computer Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM) [5].



**Fig. 1.** Model-to-model transformation based on Model Driven Architecture

Model-to-Model transformation (M2M) is a key feature. As shown in Fig. 1, M2M begins with a business model which is independent of the technique (CIM), to generate

another model which is independent of a development platform (PIM) during the first stage of the life-cycle. During the second stage, the PIM model is a source and its transformation ends to a model which is specific to an executing platform (PSM). According to [6], M2M performs globally an automatic generation of a target PSM model from a CIM source model, following two successive transformation by specifying mapping rules at the meta-model level. Mapping rules could be summarized as translation directives of models expressed by meaningful links between meta-models. In our application, let us remind that the ultimate target is a source code for a solver, written in its implementation language.

## 2.2 Previous Experiences in M2M

In recent years, many M2M transformation methods have been investigated by the scientific community working on MDE for engineering purposes. In researching applications in this engineering field, we have found many traces of such experiments. In healthcare, a prototype is designed to generate a human-machine interface on mobile technologies (target model, PSM), starting from the modeling of a care plan (source model, CIM) [7]. In [8], the authors explain how to transform a CIM model entitled “International Electrotechnical Commission Common Information Model” to a target PSM mathematical formal “Modelica” model. This target model is used to perform a dynamic simulation of a complex physical system. In [9], the authors use the BPMN notation to model business processes from a CIM (source model) to a PIM (target model). The target model includes static, dynamic and functional views, which are expressed using three appropriate UML diagram types. In terms of industrial applications, enterprise systems interoperability between partners involved in different configurations of collaborative networks have been extensively addressed by MDE. We refer to research works on Mediation Information System Engineering (MISE) [10]. The approach collects the knowledge about a company active in a collaboration network, at the CIM level through an organizational model. The PIM level represents the technical knowledge that is a collaborative process model in BPMN language. Finally, the PSM performs an extraction of knowledge from the collaborative process model to make a digital service orchestration in a bus of enterprise services [11].

The key element in model transformation is a Model Transformation Language (MTL) [12]. Initially, OMG proposed the Query/View/Transformation (QVT) as a model transformation language candidate [13]. Visual Model Transformation Language (VMTL) [14] is another MTL initiative that supports endogenous transformations (the source and target model conform to the same meta-model). The Graph Rewriting And Transformation (GReAT) language was introduced by [15], it allows transformations from one domain to another using heterogeneous meta-models. ATL (Atlas Transformation Language) is an MTL of the QVT type [16], which is extensively cited in the literature. ADOxx specific Scripting language (AdoScript) is a script language with the characteristics of a MTL, which is integrated in the ADOxx meta-modeling platform developed by OMILAB [17]. This language allows to access and process the information stored in the models, and then to perform multiple functions such as M2M, model visualization and simulation.

As far as we know, no reference exists about an application of MDE for the transformation of business models into a mathematical model that feeds a combinatorial optimization-oriented solver to solve scheduling problems. We would like to emphasize that our approach allows business experts with no background in operations research to easily build optimization models, aiming at providing them with decision-making supports. For MTL, we select AdoScript which is embedded in the ADOxx meta-modeling platform [18], because it integrates both design of modeling languages and use of M2M features. ADOxx is extensively used to build Proof of Concept (PoC) in research projects. The OMILAB community [17] shares experiences and provides efficient technical supports developed with ADOxx for a purpose of reusability.

### 3 MDE for Decision-Making Process Design

#### 3.1 Cognitive Process for Decision-Making System

We will describe the decision-making system through a knowledge chain that evolves from a preliminary stage of business requirements analysis, to a decision-making support model that meets these requirements. Three types of models are sequentially used in this chain. As shown in Fig. 2, this breakdown in many models is similar to a cognitive process that a decision-making process designer will apply. It is resumed in three steps: (1) decision-making requirements explanation for business expert's perspective, (2) decision-making support-based formulation and (3) preparation of solution calculation for OR expert's perspective. At the end of each step, the MDE tool produces a new model which is an input for the next step.

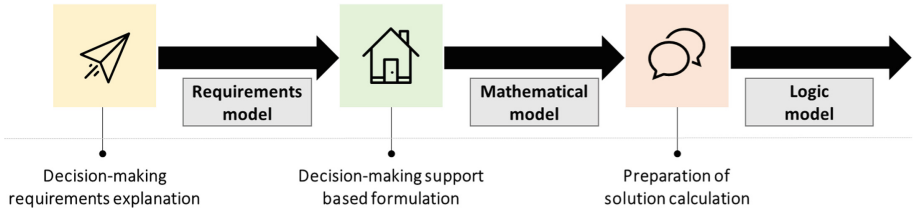


Fig. 2. Cognitive process of a decision-making system

#### 3.2 Cognitive Process-Based Model Driven Architecture

Figure 3 extends the conceptual view of Fig. 2 following the MDA principles depicted in Fig. 1. Three types of decision-making models are considered in this new representation:

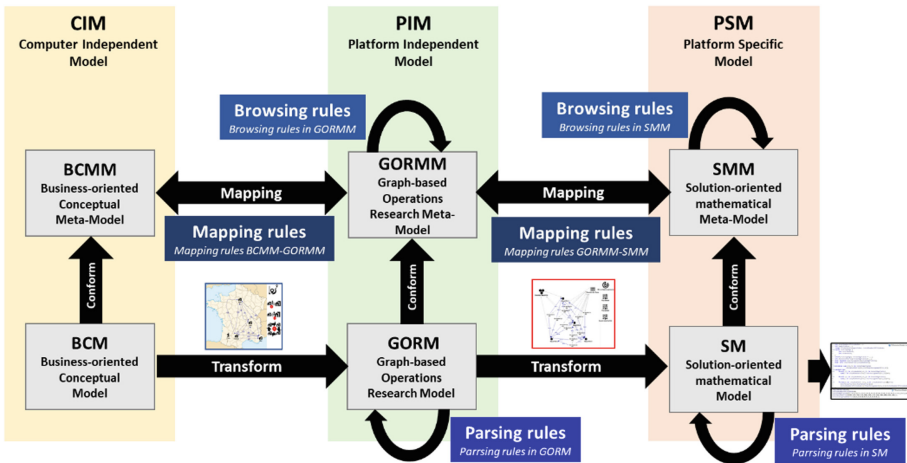
- The CIM level is the specification of a decision-making need. A **requirement model** entitled “Business-oriented Conceptual Model (BCM)” captures the relevant knowledge before addressing the decision-making problem (e.g., the TSP).

- The PIM level is typically devoted to what the operations research will perform to translate this BCM into a mathematical model of the problem. The resulting **mathematical model** is formalized by a graph and entitled “Graph-based Operations Research Model (**GORM**)”.
- The PSM level follows and has to prepare the mathematical problem-solving computations to be performed within a specific execution environment. It consists in a coding of the GORM using a solver’s specific language. Therefore, this last **logic model** is entitled “Solution-oriented mathematical Model (**SM**)”.

To complete the generation of the PSM from the CIM, a collection of transformation rules is an input of the MTL that has to be defined, and we made it with AdoScript. As shown in Fig. 3, three types of transformation rules are distinguished:

- **Mapping rules:** these rules give birth to one or several element(s) of the target model from one or several elements of the source model.
- **Browsing rules:** these rules are defined in target meta-model, by processing locally some elements. Their aim is to check consistency of the new elements with the relevant meta-model.
- **Parsing rules:** these rules also operate local treatments, but only in model level of PIM and PSM. The production of code lines must be compliant with the syntax of the programming language (e.g., OPL for computation of the TSP by CPLEX).

In [2], the transformation between CIM (BCM) and PIM (GORM) has been explained. Therefore, the content of this paper is complementary to this previous publication. We focus hereby on the transformation from PIM (GORM) to PSM (SM), providing technical information about it. Theory and practice are still illustrated in the same use case (TSP).



**Fig. 3.** Overview of the cognitive process developed following MDA principles

## 4 PIM to PSM Transformation Applied to TSP

The transformation from PIM to PSM requires (1) the declaration of relevant data (set of parameters and set of decision variables), and (2) the definition of the constraints and the objective function.

Figure 4 shows respectively both ends of this transformation work: the GORM as an input model and the SM as an output for a given TSP use case. This example concerns four cities over a metropolitan area. The objective function is the travel distance we want to minimize. The left side of Fig. 4 is a graph of the cities to be visited by the traveler, with many annotations related to the possible pathways. The right side of Fig. 4 is the OPL code with the OPL keywords written in blue color.

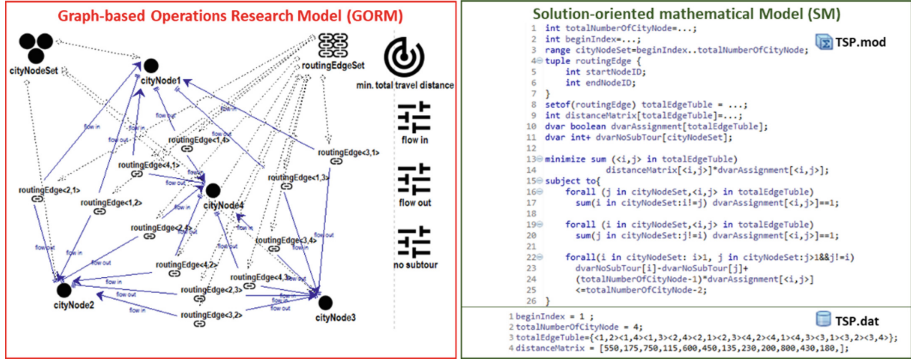


Fig. 4. GORM and SM on a case study

Because the GORM will conform to the meta-model called Graph-based Operation Research Meta-Model (**GORMM**), the reader is invited to refer to the previous paper [2] for further details on the related concepts. Similarly, as our SM must conform to a meta-model called Solution-oriented mathematical Meta-Model (**SMM**, see Fig. 3), we will just give some details about the concepts defined in this SMM in the next section.

### 4.1 Specification of Solution-Oriented Mathematical Meta-model (SMM)

The design of the SMM is inspired by [19] with the two expected parts: (1) data declaration view and (2) formulation building view. Describing each part independently in the following, we use figures (from 1 to 6, in black circles) to map the literal description with the graphical representation of the SMM.

For data declaration, we list all the concepts in Table 1 including the data types in the OPL syntax.

For formulation building, the transformation is supported by a library of mathematical relation patterns from which the OPL formulas of the constraint and the objective functions can be adapted consistently. Thus, our transformation is guided by the need to identify the better appropriate formulation pattern within the library for each constraint



**Table 1.** Overview of concepts in SMM: data declaration view

<b>1</b>	<b>Data Declaration</b>	It represents the relationship established between data and their types in OPL. The attributes include: <ul style="list-style-type: none"> <li>• name: the name retained to reference a data to be declared.</li> <li>• type: the data type in OPL.</li> <li>• isDecisionVar: a Boolean indicating if a data is a decision variable, which means that this type of data can be modified during optimization process in the relevant decision space.</li> <li>• indexRep: this attribute is enumeration type including three forms of index declaration in scalar, couple or triplet: "i or j or k", "&lt;i,j&gt;" and "&lt;i,j,k&gt;".</li> </ul>
<b>2</b>	<b>AbstractType</b>	A generic descriptor for data type (scalar, vector or hybrid structure) in OPL whose subclasses can be derived with an implementation by a keyword. Inheriting this generic concept, the other child concepts are extracted from the work of [19].

**Table 2.** Overview of concepts in SMM: formulation building view

<b>3</b>	<b>Formulation Building</b>	An abstract concept which is a container of formulation patterns written in OPL. It is characterized by: <ul style="list-style-type: none"> <li>• name: same function as the identifier (name) of the “Constraint” or “Objective Function” concept in GORM.</li> <li>• pattern content: this concept aims to identify the characteristics of OPL patterns before the allocation of the data required by the calculation.</li> </ul>
<b>4</b>	<b>Constraint Building</b>	This concept inherits the “Formulation Building” concept. It has an attribute "equation in CPLEX" aiming at displaying the complete formulation of a constraint once the components of the concerned pattern have been assembled.
<b>5</b>	<b>OF Building</b>	This concept inherits the “Formulation Building” concept. It performs the similar function to "Construction Building", but for the Objective Function.
<b>6</b>	<b>Formulation Component</b>	This concept contains the attributes to indicate the formulation components for the relevant pattern, based on the library that stores these patterns. It is characterized by the name of concerned pattern (name), parameters and decision variables with an adapted number: parameter 1 [P1], parameter 2 [P2], parameter 3 [P3], decision variable 1 [D1], decision variable 2 [D2].

and the objective function. Then, it is to perform an appropriate instantiation with the right problem data. Table 2 summarized the formulation building view of the SMM.

Figure 5 is a representation of this SMM as an UML class diagram. Note that the descriptors for the data structures (the “AbstractType” concept along with all its sub-concepts) draw directly from the work we have been inspired [19].

## 4.2 Transformation Process

The transformation process operates in three times:

1. Firstly, the M2M transformation is carried out through inferring the mapping rules between GORMM and SMM.
2. Then, a browsing takes place inside the SM at the PSM level. This is an automatic processing to set up the adapted call means for the two formulations “constructors”.

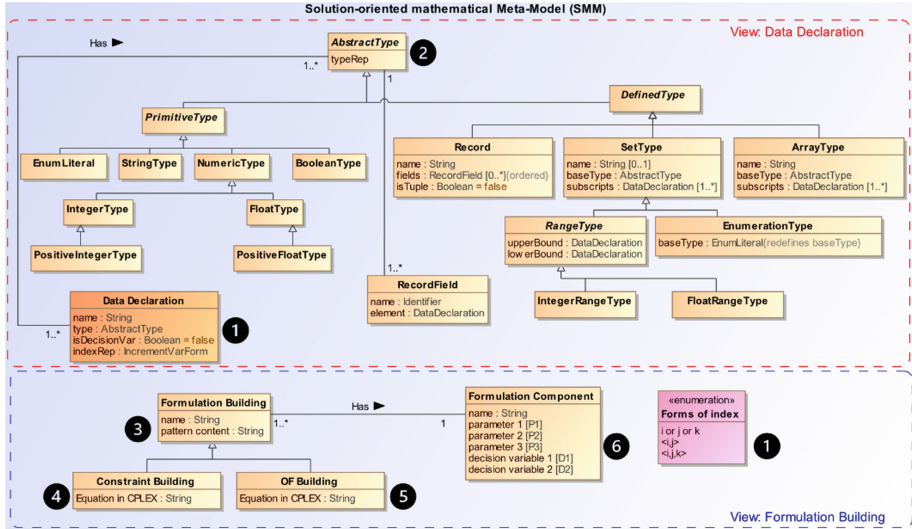


Fig. 5. Solution-oriented mathematical Meta-Model (SMM)

3. The third time is a data compiling in respect to the OPL syntactic rules. A parser developed in Java is making the coded lines assembly, which is partly directed by an interpretation of the OPL keywords.

At the end of the inference of this set of rules, we get the final model that can be submitted to CPLEX for running the search of an optimum result. We mean that this SM OPL-based model can technically be submitted to the CPLEX software to find the solution of a TSP. In the following, we will provide further details about the three categories of rules introduced above, regarding these three times of the transformation concerning the TSP use case.

## Mapping Rules Between GORMM and SMM

### Data Declaration

The translation of GORM data into the SM correct type is focused on two concepts called “sets”: the “CityNodeSet” and the “RoutingEdgeSet”. This translation is performed towards the declarative part of the OPL code with a differentiation between parameters and decision variables belonging to the model. Each of them is then assigned its correct keyword regarding the data type required by OPL. This translation is made in two parts:

1. The declaration of the “CityNodeSet” starts by choosing the primary type: “RangeType”, then primitive integer types are chosen for two components: its upper bound (“totalNumberOfCityNode”) and lower bound (“beginNodeIndex”). Finally, the declaration of the decision variable “dvarNoSubtour” is made. Notice that the expression of variable “dvarNoSubtour” in OPL is as follows:

```
dvar int+ dvarNoSubtour [CityNodeSet]
```

Taking the declaration of “*dvarNoSubTour*” as an example will help us to explain the mapping process. Firstly, we indicate the basic characteristics of this variable by creating an object of the class “Data Declaration” in the SM, including the indication of the variable name, the determination of variable type (“*isDecisionVar* = true” refer to the assignment of *dvar* for the variable by the parsing rules). Secondly, the type of “*dvarNoSubtour*” is a one-dimensional vector array (“Array” type). It begins with a declaration of its primitive type ( *int*+) and ends with an indication of the array size into brackets (“CityNodeSet”).

2. Next, a set of successive translations are performed around the concept of “RoutingEdgeSet”. Figure 6 shows an example of mapping to declare both “*totalEdgeTuple*” and “RoutingEdge”. From this figure, the declaration of the “*totalEdgeTuple*” is realized in two steps to achieve its OPL expression, which is formulated as follows:

```
setof(routingEdge) totalEdgeTuple
```

Then, the declaration of the “RoutingEdge” is done in three steps to obtain the following expression in OPL:

```
tuple routingEdge{
    int startNodeID;
    int endNodeID;}
```

### *Formulation Building*

This type of transformation uses pointers to build the code lines related to the mathematical patterns in the executable statements of the SM. Therefore, the transformation should start by building firstly a body of an equation in a library, including keywords and operators, the order must be respected one each other of an equation. Afterwards, the pointer is used to indicate the elected mathematical formula in the library. The library is indexed to map to the “Constraint” and “Objective Function” concepts defined in the GORM, and written as function types in AdoScript. After this selection performed, the pattern of each formulation is recorded in the attribute “pattern content” of the concept “Formulation building”.

After, a rule is used to generate components by completing the patterns with the relevant parameters or decision variables of the mathematical relations. This phase is based on a catalog of specification for the patterns belonging to the library. Briefly speaking, the catalog supports the necessary distribution of components (see “Formulation Component” in Fig. 6) over a given pattern in the library.

## Browsing Rules in SM

The local browsing aims to make an assembly of all the results inherited from the previous work “Formulation Building”, together with those from “Formulation Component”. The assembly process consists of (1) components-pattern assembly to build formulations, (2) final assembly assigned to the “*equation in CPLEX*” attribute of “Constraint Building” or “OF Building” concept, respectively.

## Parsing Rules for GORM and SM

This operation performs the role of a parser for the GORM and SM. An OPL model in CPLEX consists of two files: a data file (.dat) and a model file (.mod). On one hand, the parser extracts the target “.mod” file from the SM. On the other hand, the parser performs the export of the GORM to the target “.dat” file, which contains the values of the relevant parameters defined in the “.mod” file (e.g., “*beginNodeIndex*”, “*totalNumberOfCityNode*”, “*totalEdgeTuple*”, “*distanceMatrix*”).

As models (BCM, GORM and SM) are created in ADOxx and, as so, can be exported in an XML format, the parser was developed in Java using Java Architecture for XML Binding (JAXB). To complete the generation of the OPL model, a series of rules developed in Java fulfill the function of keyword assignment, for example:

- The decision variables “*dvarNoSubTour*” and “*dvarAssignment*” must have a keyword “*dvar*” as a prefix.
- The set of constraints must be surrounded by “*subject to {...}*”.

In addition, this phase allows the formatting of two output files, for example:

- The “*= ...;*” that follows the declared parameters.

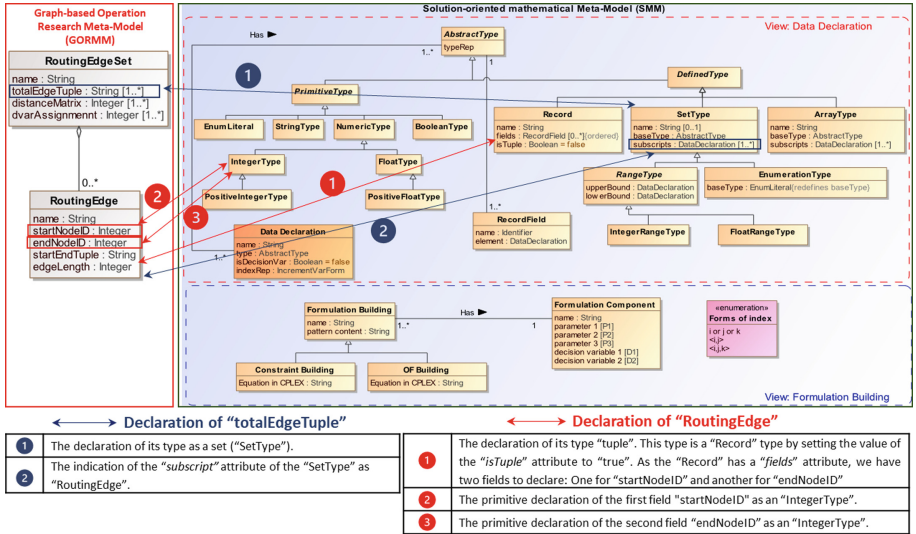





Fig. 6. Mapping rules for declaring “*totalEdgeTuple*” and “*RoutingEdge*”

- The “{ }” for data of type “tuple”.
- The “...” connector for data of type “range”.
- The separator “;” or the space (signified by “\n” in regular expression in Java) to isolate the data and the specified formulations.







## 5 Case Study

Let us go back to the BCM using the graphical operational semantics of our BCMM, which is recalled in Table 3. The same notation is used to report the results of the computation made by the solver.

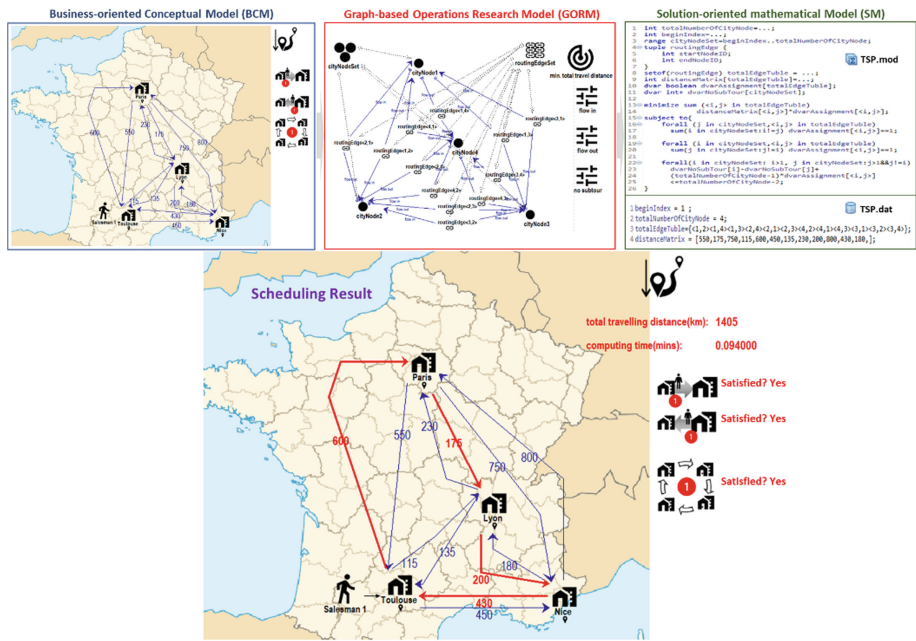
In this case, we have four cities to be visited by the “salesman 1” which are: Toulouse, Paris, Lyon and Nice. Three constraints are considered and symbolized by an operational semantics (i.e., ,  and ). The objective is to minimize the salesman’s total routing distance. All the mentioned contextual information is modeled by business experts using BCM at CIM level.

As described in the upper part of Fig. 7, the three MDA models are shown from left to right, i.e., CIM/PIM/PSM. BCM is first transformed into GORM at the PIM level, and then into SM at the PSM level through two transformation process: CIM to PIM, PIM to PSM. The PSM has been compiled by CPLEX correctly. The result given by CPLEX is for a TSP with four cities to visit and “Salesman 1” will travel is the following optimal order: Toulouse → Paris → Lyon → Nice → Toulouse. This

**Table 3.** Graphical representation of the concepts included in GORMM

Name of concept	Enumeration element for “type” attribute	Graphical rep.
Salesman		
City		
Business constraints (BuConst)	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.	
Goal	Minimize the salesman’s total routing distance	

result is visualized graphically by simply changing the color of the optimal path. The achieved goal is 1405 kms qualifying a minimal total distance along with a very low computational time on this small-size problem. The specified business constraints are all well-satisfied, as demonstrated by the annotation to the question “satisfied?” for each business constraint.



**Fig. 7.** Use case specification of TSP: the three models and the final scheduling result

## 6 Conclusion and Research Perspectives

This research work addresses a model driven approach to transform a business vision-oriented decision-making model into a solution-oriented optimization model, following the three levels (CIM-PIM-PSM) of the Model-Driven Architecture (MDA). Considering the result of our previous work [2], we have focused on the Model to Model transformation (M2M) between a Graph-based Operations Research Model (GORM) at PIM level and a Solution-oriented mathematical Model (SM) at PSM level. The AdoScript Model Transformation Language (MTL) has been used to perform the PSM generation using Model to Model translation. In the core of this M2M machinery, three categories of transformation rules have been described. To make a proof of concept of the MDE approach, we decided to apply it on the well-known Travelling Salesman Problem (TSP). Considering new perspectives, future work will address a problem that is more difficult to deal with than a TSP. We have chosen to extend our approach towards this new case study of the Home Health Care Routing and Scheduling Problem (HHCRSP)

[20], considered as a clustered multi-TSP. This problem is the subject of many variations related to different organizations in charge of home health care coordination of caregivers, including the diversity of practices, caregiver skills, means of transportation, work schedules and so on. It appears to be particularly prone to make extension of our approach. Upcoming studies of an agile model-based environment to support efficiently decision-making for HHCRSP are planned.

## References

1. Poole, J.D.: Model-driven architecture: vision, standards and emerging technologies. In: Workshop on Metamodeling and Adaptive Object Models, ECOOP. Citeseer (2001)
2. Zhang, L., Fontanili, F., Lamine, E., Bortolaso, C., Derras, M., Pingaud, H.: A systematic model to model transformation for knowledge-based planning generation problems. In: Fujita, H., Fournier-Viger, P., Ali, M., Sasaki, J. (eds.) IEA/AIE 2020. LNCS (LNAI), vol. 12144, pp. 140–152. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-55789-8\\_13](https://doi.org/10.1007/978-3-030-55789-8_13)
3. OMG: MDA Guide revision 2.0 (2014)
4. Benaben, F.: Conception de Système d'Information de Médiation pour la prise en charge de l'Interopérabilité dans les Collaborations d'Organisations (2012). <https://hal-mines-albi.archives-ouvertes.fr/tel-01206234/document>
5. Truyen, F.: The fast guide to model driven architecture: the basics of model driven architecture. Whitepaper, Architecture Oriented Services, Cephas Consulting Corp. (2006)
6. Kleppe, A.G., Warmer, J., Warmer, J.B., Bast, W.: MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Professional, Reading (2003)
7. 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 (2008). <https://doi.org/10.1109/ASE.2008.75>
8. Gómez, F.J., Vanfretti, L., Olsen, S.H.: CIM-compliant power system dynamic model-to-model transformation and Modelica simulation. IEEE Trans. Industr. Inf. **14**, 3989–3996 (2018). <https://doi.org/10.1109/TII.2017.2785439>
9. Rhazali, Y., Hadi, Y., Mouloudi, A.: Model transformation with ATL into MDA from CIM to PIM structured through MVC. Procedia Comput. Sci. **83**, 1096–1101 (2016). <https://doi.org/10.1016/j.procs.2016.04.229>
10. Mu, W., Benaben, F., Boissel-Dallier, N., Pingaud, H.: Collaborative knowledge framework for mediation information system engineering. Sci. Program. **2017**, 1–18 (2017). <https://doi.org/10.1155/2017/9026387>
11. 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**(2), 323–376 (2016). <https://doi.org/10.1007/s10257-016-0321-z>
12. Bézin, J., Büttner, F., Gogolla, M., Jouault, F., Kurtev, I., Lindow, A.: Model transformations? Transformation models! In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) MODELS 2006. LNCS, vol. 4199, pp. 440–453. Springer, Heidelberg (2006). [https://doi.org/10.1007/11880240\\_31](https://doi.org/10.1007/11880240_31)
13. Kurtev, I.: State of the art of QVT: a model transformation language standard. In: Schürr, A., Nagl, M., Zündorf, A. (eds.) AGTIVE 2007. LNCS, vol. 5088, pp. 377–393. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-89020-1\\_26](https://doi.org/10.1007/978-3-540-89020-1_26)
14. Acrețoai, V., Störrle, H., Strüber, D.: VMTL: a language for end-user model transformation. Softw. Syst. Model. **17**(4), 1139–1167 (2016). <https://doi.org/10.1007/s10270-016-0546-9>

15. Agrawal, A., Karsai, G., Neema, S., Shi, F., Vizhanyo, A.: The design of a language for model transformations. *Softw. Syst. Model.* **5**, 261–288 (2006). <https://doi.org/10.1007/s10270-006-0027-7>
16. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P.: ATL: a QVT-like transformation language. In: *Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications*, pp. 719–720 (2006)
17. Fill, H.-G., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. *Enterp. Model. Inf. Syst. Archit. (EMISAJ)* **8**, 4–25 (2013). <https://doi.org/10.18417/emisa.8.1.1>
18. Karagiannis, D., Kühn, H.: Metamodeling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) *EC-Web 2002. LNCS*, vol. 2455, p. 182. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-45705-4\\_19](https://doi.org/10.1007/3-540-45705-4_19)
19. Assouroko, I., Denno, P.O.: A metamodel for optimization problems. National Institute of Standards and Technology (2016). <https://doi.org/10.6028/NIST.IR.8096>
20. Zhang, L.: De la vision métier à la génération assistée de plannings pour la coordination centralisée de services de soins à domicile. Ph.D. thesis manuscript (2021). <https://tel.archives-ouvertes.fr/tel-03405854>