Architecture-Driven Modernization: A Systematic Review

Lab, University, Address

Abstract. Background: Architecture-Driven Modernization (ADM) has become a popular approach for modernizing legacy systems by means of Model-Driven Development (MDD). It uses standard and language independent metamodels, such as Knowledge Discovery Metamodel (KDM). The whole process involves reverse engineering a system into a KDM instance, refactoring it obtaining a modernized KDM and generating the modernized system. Although the interest in ADM has grown in the last years, to the best of our knowledge, there is neither published systematic review nor survey which presents the current state of the art and research directions in this field. Therefore, we claim that there is a need for evidence-based body of knowledge about different aspects of the adoption of ADM. **Objective:** Presenting a systematic review that exposes the current state of the art and research directions in the ADM field. **Method:** A systematic review on ADM was performed using Kitchenham's guidelines which consist of three main phases: (*i*) planning, (*ii*) conducting and (*iii*) reporting.**Results and conclusion:** Classification schemes have been defined and the 30 primary studies have been selected and classified on the basis of focus area, contribution type and research type. Four focus area have been identified. Papers that illustrated process and tools to assist the modernization of legacy system by means of ADM are in a majority. The majority of research type are evaluation research. We claim that the findings of this paper provide interesting insights into different aspects of ADM. Also, we expect that the findings can provide valuable information to readers on what can be expected from applying ADM and its metamodels to modernize legacy systems. Furthermore, the results can provide perception of new research in the modernization area for investigating and defining new tools/process to assist the modernization of legacy systems.

1 Introduction

Software systems are considered legacy when their maintenance costs are raised to undesirable levels but they are still valuable for organizations. Therefore, they can not be discarded because they incorporate a lot of embodied knowledge due to years of maintenance. As these systems still provide significant business value, they must then be modernized so that their maintenance costs can be manageable and they can keep on assisting in the regular daily activities.

In this context, OMG has defined standards in the modernization process, creating the concept of ADM. ADM follows the MDD [1] guidelines and comprises three major steps. Firstly, a reverse engineering is performed starting from the source-code and a model instance Plataform-Specific Model (PSM) is created. Secondly, successive transformations are applied to this model up to reach a good abstraction level in a model called KDM (Knowledge Discovery Metamodel). Upon this model, several modernization, optimizations and modifications can be performed in order to solve problems found in the legacy system. Then, a forward engineering is carried out and the source code of the modernized target system is generated. According to the OMG the most important artifact provided by ADM is the KDM metamodel, which is a multipurpose standard metamodel that represents all aspects of the existing IT (Information Technology) architectures. The idea behind the standard KDM is that the community starts to create parsers from different languages to KDM. As a result everything that takes KDM as input can be considered platform and language-independent. For example, a refactoring catalogue for KDM can be used for refactoring systems implemented in different languages.

In order to present the current state of the art and research directions in this field, we performed a systematic review of ADM and its metamodels. Our motivation to realize a systematic review is to identify areas that have been studied and conducted related to ADM. Although, ADM is relatively a recent approach, OMG claims that it is a standardization that aiming to join two known research field, (*i*) Model-Driven Development and (*ii*)Software Reengineering. Thus, identifying possible gaps that have not been researched and presenting results from literature are valuable from perspective of possible future enhancements and use. Although several studies have been reported in the context of ADM [2–5], to the best of our knowledge none systematic review has been conducted in the field of ADM and its metamodels. As for the fact that various types of research have appeared addressing diversifying focus areas related to the topic of modernization of legacy system by means of ADM, we claim the need for a more systematic investigation of this topic. Also this paper can help researchers in the field of modernization of legacy systems, once it provides an overview of the current state-of-the-art of the ADM. Furthermore, this paper serves as a first step towards more thorough examination of the topics addressed in it with the help of systematic literature reviews.

This paper is structured as follows: In Section 2, describes how the systematic review methodology has been planned, conducted and reported. In Section 4 there are the principle findings and the threats to validity of this study. Concluding remarks are made in Section 6.

2 The Systematic Review

This study was undertaken as a systematic review based on the guidelines proposed by Kitchenham and Brereton [6] which consist of three main phases: (*i*) planning, (*ii*) conducting and (*iii*) reporting. The following sections present details on how each phase was conducted.

2.1 Planning the Systematic Review

In this phase we defined the review protocol. This protocol contains: (*i*) the research questions, (*ii*) the search strategy, (*iii*) the inclusion and exclusion criteria and (*iv*) the data extraction and synthesis method.

The objective of this review is to find out **presenting the current state of the art and research directions in the ADM field**. In order to achieve such objective we formulated three research questions, as can be seen in Table 1.

Table 1: Research questions

RQ_1:	What are the focus areas most discussed and least discussed in the literature regarding the ADM? Moreover, what types of contributions have been presented so far?
RQ_2:	Given the ADM's standards metamodels, which one has been more used in the literature? In addition, given the identified metamodel, what are the packages most and least used?
RQ_3:	Which research types have been employed into the field herein?

To address $\mathbf{RQ}_1$, all the primary studies picked during the the selection stage were read in order to identify the focus area of each study. Next, we arrange all studies according to the focus area. Whether any kind of disagreement related to the focus area that the article meets, it was marked and was discussed with everyone involved in the review in order to clarify which topic it belongs. With respect to $\mathbf{RQ}_2$, we also read all primary studies and identified which metamodel was used in the study. Finally, with respect to $\mathbf{RQ}_3$, we used and adapted the scheme proposed by Wieringa et al., [7] in order to classify each primary study into a research type. Afterwards, we defined the search string and chose the electronic databases. The search string was created based upon a set of keywords. Figure 1 shows the search string elaborated.

("KDM") **OR** ("Knowledge Discovery Metamodel") **AND** ("Knowledge-Discovery Metamodel") **OR** ("Knowledge-Discovery Meta-model") **OR** ("Knowledge Discovery Meta-model") **OR** ("Architecture Driven Modernization") **OR** ("Architecture-Driven Modernization") **OR** ("Model Driven Modernization") **OR** ("Model-Driven Modernization") **OR** ("Model-driven software modernization") **OR** ("Abstract Syntax Tree Metamodel") **AND** ("ASTM") **OR** ("Structured Metrics Metamodel") **OR** ("SMM")

Fig. 1: Search String.

The search encompassed electronic databases which are deemed as the most relevant scientific sources [6] and therefore likely to contain important primary studies. We used the search string on the following electronic databases: *ACM*, *IEEE XPLORE*, *Scopus*, *Web of Science* and *Engeneering Village*. Note that since the features provided by various databases, as well as the exact syntax of search strings to be applied vary from one database to other, the string given in Figure 1 was actually used to construct a semantically equivalent string specific to each database. Furthermore, no limits were placed on date of publication with a view to not restrict the review study scope. Aimed at keeping track of the selected papers, we used JabRef[1], an open source system for bibliography reference management.

In order to determine which primary studies are relevant to answer our research questions, we applied a set of inclusion and exclusion criteria. In Table 2 are depicted these criteria. Not all of these inclusion criteria must be present for every primary study. However, at least the former, must be present. If all criteria were mandatory, the number of selected techniques would decrease significantly.

Table 2: Inclusion and Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
The primary study presents at least one solution of modernization by means of ADM	Papers which mentioned ADM and its metamodels only in the abstract
Studies that explicitly present an ADM approach	Introductory papers for books and workshops
The primary study presents at least one type of evaluation technique for ADM	The primary study is a short paper.

We elaborated data extraction forms to accurately record the information obtained by the researchers from the primary studies. The form for data extraction provides some standard information, such as (*i*) a brief of the primary study, highlighting where ADM and its metamodels were used, (*ii*) date of data extraction, (*iii*) title, authors, journal, publication details and (*iv*) a list of each conclusion and statement encountered for each question. During the extraction process, the data of each primary study were independently gathered by all reviewers. The review was performed in August, 2013 by three M.Sc., a Ph.D. student and three expert. All the results of the search process are documented in the web material(tinyurl.com/99spmaz). So, it is clear to others how thorough the search was, and how they can find the same documents.

2.2 Conducting the Systematic Review

Firstly we applied the search string given in Figure 1 in the following digital libraries: IEEE, ACM, Scopus, Web of Science and Engeneering Village. An overview of results acquired from these digital libraries is depicted in Table 3. As can be seen in this table, the digital libraries Scopus returned more primary studies than the others (150), ACM, Engeneering Village, Web of Science and IEEE returned 51, 30, 17 and 11, respectively. Possibly, this occurred because Scopus indexes studies of others libraries, such as IEEE and ACM. Summarizing, we obtained 259 primary studies. After performing automatic search, we excluded the duplicate publications. If a primary study was found in more than once, we selected the most recent and detailed version of the paper. Afterwards,

[1] http://jabref.sourceforge.net/

Table 3: Overview of search results.

Digital Libraries	Number
Scopus	150
ACM	51
Engeneering Village	30
Web of Science	17
IEEE	11
Total	259
Candidates	82
Final set	30

we selected the primary studies by reading the titles and abstracts and the application of the inclusion and exclusion criteria. At this stage, we also narrowed down the categories of publications to some extent by excluding non-peer reviewed publications, in order to ensure a level of quality as well as to avoid redundancy in contributions. As a result, we acquired a total of 82 primary studies that were read entirely, so the upshot obtained were 30 studies. A total of 229 studies were excluded either due to their limited relevance or meeting one of the other exclusion criterions.

We applied the classification schemes proposed by Petersen et al. [8] and classified the publications into categories from three perspectives, as follows: (*i*) focus area, (*ii*) contribution type and (*iii*) research type. We chose this classification scheme because it is highly used in secondary study, i.e., papers which describe systematic review [9].Thus, the aforementioned categories were adapted to our systematic study. We identified four focus areas, five contribution types and five research types. Notice that the research types reflects the research approach used in the primary study. We used and adapted the scheme proposed by Wieringa et al [7] herein. The resultant classification schemes are as follows:

1. Focus Areas:
   (a) Approaches to modernize legacy systems to another platform/architecture: This focus area is related to primary studies which describe process, method or approach that uses ADM and its metamodels to modernize legacy systems either to another platform or architecture;
   (b) Business Knowledge Extraction: Describes primaries studies which address process, method or approach to extract business process of a legacy system;
   (c) Extension of ADM's Metamodels: This focus area represents primary studies which report an approach, method or process to extend one of the ADM's metamodels;
   (d) Applicability: This category includes papers that mainly focus on reporting evidence related to applying ADM and its metamodels in practice. In other words, papers which enable researchers and practitioners to get a better understanding and utilization of ADM and its metamodels.
2. Contribution Type:

(a) Tool: Refers to primary studies that focus on providing tools to support the modernization of legacy system by using ADM;
(b) Process: Refers to papers which describe processes to assist the modernization of legacy system by means of ADM;
(c) Model Transformation: Refers papers that describe the use of language transformation such as Query/Views/Transformations (QVT)[2] or ATL Transformation Language (ATL)[3] to realize transformation among the ADM's metamodels;
(d) Metamodel: Describes primary studies which create or extend the ADM's metamodels to deal with a specific problem, for instance, providing a KDM light-weight extension in order to either represent the aspect oriented paradigm or supports a component-oriented decomposition;
(e) Metrics: Describes papers that focus on proposing or applying metrics to effectiveness of ADM and its metamodels.

3. Research Type:
   (a) Validation research: Validation research is conducted in a systematic way and may present any of these: prototypes, math analysis, etc;
   (b) Evaluation research: In contrast to validation research, evaluation research aims at examining a solution that has already been practically applied. It investigates the practical implementation of solution and usually presents results using field studies, experiments, or case studies, etc;
   (c) Conceptual proposal: A conceptual proposal presents an arrangement to perceive things that already exist, in a novel way;
   (d) Experience paper: An experience paper reports on personal experience of the author from one or more real life projects;
   (e) Opinion paper: Opinion papers report on personal opinion of the author on suitability or unsuitability of a specific technique or tool.

2.3 Reporting the Systematic Review

The focus of this section is to present broad overview of research within ADM after conducting the systematic review. Moreover, we used information drawn from this overview to answer the research questions.

Aiming to show our results and also the frequencies of all publication related to ADM a bubble plot was plotted, which is depicted in Figure 2. Figure 2 has three facets: **contribution type**, **focus area** and **research type**. It is worth highlighting that certain primary studies were grouped in more than one category, affecting the frequency count; i.e., the sum of the frequencies shown in each facet can be greater than the total of selected studies presented earlier (30). By observing Figure 2 can be seen that the majority of research papers are specifically dedicated for providing **Process** to assist software engineer during the modernization of legacy system to another platform/architecture. Similarly,

[2] http://www.omg.org/spec/QVT/1.1/
[3] www.eclipse.org/atl/

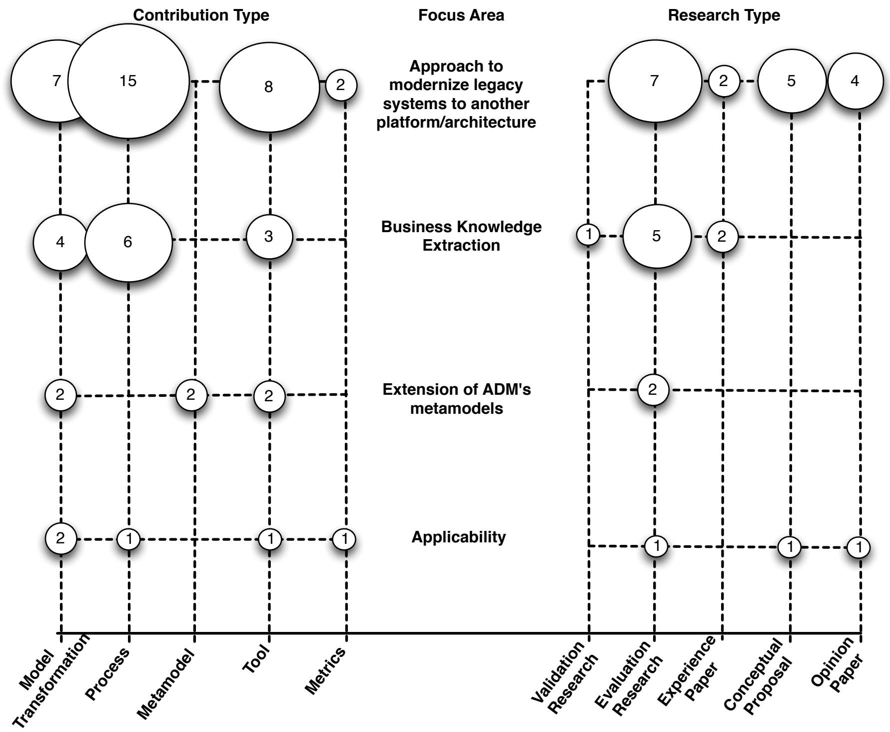
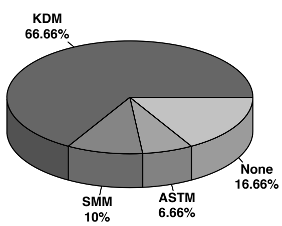


Fig. 2: Map of research focus on ADM and its metamodels.

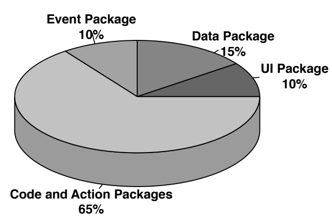
Model transformation and **Tools** are also another field which have been researched. Maybe this happened once the majority of the primary studies found which describe a process, usually propose a set of model transformation and also either semi-automatic tool or a fully-automatic one. In other hand, the contribution type with less studies are **Metamodel** and **Metrics**. Thus, it is argued that primary studies that describe process to assist the modernization of legacy systems by means of ADM, papers which show a set of rules to be applied during model transformation among the ADM's metamodels (KDM, SMM and ASTM), and papers which devise tools to assist ADM's process are evidence clusters. In other words, where there may be scope for more complete literature reviews to be undertaken. Whereas metamodels (i.e., papers that explain how to extend ADM's metamodels) and metrics (i.e., papers which describe how to apply metrics in ADM's metamodel) can be regarded as gaps, thus new or better primary studies are required.

As result of this analysis we answered partially the $\mathbf{RQ}_1$. We answered the types of contributions that have been presented so far in literature related to ADM and its metamodels. Another part of the $\mathbf{RQ}_1$, i.e., (a discussion about the focus area regarding the ADM) has been covered in the following Sections 3.1 through 3.4.

As for answering the first part of $\mathbf{RQ}_2$ we analyzed all primary studies focus on gathering which ADM standard metamodels have more been used in the literature. In Figure 3a is depicted a pie chart wherein we plotted the collected data. As can be seen in this figure, KDM seems to be the metamodel which has been most used in the literature, covering over 66%. A small percentage of primary studies have reported on the use of SMM (10%). While ASTM has been presented by rather small percentage of 6.66%. Finally, we found a total of 16.66% of primary studies that does not show explicitly which metamodel has been used during the process of modernization of a legacy system. In order to answer the second part of $\mathbf{RQ}_2$ we gathered which are the packages most and least used within the KDM according to the identified studies. In Figure 3b it is fairly evident that the packages Code and Action are the most used in the literature (65%). Maybe the reason for this happened are twofold: (*i*) these packages are often used to represent the source-code of system and since most of the studies found use somehow the source-code as input to start the modernization process; and (*ii*) the absence of a complete parser to instantiate all KDM's layers. The third one most used is the Data package (15%). This package is used to represent relational data, such as database, XML, etc. The packages least used are Event and UI, see Figure 3b.



(a)



(b)

Fig. 3: Frequency of ADM's metamodels used in literature and packages most and least used.

To answer the $\mathbf{RQ}_3$ the data on the facet **Research Type** were analyzed, see Figure 2 on the right side. As far as the research type is concerned, **Evaluation Research** are in vast majority, covering 15 primary studies. A small number of publications have reported on **Validation Research** and **Experience Paper**, i.e., 1 and 4, respectively. While **Conceptual Proposal** and **Opinion Paper** have been presented collectively by rather of 11 primary studies.

3 A brief discussion on the focus areas identified within ADM

Here we organized each subsection in a way to describe the focus areas identified in this systematic review.

3.1 Approach to modernize legacy systems to another platform/architecture

Jorge Maratalla et al., propose **GAFEMO** [10] which aims to modernize a legacy systems to service-oriented approach taking advantage of the features provided by gap-analysis techniques. This approach takes as input a legacy system and then creates KDM representation of it. After, a set of rules are applied in this model to create the services. In [11] the authors present a method that combines ADM with program analysis techniques to support analysis across the components of a component-based system. They builds upon the foundations laid out by OMG's KDM to reverse engineer a homogeneous system-wide dependence model from a software system's heterogeneous source and configuration artifacts, and use this model as the basis for the analysis. In [12] the authors propose a modernization approach for the modernization of Data warehouses following the concepts of ADM. The approach automatically performs the following tasks: (*i*) obtain a logical representation of data sources (*ii*) mark this logical representation with MD concepts, and (*iii*) derive a conceptual MD model from the marked model. In [13] is defined an approach which is focused on the analysis of legacy systems to discover and create functionalities to be exposed as services using Web Services. It is based in five steps: (*i*) Database reverse engineering: database schema is reversed and a suitable model is built; (*ii*) First service extraction: based on the structure of the database schema, a first service extraction can be undertaken; (*iii*) PIM generation: is obtained from the PSM representation using a model-to-model transformation, CRUD operations are automatically created; (*iv*) Service discovering: abstract objects are identified in the PIM; (*v*) WSDL (Web Service Description Language) generation: using the PIM, a model-to-model transformation and a WSDL metamodel are generated to expose the services discovered and created in the PIM and the PSM.

In [3, 14] is proposed an approach based on ADM named CloudMIG that aims at supporting SaaS (Software as a Service) providers to semi-automatically migrate legacy software systems to the cloud. It is composed of six major steps: (*i*) Extraction: Includes the extraction of architectural and utilization models of the legacy system, the approach uses KDM; (*ii*) Selection: Select an appropriate CEM- compatible cloud profile candidate; (*iii*) Generation: Produces the target architecture and a mapping model; (*iv*) Adaptation: The adaptation activity enables a reengineer to manually adjust the target architecture; (*v*) Evaluation: Realize static analyses and a runtime simulation of the target architecture; (*vi*) Transformation: The actual transformation of the existing system from the generated target architecture to the aimed cloud environment. In [15] the authors propose an approach that uses ADM which is focused on the analysis of legacy systems to discover and create functionalities to be exposed as services using Web Services.

Pérez-Castillo et al., [16–18] present an approach to modernize legacy systems together with the legacy relational database. This approach recovers the code-to-data linkages and obtains three kinds of models according to the ADM approach: (*i*) The KDM Code Model, which represents the inventory of legacy source code. It has also the points that link the SQL Sentence Models and Database Schema Models. (*ii*) The SQL Sentence Model for modeling a certain SQL query that was embedded in legacy source code. (*iii*) The Database Schema Model, which represents the specific database fragment derived by an SQL Sentence Model. In [4] presents the XIRUP modernization methodology, which proposes a highly iterative process, structured into four phases: preliminary evaluation, understanding, building and migration. This modernization process is feature-driven, component-based, focused on the early elicitation of key information, and relies on a ADM.

In [19] the authors present an approach for the definition of a systematic process for Web Applications (WA) to RIA modernization, by applying ADM principles. The approach presented by the authors consists on generating a RIA client from the legacy WA presentation and navigation layers and its corresponding service-oriented connection layer with the underlying business logic at server side. Boussaidi et al., [20] propose an approach that makes use of the KDM to reconstruct and document software architectural views of the legacy system. They consider an architectural view to be a way of partitioning a system using a specific set of KDM relevant concepts and relations and they propose clustering algorithms that target specific views mainly a layered view that we call horizontal view and a feature based view that we call vertical view. In [1] ADM is used into practice by building a modernization tool to generate metric reports of legacy Oracle Forms applications to assess migration efforts. The authors devised an extractor that generates KDM models from PL-SQL code (PL/SQL-to-KDM) and a metrics report generator for these KDM models.

3.2 Business Knowledge Extraction

Pérez-Castillo et al., [21–24] present an approach to recover business processes from legacy systems. This approach is based on a set of transformation: (*i*) transformation obtains PSM models from each legacy software artifact using a specific metamodel for each artifact; the traditional reverse engineering techniques such as static analysis, dynamic analysis, and formal concept analysis and so on, can be used to extract the needed knowledge; (*ii*) a set of model transformations to obtain a KDM model built from the PSM models at (*i*); (*iii*) a transformation finally obtains the current business process model, this transformation is based on a set of business patterns. In [2] the authors report the results of a family of case studies that were performed to empirically validate this approach.

Normantas and Vasilecas [25] present an approach that facilitates software comprehension by enabling traceability of business rules and business scenarios in software system, i.e., their approach aim to extract business specific knowledge from the knowledge about the existing software system represented within the KDM. Ropero et al., [26] describes a set of rules to transform Mining XML (MXML) metamodel, which is common used to represent the sequence of business activities executed by an enterprise system to KDM.

3.3 Extension of ADM's metamodels

In [27] the authors propose the COMO (Component-Oriented MOdernization) metamodel an KDM's extension, by borrowing recurring concepts from component-based solutions and software architectures, and to support a proper componentization of the system to assist the modernization of legacy systems. In [28] propose an extension to the KDM that aims to represent all the information registered in a MXML model in the KDM model. In [29] the author proposes an extension of the KDM to represent all elements of the Aspect Oriented Paradigm, i.e., aspect, advice, point-cut, can be represented using the KDM. All these three paper have in common is that the authors claimed the impact of these extensions on well-proven and KDM based tools is not problematic since they are performed with the own extension mechanism of the KDM standard.

3.4 Applicability

Pérez-Castillo et al., [5, 23, 28] present how to apply KDM to modernize legacy systems. Also, the authors described each layer of the metamodel KDM, they also presented a set of example of how to use ADM and KDM during the modernization of a legacy systems. The authors claim that the paper enables researchers and practitioners to get a better understanding KDM

4 Discussion

Herein is provided a summary of key findings of this review. Also, here is pinpointed some open issues that still need to be researched in ADM.

4.1 Main Findings and Open Issues

Recent proposals in ADM have focused mainly on providing approaches to modernize legacy system to another platform/architecture resulting in appearance of several proposals, see Section 3.1. However, if we look at overall problem of the integration of modernization into an ADM context, there is still a significant work to be done. Following the ADM approach, models are the main focus for visualizing an executable view of the system and are also used to modernize a system. Therefore, in order to integrate ADM's metamodels into this larger context, the area of discovering knowledge, i.e., parsers, needs more attention along with solution to verification of models. Very few works have been reported on in the literature (e.g., [1, 30]) so far that have addressed parsers to represent instances of KDM, but even these parsers provide limited infrastructure to represent all KDM's layers. Thus, new researches must be conducted to create a complete parser in order to represent all KDM's layer. Besides, the discovering of knowledge are often mostly static in a sense that they are unable to obtain knowledge during the executing of the target legacy system. In [31] the authors presented an approach that dynamically discovery knowledge of the target system during its execution. Nevertheless, this approach lacks support for complex discovery of knowledge once it is just based on the KDM's event metamodel package. Hence further research is required to boost the discovering of knowledge, e.g., use the knowledge dynamically identified combined with other models of KDM as the code and data models in order to acquire more meaningful models, since these models make it possible to consider additional sources of embedded knowledge.

In addition, we observed that there are three main hurdles in need to be more researched so that modernization techniques can be used in the ADM approach in an effective and widespread way. The first hurdle is the present lack of a fully developed idea of "good" KDM style. This is an important issue, for a clear notion of style is a fundamental prerequisite for the use of modernization, enabling software engineers to see where they are heading when modernizing their legacy system with KDM. Fowler et al. [32] advocated a specific notion of style for Object-Oriented Programming (OOP) through a catalog of 22 code smells, compounded by a catalog of 72 refactorings through which those smells can be removed from existing code. A second one - both a cause and a consequence of the first - is the present lack of a KDM equivalent of such catalogues. We assume that the process of modernization by using KDM would equally benefit from KDM specific catalogues of smells and refactorings, helping software engineers to detect situations where the KDM could be improved and guiding them through the corresponding transformation processes. A third hurdle is the absence of tool that supporting refactoring by using the KDM specification in current integrated development environments. The catalogue presented by Fowler et al. [32] provided a basis on which developers could rely to build tool support for object-oriented refactoring: similar catalogue for the KDM specification are likely to bring similar benefits to assist software engineers during the modernization process.

5 Threats to Validity

The main threats to validity identified in the review are described next:

Primary studies selection: Aiming at ensuring an unbiased selection process, we defined research questions in advance and devised inclusion and exclusion criteria we believe are detailed enough to provide an assessment of how the final set of primary studies was obtained. However, we cannot rule out threats from a quality assessment perspective, we simply selected studies without assigning any scores. In addition, we wanted to be as inclusive as possible, thus no limits were placed on date of publication and we avoided imposing many restrictions on primary study selection since we wanted a broad overview of the research area. **Missing important primary studies**: We conducted the review in several search engines, even though it is rather possible we missed some primary studies. We mitigated this threat by selecting search engines which have been regarded as the most relevant scientific sources [6]. **Reviewers reliability**: The reviewers of this study are researchers in the software reuse field. So, we are not aware of any bias we may have introduced during the analyses. **Data extraction**: Another threat for this review refers to how the data were extracted from the digital libraries, since not all the information was obvious to answer the questions and some data had to be interpreted. In order to ensure the validity, multiple sources of data were analyzed, i.e. papers, technical reports, white papers. In the event of a disagreement between the two primary reviewers, a third reviewer acted as an arbitrator to ensure full agreement was reached.

6 Concluding Remarks

Our long-term research goal is to find out how ADM has been applied in the literature to assist software engineers during the process of modernization of legacy system. To do so in this paper we presented a systematic review of ADM and its metamodels, following the process described by Kitchenham [6]. Through an examination of 30 primary studies, this review has presented 22 papers which describe processes, 15 that propose a set of model transformation, 9 that present tools, 3 which present metrics and 2 that illustrate extension in the ADM's metamodels. Notice that certain primary studies were grouped in more than one category, affecting the frequency count; i.e., the sum of the frequencies shown in each facet of the Figure 2 can be greater than the total of selected studies identified, i.e., 30.

In summary, this review shows that there are some researchers investigating ADM and its metamodels. As consequence, we were able to identify that there are a set of processes which have been commonly used in literature to assist the software engineer to modernize legacy system by means of ADM. Afterwards, it was also possible to identify that KDM is the ADM's metamodel more utilized in the literature. Similarly, the KDM's packages most used in the literature are Code and Action - the least used are Event and UI packages. The main future directions that emerged from this review are highlighted in Section 4.1. Likewise, as long term future work, related systematic reviews are about to be concluded and the relationship among their results should be investigated aiming to characterize this area in a deeper way. As soon as they are concluded, their results should be presented for the academic community as a technical report that will become available online.

References

1. Izquierdo, J., Molina, J.: An architecture-driven modernization tool for calculating metrics. Software, IEEE (2010)
2. Perez-Castillo, R., Cruz-Lemus, J.A., de Guzman, I.G.R., Piattini, M.: A family of case studies on business process mining using marble. Systems and Software (2012)

3. Frey, S., Hasselbring, W., Schnoor, B.: Automatic conformance checking for migrating software systems to cloud infrastructures and platforms. Journal of Software: Evolution and Process (2012)
4. Fuentes-Fernandez, R., Pavon, J., Garijo, F.: A model-driven process for the modernization of component-based systems. Science of Computer Programming (2012)
5. Perez-Castillo, R., de Guzman, I.G.R., Piattini, M.: Knowledge discovery metamodel-iso/iec 19506: A standard to modernize legacy systems. Computer Standards & Interfaces **33**(6) (2011) 519 – 532
6. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering. Information and Software Technology (2009)
7. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria. (2005)
8. Petersen, K., Feldt, R., Mattsson, M.: Systematic mapping studies in software engineering. In: 12th Inter. Conf. on Evaluation and Assessment in Software Engineering. (2008)
9. Anonymized: Anonymized. In: 28th Annual ACM Symposium on Applied Computing. (2013)
10. Moratalla, J., de Castro, V., Sanz, M., Marcos, E.: A gap-analysis-based framework for evolution and modernization: Modernization of domain management at red.es. In: SRII Global Conf. (2012)
11. Yazdanshenas, A.R., Moonen, L.: Crossing the boundaries while analyzing heterogeneous component-based software systems. In: 27th IEEE Inter. Conf. on Software Maintenance. (2011)
12. Mazón, J.N., Trujillo, J.: A model driven modernization approach for automatically deriving multidimensional models in data warehouses. In: 26th Inter. Conf. on Conceptual modeling. (2007)
13. Guzman, I.G.R.d., Polo, M., Piattini, M.: An adm approach to reengineer relational databases towards web services. In: 14th Work. Conf. on Reverse Engineering. (2007)
14. Frey, S., Hasselbring, W.: An extensible architecture for detecting violations of a cloud environment's constraints during legacy software system migration. In: Software Maintenance and Reengineering, 15th European Conf. on. (2011)
15. Garcia-Rodriguez de Guzman, I.: Pressweb: A process to reengineer legacy systems towards web services. In: Reverse Engineering, 14th Working Conf. on. (2007)
16. Perez-Castillo, R., Garcia Rodriguez de Guzman, I., Piattini, M., Piattini, M.: On the use of adm to contextualize data on legacy source code for software modernization. In: 16th Work. Conf. on Reverse Engineering. (2009)
17. del Castillo, R.P., García-Rodríguez, I., Caballero, I.: Preciso: A reengineering process and a tool for database modernisation through web services. In: ACM SAC. (2009)
18. Pérez-Castillo, R., de Guzmán, I.G.R., Piattini, M.: Database schema elicitation to modernize relational databases. (2012)
19. Rodríguez-Echeverría, R., Clemente, P.J., Preciado, J.C., Sanchez-Figueroa, F.: Modernization of legacy web applications into rich internet applications. In: 11th Inter. Conf. on Current Trends in Web Engineering. (2012)
20. Boussaidi, G., Belle, A., Vaucher, S., Mili, H.: Reconstructing architectural views from legacy systems. In: Reverse Engineering, 19th Work. Conf. on. (2012)
21. Pérez-Castillo, R., de Guzmán, Weber, B., Places, A.S.: An empirical comparison of static and dynamic business process mining. In: ACM Symposium on Applied Computing. (2011)
22. Perez-Castillo, R., Fernandez-Ropero, M., Garcia-Rodriguez de Guzman, I., Piattini, M.: Marble. a business process archeology tool. In: 27th IEEE Inter. Conf. on Software Maintenance. (2011)
23. Perez-Castillo, R., Garcia-Rodriguez de Guzman, I., Piattini, M.: Mimos, system model-driven migration project. In: 17th European Conf. on Software Maintenance and Reengineering. (2013)

24. Pérez-Castillo, R., De Guzmán, I.G.R., Piattini, M.: Implementing business process recovery patterns through qvt transformations. In: Inter. Conf. on Theory and practice of model transformations, Springer-Verlag (2010) 168–183
25. Normantas, K., Vasilecas, O.: Extracting business rules from existing enterprise software system. In: Information and Software Technologies. Communications in Computer and Information Science, Springer Berlin Heidelberg (2012)
26. Fernández-Ropero, M., Pérez-Castillo, R., Weber, B., Piattini, M.: Empirical assessment of business model transformations based on model simulation. In: Inter. Conf. on Theory and Practice of Model Transformations. (2012)
27. Baresi, L., Miraz, M.: A component-oriented metamodel for the modernization of software applications. In: Engineering of Complex Computer Systems, 16th IEEE Inter. Conf. on. (2011)
28. Pérez-Castillo, R., de Guzmán, I.G.R., Piattini, M., Weber, B.: Integrating event logs into kdm repositories. In: 27th Annual ACM Symposium on Applied Computing. (2012)
29. Shahshahani, P.M.: Extending the knowledge discovery metamodel to support aspect- oriented programming. (2011)
30. Bruneliere, H., Cabot, J., Jouault, F., Madiot, F.: Modisco: a generic and extensible framework for model driven reverse engineering. In: IEEE/ACM Inter. Conf. on Automated software engineering. (2010)
31. Perez-Castillo, R., Weber, B., de Guzman, I.R., Piattini, M.: Process mining through dynamic analysis for modernising legacy systems. Software, IET (2011)
32. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: Refactoring: Improving the Design of Existing Code. Addison-Wesley (2000)