



## APPROACH FOR MANAGING VARIABILITY IN DATABASE SCHEMA



Nesrine Khalfallah<sup>1\*</sup>

Sami Ouali<sup>2</sup>

Naoufel Kraiem<sup>3</sup>

<sup>1,2,3</sup>RIADI Laboratory, ENSI, Campus of Manouba Manouba, Tunisia

<sup>1</sup>Email: [khalfallahnesrinee@gmail.com](mailto:khalfallahnesrinee@gmail.com) Tel: +21695740444

<sup>2</sup>Email: [samiouali@gmail.com](mailto:samiouali@gmail.com) Tel: +21623695033

<sup>3</sup>Email: [naoufel.kraiem@ensi.rnu.tn](mailto:naoufel.kraiem@ensi.rnu.tn) Tel: +9689208657208



(+ Corresponding author)

### ABSTRACT

#### Article History

Received: 14 May 2018

Revised: 6 August 2018

Accepted: 13 August 2018

Published: 16 August 2018

#### Keywords

Databases

Schemas

Variability

Software product lines

Metamodels

Model transformation.

The variability in software product lines is well studied, but this study is well neglected at the level of the databases. This insufficiency drives us to work on the modelling and the implementation of the variability in the databases in SPLs. Therefore, our contribution is proposing an approach to manage variability in database schemas. In this approach, we will focus primarily on modelling variability in the conceptual databases schemas at the domain engineering level. We will use model-driven engineering to ensure the different data models transformation to obtain a variable database model. Then, we will deal with application engineering by managing variability at the logical database schema through a configuration to obtain a variable relational database schema. Moreover, we proposed extensions for a metamodel to support the variability notions proposed. We also offer a tool support to apply and automate our approach and to test case studies.

**Contribution/ Originality:** The paper's primary contribution is proposing an approach to model and implement variability in the different database schemas in a software product line. This approach models variability in the conceptual database schema which generates a relational and variable schema. A tool support is implementing to verify the approach validity.

## 1. INTRODUCTION

The software product lines engineering [1]; [2] is a method of software engineering, which focuses on a group of similar systems sharing a set of identical and different features. Although the database schemas are an integral part of information systems. The use of software product lines (SPLs) has been principally studied for the production of an executable code. The impact on data management in the database schemas is poorly documented and rarely studied in the literature [3]; [4]; [5]; [6]. Yet there are various interests like flexibility, genericity, adaptation to the user's needs, interoperability, etc. The current approaches are managing the modelling and the implementation of databases, regardless of how to address the problems of the variability in the databases schemas. So, it would be beneficial to address a field of engineering that focuses on the analysis and modelling of future adaptation in databases in SPL. So, our goal is to propose an approach which allows creating a relational and variable schema of a database using the techniques of software product line engineering. Indeed, a databases set is considered as a product line whose product is a particular database. The variable schema will ensure a continuous evolution of the database, as it adapts to any changes in its product line. In addition, we will work on the modelling

of the variability at the level of the conceptual schemas of the database lines as well as on the implementation of the variability in the logical and physical schemas of these last ones. Therefore, to study and understand greatly the variability in the database we need a variability management framework. By consequence, we first use the multi-facet and multi-view approach to propose a comparison framework [7]. Moreover, we will propose an approach to manage variability in the database schema [8]. Through this approach, we have tried to propose solutions to remedy the insufficiencies extracted from the comparison framework. Therefore, in this paper, we have tried to manage the complexity of modelling variable databases schemas and to implement an automatic method of generating this type of schema, and then test it on a case of study. To carry out this task, we decided to begin managing the variability from the most abstract model to coding. As our work is based on modelling and meta-modelling by using UML, we chose to work with model-driven engineering (MDE) approach of Object Management Group (OMG): Model-driven architecture (MDA). Actually, we prospered to manage the variability criterion automatically within the database schema through creating new parameter tables in the variable database schema that contains optional features of the principal tables of the initial database schema. These tables are mainly generated after a normalization of a variable conceptual model of the database created after the execution of our approach. Finally, in this paper, we proposed a tool support to apply our approach and verify its validity through case studies. After, we proceed to an evaluation of the variable database schema resulting from our approach by contributions to the other schemas resulting from the existing approaches. Later, we close with the positioning of our approach by contributions to the existing ones.

In this paper, we describe our proposed comparison framework in section 2. In section 3, we present our proposed approach to modelling and implementing variability in the databases schemas. Furthermore, in the same section, we introduce the proposed metamodel which supports the variability concepts. The application of our approach is illustrated by a case study in section 4. Then, in section 5 we present our tool support developed to execute our proposed approach. We evaluate it in a qualitative and quantitative way in section 6. Finally, we discuss related work in section 7 and conclude the paper in section 8.

## 2. COMPARISON FRAMEWORK

The modelling and the development of variability in the database is a multifaceted activity, concerning not only technical problems but also organizational, managerial issues. So to understand well, this discipline should be studied deeply from all sides. Then, in this work, we present an attempt to explore some of the problems underlying variability management in the database through analysis of some of the current approaches used to the management of the variability issue in the database. Therefore, we thought about proposing a comparison framework. This Framework aims to:

- Facilitate the comprehension and the classification of problems relating to modelling process and implementation of the variability in the databases.
- Provide domain experts and not experts a framework for identifying different facets of modelling and implementation of the variability in the databases.
- Evaluate current methods of modelling and implementation of variability in databases.
- Summarize the advantages and the inconveniences of different approaches of database variability management.

So, we adopted a comparative approach able to handle this discipline in worlds to facilitate its analysis in order to have more reliable results. Hence, we will present the variability management in the database from four different viewpoints showing each one a particular aspect of this discipline, which are: subject, system, usage, and process. Thus, the most suitable is to work with a faceted classification approach which is used in many engineering works in the literature [9]; [10]. To each view is associated a set of facets which are respected as viewpoints appropriate to specify and classify approaches. A metric is associated with each facet which is characterized by a set of important

attributes. Approaches are placed in the framework by assigning values to the attributes of each facet. Attribute values can be a predefined type (Integer, Boolean, etc.), an enumerated type (ENUM{x, y, z}), or a structured type (SET or TUPLE).

The multi-facet and multi-view approach allows having an overview of the modelling of the variability in the databases. The views show the variety of modelling variability in the databases. The facets provide a deep description of these aspects [11]. The four views which construct the framework respond to four questions about the modelling of the variability in the databases. What is the management of the databases variability? How methods for modelling variability in databases are represented? What are the objectives of users regard database variability and the results derived from this variability? How to implement the variability in applications integrating a database? Each view with its facets answers and explains one of the four questions presented above. The subject view answers to the first question by presenting the nature of our discipline. Then, the usage view answers to the second question by determining the means used to present this discipline. Next, the system view answers to the third question by presenting the variability management method goal. Finally, the process view answers to the fourth question by focusing on the implementation process and its details.

According to this literature, we have proposed a variability management framework presented in figure 1.

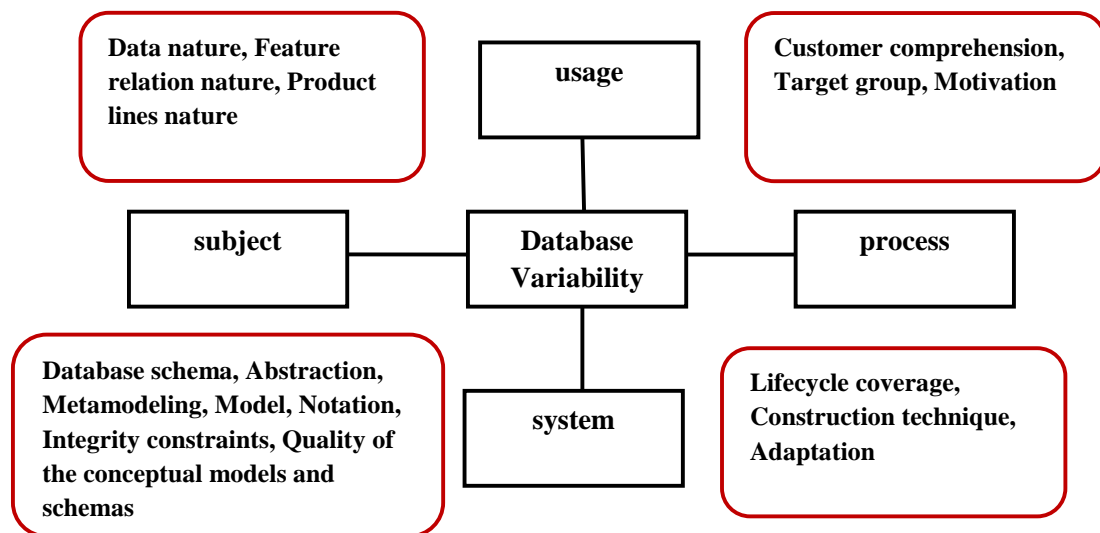


Figure-1. Variability management framework [7].  
Source: Khalfallah, et al. [7]

### 3. MODEL-DRIVEN ENGINEERING FOR MANAGING VARIABILITY IN DATABASE

#### 3.1. Model-Driven Variability Modelling Process

In this work, we will present a variability management process in database schema in SPLs. This process has as input a source model  $M_a$  and as output a target model  $M_b$  generated using transformation rules. Figure 2 illustrates this process in a general way, at the level of the domain engineering, there is as input models:  $M_a$  that is represented by the class diagram and the feature model of the studied software product line. And  $M_b$  is represented by a database model of software product line. Then, at the level of the application engineering after a configuration, we obtain a variable relational schema.

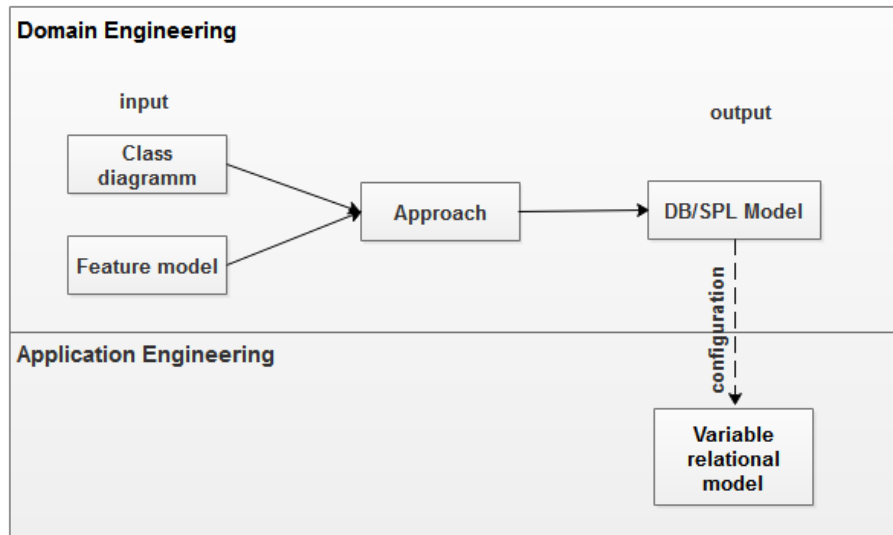


Figure-2. Illustrative schema of the proposed approach

In our work,  $M_a$  is a class diagram of an academic case study “Media library” which we will work on it. It is presented in figure 3.

In this process, there are two levels:

- At the conceptual level, there is a transformation from a source model to another target model based on the MDE approach. Then, after a normalization of the target model, the variable database schema is generated. Figure 4 schematizes the passage at this level.
- At the physical level, there is a transformation from a conceptual to a physical level.
- In this section, we will be interested in the conceptual level where we found transformations from one model to another.

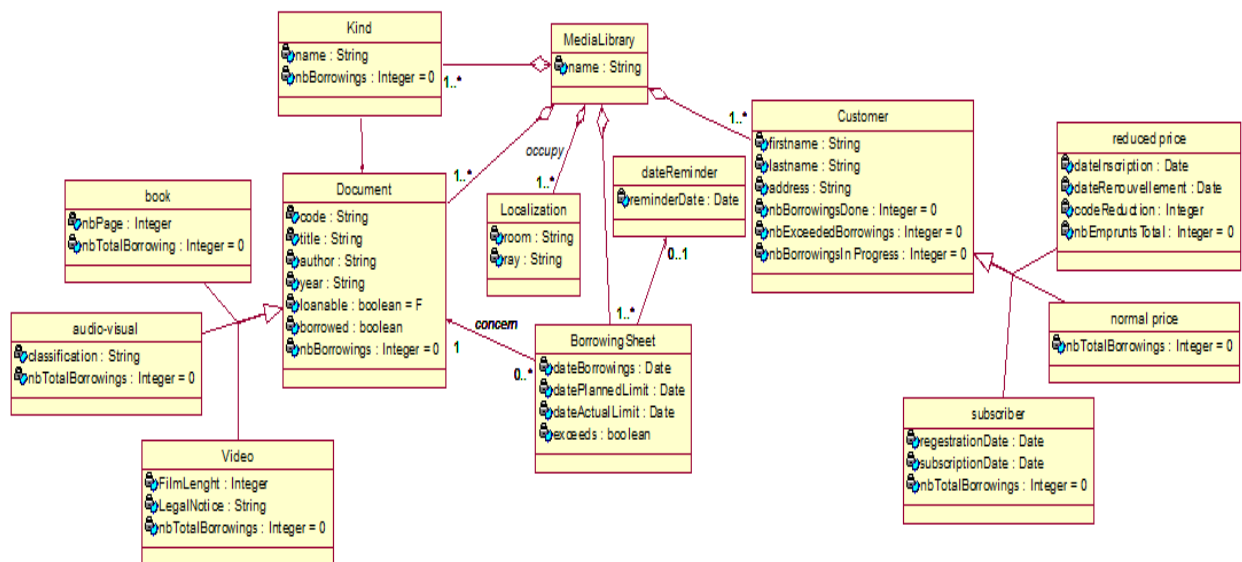


Figure-3. Class diagram Media Library

Source: Khalfallah, et al. [8]

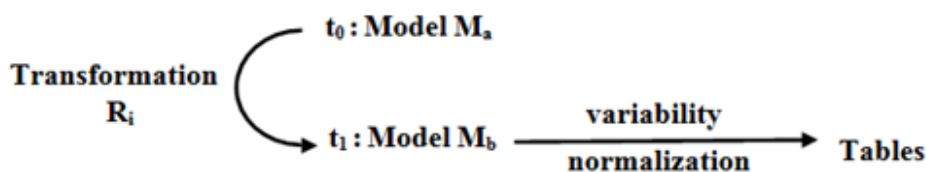


Figure-4. The process of managing variability at the conceptual level

In the beginning, we will work with a part of the class diagram mentioned in figure 5 including only the classes: "Borrowing Sheet", "Document" and "Kind", as we want to lighten the volume of the class diagram at this stage of the work. This allows us to focus on the part that is most useful to our work at this stage and to make the process to follow clearer and easier to understand. It should also be mentioned that the variability in this part of the diagram is determined by the FODA model shown in figure 6. Indeed, this feature model specifies the optional and mandatory attributes of the classes of the diagram.

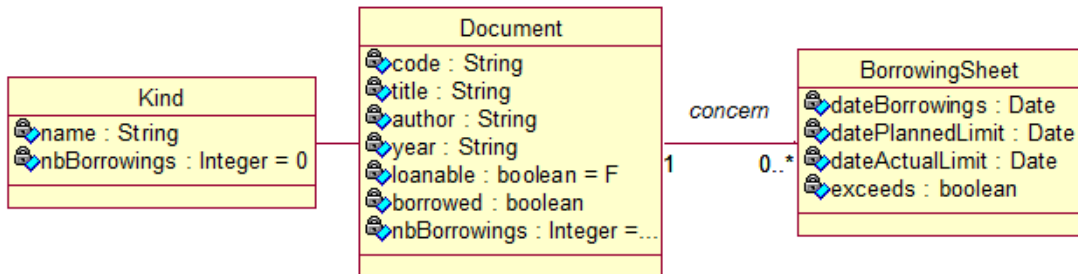


Figure-5. Source model M<sub>a</sub>: Part of the class diagram Media Library

Source: Khalfallah, et al. [8]

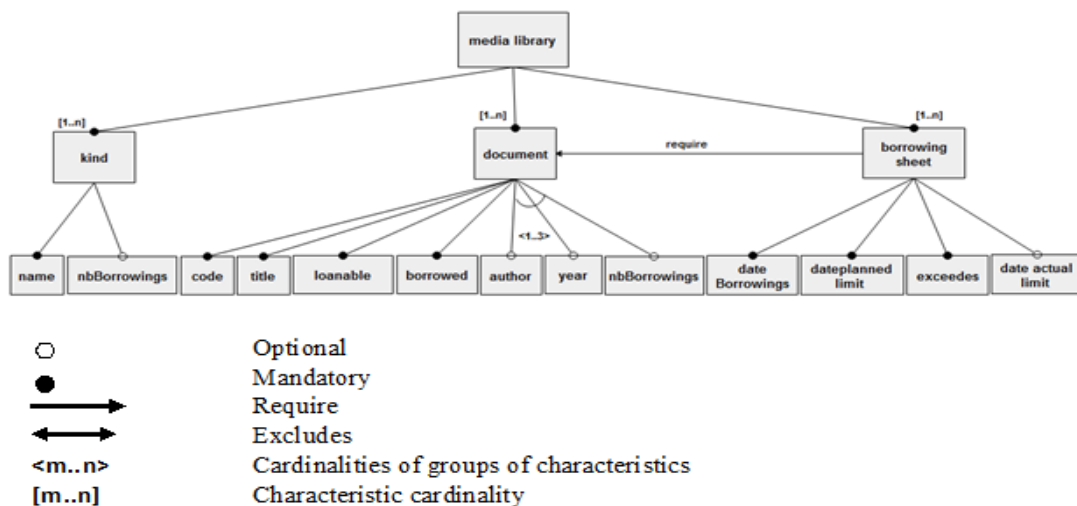


Figure-6. The feature model FODA relating to the class diagram in figure.5.

Source: Khalfallah, et al. [8]

In the first step, we began to work on modelling and metamodeling by UML using the MDE approach [12] of OMG: MDA [13] to solve the problem of variability modelling in the database schema. In MDE, there are three principal abstraction models [14] that we will work on, which are CIM [15] PIM and PSM. We have started working on the CIM level to create a valid general metamodel to model variability for any information system. Then, we refined modelling to manage variability in a lower level (PIM). Here, we tried to pass from a source model to a target model that manages variability. To achieve this model's transformation [14]; [16]; [17] we will use strategies and rules that will be defined along this work. So, we choose to use an endogenous and horizontal transformation [18] to assure the generation of the target model M<sub>b</sub>. This choice is explained for two reasons. First, we are going to put as a hypothesis at the beginning that the initial model M<sub>a</sub> and the target model M<sub>b</sub> are stemming from the same metamodel MM<sub>a</sub>. Thus the most adequate is to choose the use of an endogenous transformation. Second, we choose a horizontal transformation because the transformation of the PIM of the M<sub>a</sub> to the PIM of the M<sub>b</sub> is made by restructuration and normalization, so, there is no passage between the abstraction levels. Finally, we succeeded in generating a target metamodel and a target model where there is variability management. Moreover, we succeeded in managing the variability criterion automatically within the database schema via parameter tables that contain optional features of the main tables of the database schema. Figure 7 illustrates the process of models transformation proposed by our approach on the three abstraction levels which are

M2 where we find the metamodel of the class diagram, M1 where we find the model and M0 where we find the coding.

In a second step, we switched to the variability implementation through the formalization of rules using the ATL language and the coding of the grammar of metamodel using Xtext on Eclipse.

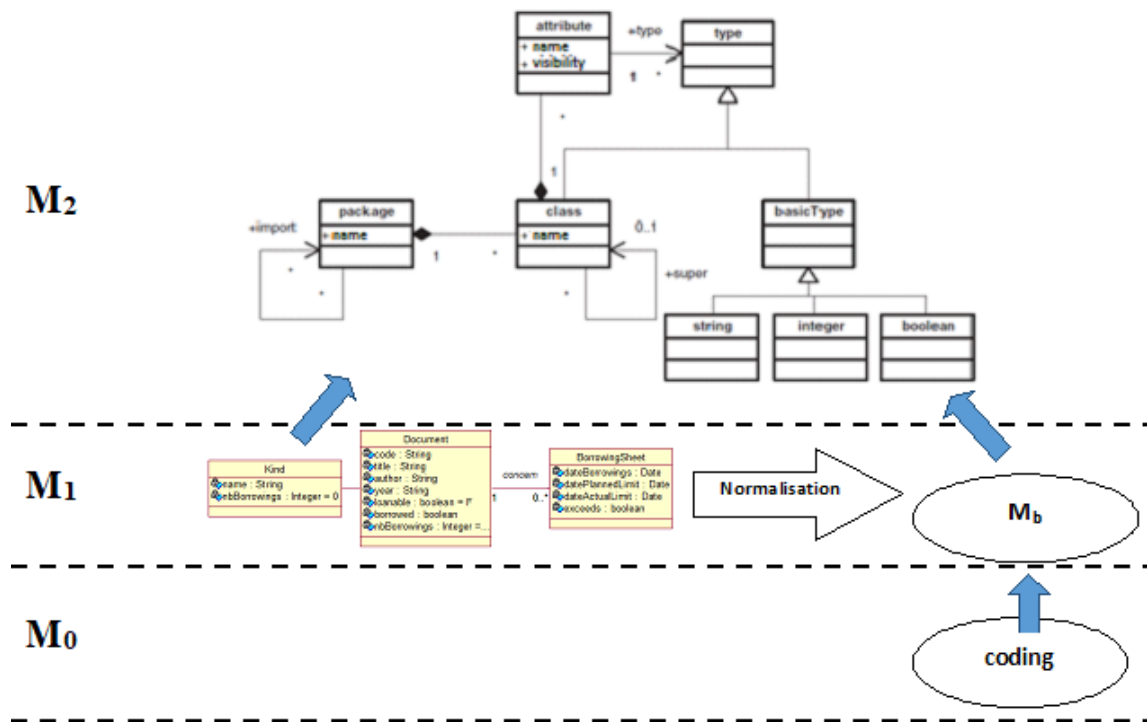


Figure-7. The abstraction levels of the proposed class diagram [8].

Source: Khalfallah, et al. [8]

### 3.2. Transformation Strategies

For the model generation we have chosen to work with the following strategies:

- The matching strategy defines the logic of comparison between the model elements. They depend essentially on the source language. It consists in creating a certified true copy of the elements of source model in the target model.
- The translation strategy defines a strategy to modify the element source or creating new elements in the target model to answer to the particular needs of the user.

These strategies can be executed in a total way if there is a complete conformity between the two models elements or in a partial way by keeping or modifying the starting elements and/or by creating a new element sometimes having properties common to sources elements.

### 3.3. Transformation Rules

In this section, we used human language to define our transformation rules to allow transformation from the PIM of  $M_a$  to that of  $M_b$ . Many criteria are possible to classify the transformation rules. So to classify ours, we considered the type of the elements to which they are applied. We distinguish rules concerning the compound elements (packages, classes, etc.) from those who concern the primitive elements (attributes, parameters, etc.). Then, we detailed more our rules classification, so each category will be divided into two subcategories which are:

- The first subcategory groups the matching rules which allow creating the relations of correspondence between the corresponding elements between both models.
- The second subcategory groups the rules of translation which express the transformation or the creation of elements in the target model which have no correspondents in the source model.

First, for matching rules [8] their behaviour differs depending on the diagram element to which they are applied. As a consequence, there are the matching rules between the classes which make it possible to check the correspondence in a total way between the latter in order to create identical classes stereotyped <<normal>> in the target models. For matching rules between attributes, operations, and associations, they work in the same way as the matching rules between classes. There is always a comparison between the elements treated and then the creation of identical elements in the target model. The only difference is in the nature of the elements treated but not in the behavior. Second, for translation rules [8] they aim to create new classes stereotyped <<type>> to model variability in the target class diagram. It applies according to the localization of the variability within the diagram. If the variability is in the attributes then we generate for each class stereotyped <<normal>> of the target diagram, a class stereotyped <<type>>. The latter has as attributes the optional attributes of the corresponding class stereotyped <<normal>>. Otherwise, if the variability resides at the level of the classes, then we must first identify the classes stereotyped <<normal>> of the target diagram having optional classes stereotyped <<normal>> in order to generate for them classes stereotyped <<type>> which have as attributes the names of the optional classes. Third, we can also use the translation rules to create the class <<administrator>> in the target model. Its attributes are obtained by the translation of the identifiers of classes stereotyped <<type>>. In addition, its associations are equal to the number of classes stereotyped <<type>> as they will be linked directly to the class <<administrator>>. In fact, every transformed diagram must include this class to ensure the configuration between different parameter tables generated by normalization of classes <<type>> such as the management of variability is assured. Indeed, the Class <<administrator>> will play the role of an orchestra conductor.

### 3.4. Results Obtained

Finally, we have succeeded in proposing a generic approach to model variability from an abstract level to a low level. Indeed, we have succeeded in generating a model  $M_b$  which models the variability of our SPL after executing a series of transformation rules that guarantee the reliability of the variability management. The model  $M_b$  ensures the modelling of the variability through the classes stereotyped <<type>> and the class stereotyped <<administrator>>. Figure 8 presents the variable transformed class diagram that ensures modelling of variability at the PIM level through a model transformation. This model is generated after executing the transformation rules related to variability within attributes.

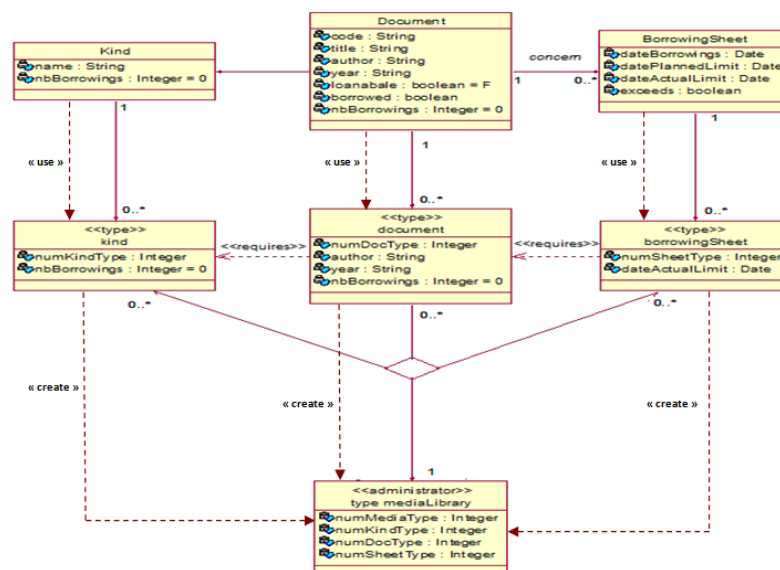


Figure-8. Model  $M_b$  generated after the execution of the rules.

Source: Khalfallah, et al. [8]

### 3.5. Proposed Metamodel

Remember that initially in section 3.1 we put a hypothesis that our type of transformation used in this approach is endogenous. As we have assumed that the two models  $M_a$  and  $M_b$  are derived from the same metamodel  $MM_a$ . But at an advanced stage in our approach and after the definition and application of our transformation rules, we have succeeded in proposing a new metamodel [8] that manages variability in databases at an abstract level (CIM) and that can be respected for the modelling of any other variable model at a lower level. Indeed, our approach extends the original metamodel (figure 7), used at the beginning of the transformation, to produce a new  $MM_b$  metamodel [8] related to the model  $M_b$ . In fact, this extended metamodel supports the concepts of variability used during our transformation approach. It is illustrated in figure 9. Therefore, this distorts our starting hypothesis as our transformation can be now exogenous. As from the rules of transformations, we were able to generate a generic metamodel  $MM_b$ , available to model the variability within any type of target model.

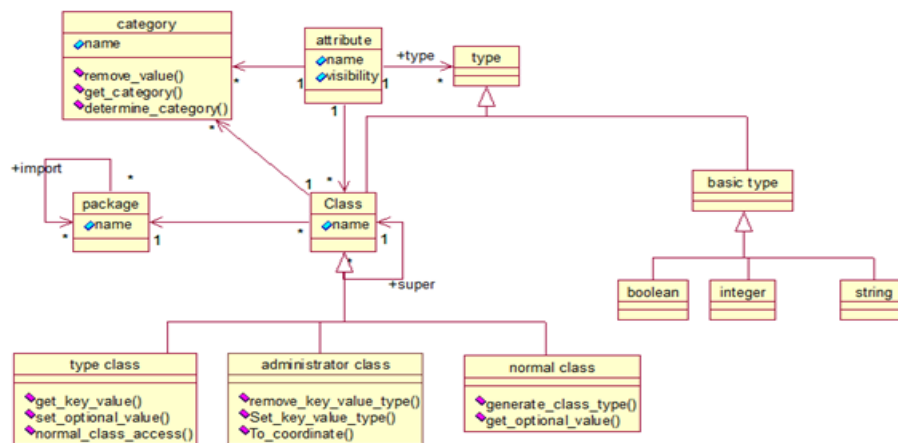


Figure-9. Metamodel related to the  $M_b$  model generated after rules definition.

Source: Khalfallah, et al. [8]

### 4. ILLUSTRATIVE CASE STUDY

In this section, we resume working with the global class diagram of the case study "Media library" existing in figure 3 because here we will work on the variability within the classes. Indeed, we are going to apply the rules relative to the transformation of the model when the variability exists at the level of the classes. Thus, we will treat the case study from a different point of view, which makes it possible to verify the validity, the feasibility and the genericity of another side of our approach. To apply our approach we also need the FODA model (figure 10) as input with the class diagram. This model determines the category of optional or mandatory classes existing in the class diagram.

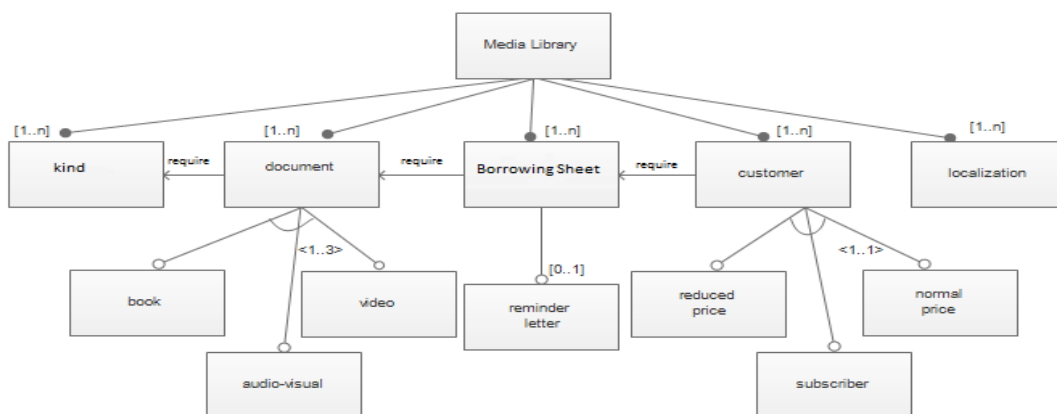


Figure-10. FODA model.

Source: Khalfallah, et al. [8]



Finally, after the execution of the appropriate rules, we obtained the class diagram result in figure 11.

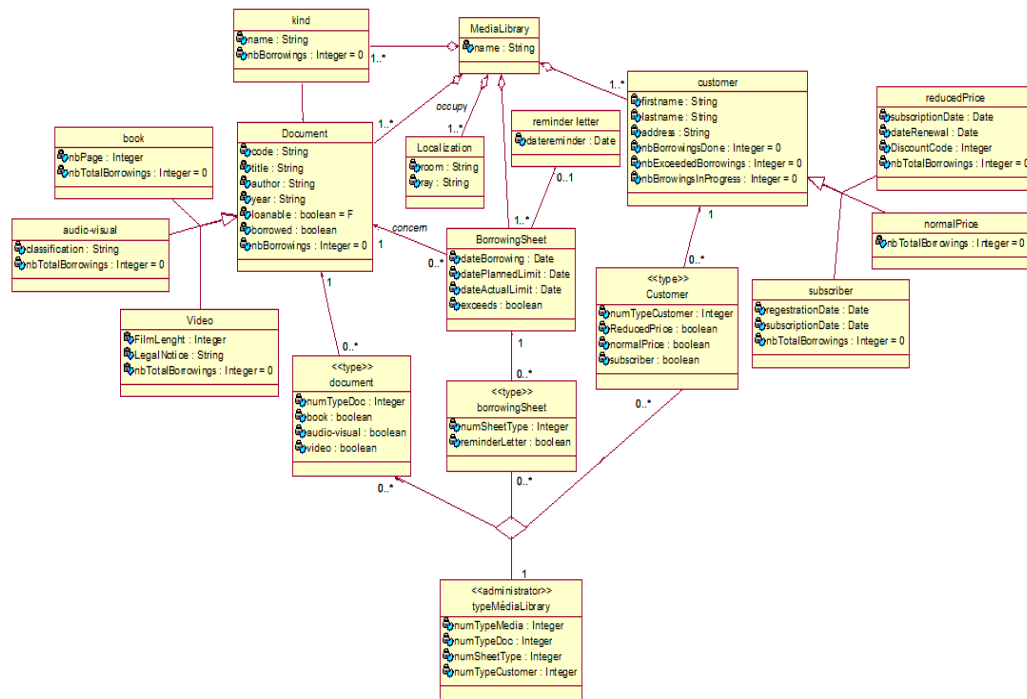


Figure-11. Target model Mb, generated after rules execution.

Source: Khalfallah, et al. [8]

We Explain in Detail the Application of Our Approach to the Case Study:

- 1- For every class of the source class diagram where there is variability management according to the FODA model (document, borrowing file, and customer) we generate a class stereotyped <<type>> in the target class diagram.
- 2- The attributes of the classes stereotyped <<type>> are the names of the classes of the optional type in the FODA model, which have as father a class identified in step 1 such as: {"Audiovisual", "Book" and "Video"}, {"Reminder letter"}, {"Subscriber", "Reduced price" and "normal price"}. Hence, the attributes of the classes <<type>> are:
  - For the class <<type>> Document its attributes are book, audio-visual and video.
  - For the class <<type>> Borrowing sheet its attribute is reminder letter.
  - For the class <<type>> Customer its attributes are reduced price, normal price, and subscriber.
- 3- Create a class stereotyped <<administrator>> to manage all classes stereotyped <<type>> at the same time. This ensures the management of variability within the database schema.

## 5. TOOL SUPPORT

In this section, we worked on the PSM model by proposing and developing a tool support to manage the variability in the database schema. Our tool allows the automation of our approach presented in section 3. This tool will generate a tailored database schema automatically which meets customer needs. It will complete our contribution and test it with a case study. We have chosen to develop a web site for the management of variability in database schemas. Our site offers an open online space for all users to create their own variable schema suited to their requirements for any database integrated into a SPL. This tool will allow online modelling of class diagrams and FODA models. These can be entered by users, and even if the FODA model is not entered by the user, our tool will automatically generate it from the class diagram entered at the beginning. Then, it will transform the entered class diagrams into a target class diagram modelling the variability. Next, it will normalize the transformed class diagram to automatically generate its variable and relational database schema. In addition, it supports the two types of variability explained earlier (within classes and within attributes). Indeed, it is a simple tool to use, practical and has clear menus readable for the user. Technologies used to create this tool are HTML 5, PHP 5, Bootstrap 3, Wamp server and SQL. Finally, our tool support ensures well the automatic modelling of user requirements at the conceptual level through the transformed and variable class diagram and at the logical level through the relational and variable database schema.

## 6. APPROACH VALIDATION

### 6.1. Qualitative Validation

In this section, we want to accentuate the validity and reliability of our approach to highlight our contribution. This development is done through an evaluation of the variable database schema generated by our approach compared with those generated by the existing approaches. The attributes of the facet quality of conceptual models and of database schema of the comparison framework [7] are used for the evaluation of the variable database schema. In Table 1, the result of this evaluation is obtained according to the quality factors of a database schema. We find in the first line of the table the qualitative factors which are: completeness, complexity of the schema, data integrity, flexibility, comprehension, and implementation. And in the first column, we find the approaches chosen to make the comparison. In fact, these approaches are those used in the comparison framework [7].

From Table 1, the global schema approach, the views approach, the frameworks approach, the variable schema approach and our approach ensure completeness in the database schema. Note that only the variable schema approach, the virtual decomposition approach, the Approaches based on extended UML diagrams and our approach effectively manage the complexity of the database schema. In addition to that, the most approaches do not properly treat data integrity when managing variability. Unlike our approach where we worked on the integrity as well as the variability within the database schema. As in our approach, we considered, when managing the variability, the integrity constraints in the database and the dependencies and the cardinalities in the feature model. Moreover, our approach promotes flexibility within the database schema, like the tailored DBMS approach and unlike other approaches that have neglected this aspect. Thus, our approach is one of the approaches that are easy to understand by the user. Finally, our approach is also implementable and equipped with a tool support. This factor is absent in other approaches or suffering from some insufficiencies.

Table-1. Evaluation of our variable schema by contribution to existing approaches.

Quality factors Approaches	Completeness	Complexity of the schema	Data integrity	Flexibility	Comprehension	Implementation
Global schema	++	-	-	-	--	-
Views	++	--	+/-	-	-	--
Frameworks	++	+	-	-	+/-	+/-
Variable schema	++	++	++	+/-	+	+/-
PDO	-	-	-	-	+	++
Tailored DBMS	-	+	+	++	++	+/-
Virtual decomposition	-	++	--	+	++	+
Approaches based on extended UML diagrams	+/-	++	+/-	++	++	--
Database evolution in SPL	-	+/-	++	-	++	++
Variable data model	--	-	--	++	-	--
<b>Our approach</b>	++	++	++	++	++	++

++ very good, + good, - unwieldy, -- very unwieldy

6.2. Quantitative Validation

After a qualitative evaluation of the database schema generated by our approach, we will proceed in this section to a quantitative evaluation based on numerical metrics. We opt for the choice of the most relevant metrics when running the database schema by DBMS, in order to highlight the contributions of our approach. The metrics chosen for our validation are: the CPU execution time, the estimated input/output cost, the operator estimated cost and the estimated used size. The choice of these metrics is due to the fact that the execution of a request is almost to execute its various operators arranged in an execution plan. This validation requires, first of all, an implementation of the various database schemas generated by the different existing approaches. Then, we extracted the necessary metrics to compare them with those obtained from the execution of the other schemas generated by the other existing approaches. We choose the two closest approaches to our approach to implement their schemas which are: the view approach and the variable schema approach. To make this comparison, we worked with Microsoft SQL server management studio 2008 on a local workstation whose processor is Intel® Core™ i7-5500U and has a frequency of 2.40 GHZ. In addition, it has a RAM of 8 GB. Its operating system is Windows 10, 64 bits. As the Microsoft SQL Server management studio 2008 environment makes it possible to return these metrics each time a query is executed by displaying its estimated execution plan. In Figure 12, we find an overview of the execution plan estimated by SQL Server when executing a query.

Indeed, for each feature, a selection request is executed and the metrics relating to this query are extracted for each schema. Tables 2, 3 and 4 illustrate the metrics for each run of the same query for each different schema.

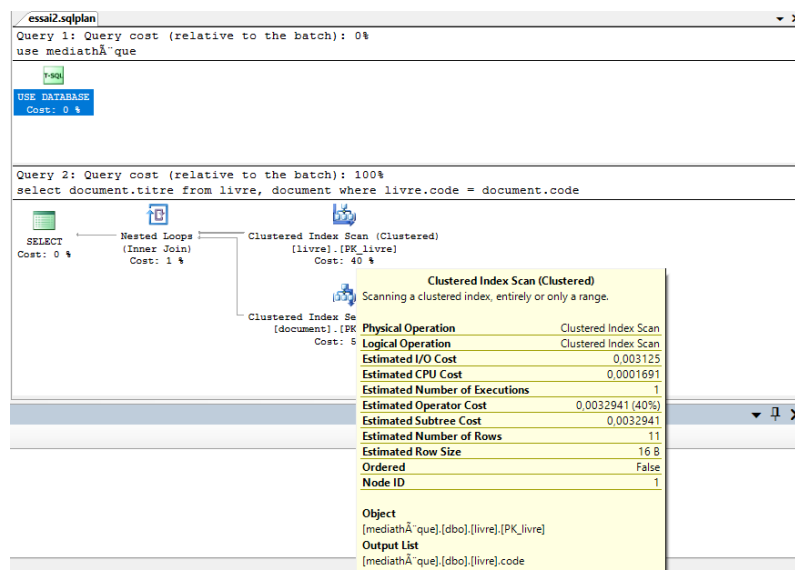


Figure-12. Example of an execution plan estimated by SQL Server when executing a query. Source: Microsoft SQL Server management studio 2008

Table-2. The metrics for the book feature selection query.

Metrics Approaches	CPU execution time	Estimated input/output cost	Operator estimated cost	Estimated used size
Our approach	0,000179	0.003125	0,003304	45B
View approach	0.0003272	0.003125	0.0081582	52B
Variable schema Approach	0.0003272	0.003125	0.0081582	52B

Source: The results from the book feature selection query execution

Table-3. The metrics for the subscriber feature selection query.

Approaches	CPU execution time	Estimated input/output cost	Operator estimated cost	Estimated used size
Our approach	0,000179	0,003125	0,003304	70B
View approach	0,0003217	0,003125	0,0073622	77B
Variable schema approach	0,0003217	0,003125	0,0073622	77B

Source: The results from the subscriber feature selection query execution

Table-4. The metrics for the reminder letter feature selection query.

Approaches	CPU execution time	Estimated input/output cost	Operator estimated cost	Estimated used size
Our approach	0,000168	0,003125	0,003293	25B
View approach	0,0003261	0,003125	0,007999	38B
Variable schema approach	0,0003261	0,003125	0,007999	38B

Source: The results from the reminder letter feature selection query execution

We will use the existing values in tables 2, 3 and 4 for the tracing of figures 13, 14, 15, 16, 17 and 18.

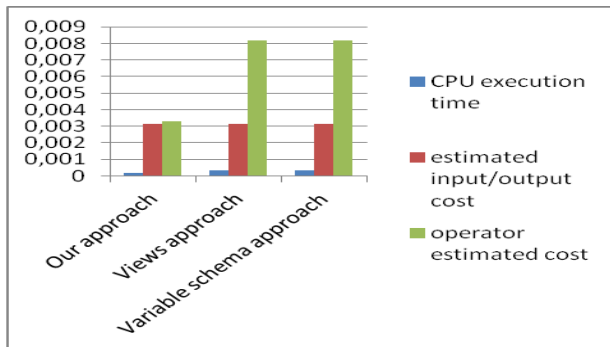


Figure.13. Illustrative graph of metrics: CPU execution time, estimated input/output cost and operator estimated cost relating to the select request of the book feature.

Source: The results from the book feature selection query execution

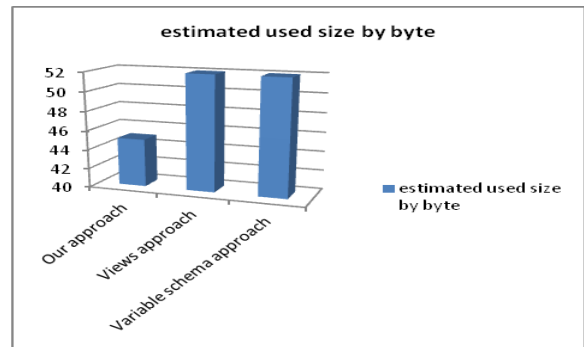


Figure.14. Illustrative graph of the estimated used size metric relating to the select request of the book feature.

Source: The results from the book feature selection query execution

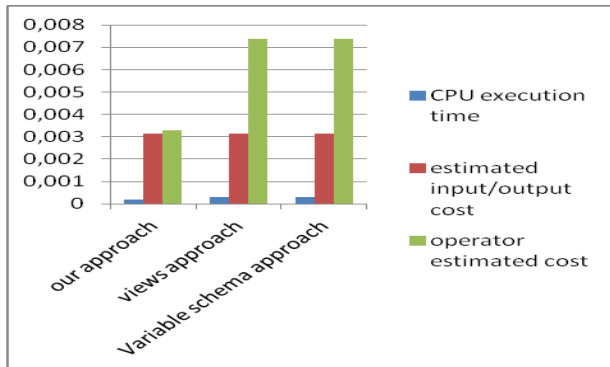


Figure.15. Illustrative graph of metrics: CPU execution time, estimated input/output cost and operator estimated cost relating to the select request of the subscriber feature.

Source: The results from the subscriber feature selection query execution

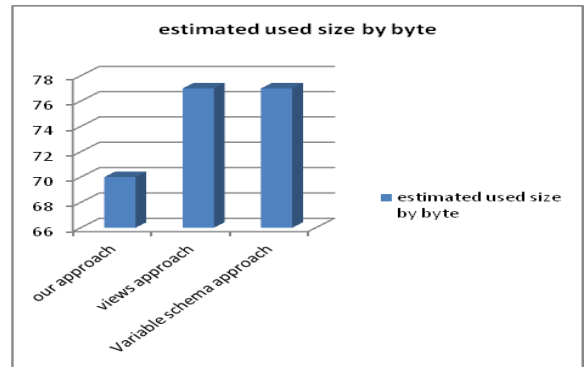


Figure.16. Illustrative graph of the estimated used size metric relating to the select request of the subscriber feature.

Source: The results from the subscriber feature selection query execution

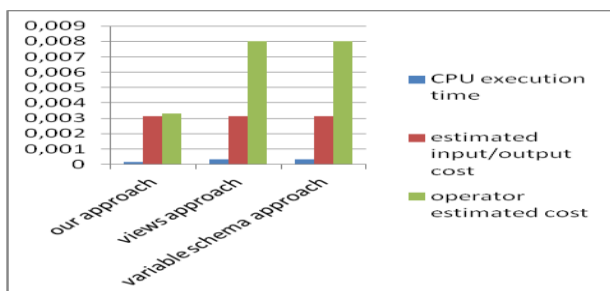


Figure.17. Illustrative graph of metrics: CPU execution time, estimated input/output cost and operator estimated cost relating to the select request of the reminder letter feature.

Source: The results from the reminder letter feature selection query execution

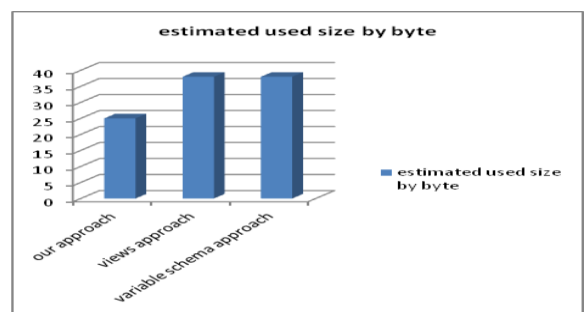


Figure.18. Illustrative graph of the estimated used size metric relating to the select request of the reminder letter feature.

Source: The results from the reminder letter feature selection query execution

Figures 13, 15 and 17 are corresponding to the selection of the optional features book, subscriber, and reminder letter. We observe that the estimated input/output cost is always the same for the different approaches whatever the selected feature. On the other hand, for the CPU execution time and the estimated cost of the operator, the values relating to our approach are much better than those generated by the two other approaches whatever the selected feature. This shows the good performance of our approach compared to other existing ones, as it allows a significant time saving during the execution of queries.

In Figures 14, 16 and 18, we note that the view approach and the variable schema approach give almost the same values of the estimated used size whatever the selected feature. On the other hand, we note that our approach uses a much smaller size whatever the selected feature. So not only our approach saves time but also saves space. It should be mentioned that the number of tuples in the different tables in common between the different relational schemas of the approaches used is the same.

## 7. RELATED WORK

Many approaches related to managing variability in the database schema are proposed in the literature. Indeed, we have chosen the closest approaches to us and which have the same research axis and the same goal. Parsons [19] presented the global schema approach. This approach has presented a lot of disadvantages in terms of schema complexity, data integrity, flexibility, comprehension, and implementation. In addition, he only worked on the logical and physical schema at PSM level. The goal in developing the global schema is to represent the requirements of different information to combine and resolve inconsistencies between several partial schemas. Bolchini, et al. [20] presented the view approach. This approach generates views in addition to the global schema that acts as a schema variant for the client. This approach inherits the disadvantages of the global schema approach. Only the integrity of the data is much improved in this approach. Schäler, et al. [15] present an approach for the construction of information system variants with database schemas adapted to the features desired by the client. This approach treats the client side and the database side of a SPL, in the same way, using a variable database schema. Thus, they have shown how to model a variable schema by using the superimposition composition mechanism that generates a variant of the particular information system, having an adapted schema, automatically. The strongest advantage of the variable schema is improving data integrity, while the modelling and the implementation of the database schema are much more complex. This approach is considered reliable only to manage variability in the logical and physical database schema at the PSM level. So many proposals are only interested in solving the problem of variability at the PSM level while neglecting the problem of variability in the CIM and PIM levels. Rare are the proposals that are interested in this problem, for example, Siegmund, et al. [5] have tried to work on the conceptual level using the virtual decomposition technique. The aim of this approach is to propose an adequate separation of the participants in the conceptual modelling. This approach is based on the virtual decomposition of the schema in terms of features. In this decomposition, the schema elements are not physically separated into different files, but only annotated. Unfortunately, this approach treated only the PIM level. In addition, it is weak at the level of the model's completeness and at the level of data integrity. There is also Khedri and Khosravi [21]; Khedri and Khosravi [22] used the delta-oriented programming first to manage variability in both logical and physical schemas at the PSM level, and then to model variability in the conceptual schema at the PIM level. It is an evolutive approach in contrast to the virtual decomposition approach [5]. There is even a proposal for a metamodel at the CIM level. This is an implementable approach but suffers from several problems of completeness, flexibility, data integrity and complexity of the schema. There are also approaches based on extended UML diagrams [23]; [24]; [25]; [26]; [27] that have been interested in proposing metamodels and variable conceptual models for databases. But their proposed models suffer from many insufficiencies and imprecision. In addition, these approaches can't be implemented to be tested in real-world scenarios. Also, they are not flexible enough to ensure the evolution of the database in a product line throughout its life cycle. Moreover,

Herrmann, et al. [28] proposed a DAVE tool for managing the evolution of the database in a product line. Indeed, they tried to solve the weaving problem to ensure this evolution by adopting the technique of schemas versioning to create a global, correct and effective script. But unfortunately DAVE solves the weaving problem only at the logical level. The physical optimization of a global evolution scenario couldn't be addressed as there is no evaluation containing, for example, performance measures. So, DAVE simply offers tool support for manual evolution and database migration. An automated and controlled process for the evolution of the database in software product lines is still missing. Finally, Abo and Troyer [29] discussed the need to model the data variability as well as the variability of applications when designing software product lines. Indeed, they proposed an approach for modelling data variability as an integral part of the entire data intensive software product line at the conceptual level. They proposed a variable data model using the EER [30]. To carry out this modelling, they have extended the Feature Assembly Modelling (FAM) technique defined in Abo, et al. [31] by a persistency perspective in which the features are defined from the point of view of their need for the manipulation of the persistency data in the SPL. This approach simplifies the process of access to the data, which simplifies the queries and avoids the dummy values in the requests of insertion and update. It optimizes the storage space. Thus, the variable data model proposed by this approach can be expressed in a similar manner with other data modelling techniques such as UML or ORM. But unfortunately this approach suffers from insufficiencies such as the lack of a support tool for modelling the persistency perspective and to support the mapping to the variable data model. In addition, this approach is not yet applicable or validated on real industrial scenarios. In the literature, we found a lack of tool support for most of the proposed methods already cited before.

## 8. CONCLUSION AND FUTURE WORK

We started work with an extensive bibliographic study, which allowed us to develop a comparison framework. This framework has helped us to better understand our field of research, to analyse and compare existing approaches that deal with this field and to extract existing limitations and problems. This extraction allows us to pinpoint our problematic, which makes it possible to better solve it to meet the insufficiencies already found in the previous steps. All this helped us a lot in determining our proposed solution which is a proposed approach for the modelling and the implementation of the variability in the schema of a database. Indeed, our contribution was the development of a variability management process for the database schema using the MDE approach based on the models' transformation respecting a series of transformation rules that we defined and formalized using ATL language. The execution of these rules generates a generic and variable data model from basic input models and a FODA variability model. As a consequence, we succeeded in obtaining a data model that assures the variability management at PIM level which was much neglected in others research done to manage variability as most of them work only on the PSM level. And we produced a generic metamodel which assures variability in any system, at CIM level. We implemented the model grammar with Xtext language. Then, we illustrated the application of this proposed technique by a case study "Media library". Finally, we proposed a tool support to apply and automate our approach. Indeed, our tool support offers two-level of automation. The first is in the engineering domain through a passage between the different data models and between different levels of abstraction. The second is in the application engineering through the passage from a data model to a particular configuration of its relational database schema.

**Funding:** This study received no specific financial support.

**Competing Interests:** The authors declare that they have no competing interests.

**Contributors/Acknowledgement:** All authors contributed equally to the conception and design of the study.

## REFERENCES

- [1] S. Ouali, N. Kraiem, and H. Ben ghezala, "Security requirement engineering for intentional software product line," *Global Journal of Computer Science and Technology*, vol. 12, pp. 27- 30, 2012. [View at Google Scholar](#)
- [2] S. Ouali, N. Kraiem, and G. H. Ben, "Intentional software product line using model driven engineering," *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, pp. 402-410, 2015. [View at Google Scholar](#)
- [3] C. Dyreson, R. T. Snodgrass, F. Currim, S. Currim, and S. Joshi, "Weaving temporal and reliability aspects into a schema tapestry," *Data & Knowledge Engineering*, vol. 63, pp. 752-773, 2007. [View at Google Scholar](#) | [View at Publisher](#)
- [4] W. Mahnke, "Towards a modular, object-relational schema design," presented at the The 9th Doctoral Consortium of the 14th International Conference on Advanced Information Systems Engineering, CAiSE'2002, Toronto, Canada, 2002.
- [5] N. Siegmund, C. Kästner, M. Rosenmüller, F. Heidenreich, S. Apel, and G. Saake, "Bridging the gap between variability in client application and database schema," presented at the 13th GI Symposium Database Systems for Business, Technology and Web, 2009.
- [6] P. Ye, X. Peng, Y. Xue, and S. Jarzabek, "A case study of variation mechanism in an industrial product line," presented at the International Conference on Software Reus, Berlin, 2009.
- [7] N. Khalfallah, S. Ouali, and N. Kraiem, "A proposal for a variability management framework," presented at the The 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Tunisia, Hammamet, 2016.
- [8] N. Khalfallah, S. Ouali, and N. Kraiem, "Managing variability in database context using an MDE approach," presented at the The 4th Int. Conf. on Control Engineering & Information Technology (CEIT), Hammamet, Tunisia, 2016.
- [9] S. Ouali, N. Kraiem, and H. B. Ghezala, "Framework for evolving software product line," *International Journal of Software Engineering & Applications*, vol. 2, pp. 34-51, 2011. [View at Google Scholar](#) | [View at Publisher](#)
- [10] C. Rolland, "A comprehensive view of process engineering," presented at the International Conference on Advanced Information System Engineering, Italie, 1998.
- [11] C. Rolland, C. Souveyet, and C. B. Achour, "Guiding goal modeling using scenarios," *IEEE Transactions on Software Engineering*, vol. 24, pp. 1055-1071, 1998. [View at Google Scholar](#) | [View at Publisher](#)
- [12] R. Bouhaouel, N. Kraïem, C. Salinesi, Z. Al-khanjari, S. Ouali, and T. Esprit, "Framework to compare the model generation methods," *Asian Journal of Scientific Research*, vol. 8, pp. 54-66, 2015. [View at Google Scholar](#) | [View at Publisher](#)
- [13] O. Topçu, U. Durak, H. Oguztuzun, and L. Yilmaz, *Distributed simulation: A model driven engineering approach: Part of the series simulation foundations, methods and applications*: Springer InternationalPublishing Switzerland, 2016.
- [14] A. Adil, "Formalization by an IDM approach of model composition in the VUML profile," PhD Thesis, Research Institute in Computer Science of Toulouse, France, 2009.
- [15] M. Schäler, T. Leich, M. Rosenmüller, and G. Saak, "Building information system variants with tailored database schemas using features," presented at the International Conference on Advanced Information Systems Engineering, 2012.
- [16] J. Bézivin, F. Jouault, P. Rosenthal, and P. Valduriez, "The AMMA platform support for modelling in the large and modelling in the small," Research Report, LINA, 2003.
- [17] J. Garcia-Molina, A. Moreira, and G. Rossi, "UML and model engineering," *European Journal for the Informatics Professional*, vol. 5, pp. 3-5, 2004.
- [18] B. Combemale, "Metamodeling approach for simulation and model verification - application to process engineering," PhD Thesis, National Polytechnic Institute of Toulouse, France, 2008.
- [19] J. Parsons, "Effects of local versus global schema diagrams on verification and communication in conceptual data modeling," *Journal of Management Information Systems*, vol. 19, pp. 155-183, 2002. [View at Google Scholar](#) | [View at Publisher](#)
- [20] C. Bolchini, E. Quintarelli, and R. Rossato, "Relational data tailoring through view composition," presented at the International Conference on Conceptual Modelling, 2007.

- [21] N. Khedri and R. Khosravi, "Handling database schema variability in software product lines, school of electrical and computer engineering," presented at the 20th Asia-Pacific Software Engineering Conference (APSEC), 2013.
- [22] N. Khedri and R. Khosravi, "Incremental variability management in conceptual data models of software product lines," presented at the 22nd Asia-Pacific Software Engineering Conference (APSEC), India, 2015.
- [23] M. Clauß and I. Jena, "Modelling variability with UML," presented at the GCSE 2001 Young Researchers Workshop, 2001.
- [24] M. Clauß, "Generic modelling using uml extensions for variability," presented at the Workshop on Domain Specific Visual Languages at OOPSLA, 2001.
- [25] T. Ziadi, "Manipulation of product lines in UML," PhD Thesis, University of Rennes, France, 2004.
- [26] H. Gomaa, *Designing software product lines with UML: From use cases to pattern-based software architectures*. Canada, USA: Addison-Wesley Pearson Education, 2005.
- [27] B. Korherr and B. List, "A UML 2 profile for variability models and their dependency to business processes," presented at the Database and Expert Systems Applications, DEXA Workshops, Regensburg, Deutschland, 2007.
- [28] K. Herrmann, J. Reimann, H. Voigt, B. Demuth, S. Fromm, R. Stelzmann, and W. Lehner, "Database evolution for software product lines," presented at the 4th International Conference on Data Management Technologies and Applications (DATA 2015), Colmar, Alsace, France, 2015.
- [29] Z. L. Abo and D. O. Troyer, "Towards modelling data variability in software product lines," presented at the Enterprise, Business-Process and Information Systems Modelling: 12th International Conference, BPMDS 2011, and 16th International Conference, EMMSAD EMMSAD 2011, held at CAiSE 2011, London, UK, Proceedings, 2011.
- [30] B. Thalheim, "Extended entity-relationship model," *Encyclopedia of Database Systems*, vol. 1, pp. 1083-1091, 2009.
- [31] Z. L. Abo, F. Kleinermann, and D. O. Troyer, "Feature assembly: A new feature modelling technique," presented at the 29th International Conference on Conceptual Modelling, Lecture Notes in Computer Science, 2010.

*Views and opinions expressed in this article are the views and opinions of the authors, Journal of Asian Scientific Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.*