A Systematic Review on Architecture-Driven Modernization

ABSTRACT

Background: Software modernization is critical for organizations in need of cost-effective solutions to cope with the rapid obsolescence of software and the increasing demand for new functionalities. OMG (Object Management Group) has proposed the Architecture-Driven Modernization (ADM) an approach for model-based modernizations, which contains a set of standard metamodels. Nevertheless, to the best of our knowledge, there is no a literature review available in this field. Thus, we argue that there is need for evidence-based body of knowledge about different aspects of the adoption of ADM. **Objective:** To conduct a systematic review describing research into ADM. **Method:** A systematic review based on searching of major electronic databases was undertaken. **Results:** We identified 30 primary studies which were classified by their contribution, focus area and research type. Among these studies we found out a set of process and tool to assist ADM. **Conclusion:** ADM has become a popular approach for modernizing legacy systems by means of Model-Driven Development (MDD). There is an overall positive perception about this approach. The findings of this paper provide interesting insights into different aspects of ADM. We expect that the findings can provide valuable information to readers on what can be expected from applying ADM and its metamodels to modernize a legacy systems. Furthermore, the results can provide percetion of new research in the modernization area for investigating and defining new tools/process to assist the modernization of legacy system.

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 embodied knowledge due to years of maintenance and this constitutes a significant corporate asset. 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.

The first task that must be performed in order to carrying out a software modernization is understand the legacy system. This is not a trivial task; in fact studies estimate that between 50 percent and 90 percent of software maintenance involves developing an understanding of the software being maintained [33], thus several approaches have been developed to support software engineers in the comprehension of systems where reverse engineering (RE) is one of them [5]. RE supports program comprehension by using techniques that explore the source code to find relevant information related to functional and non-functional features [6].

In this context, OMG has employed effort to define standards in the modernization process, creating the concept of ADM. ADM follows the MDD [15] 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 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 get an overview of existing research in this context, we performed a systematic review of ADM and its metamodels. Apart from getting an overview, this study also aims at identifying and presenting results from literature that are valuable from perspective of possible future enhancements and use. Systematic review studies belong to Evidence-Based Software Engineering (EBSE) paradigm [16]. It provides new, empirical and systematic methods of research. Although several studies have been reported in the context of ADM [11,12,21,24], 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. Thus, this paper aims to conduct a systematic review describing research into ADM. Also we argue that this paper 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, it serves as a first step towards more thorough examination of the topics addressed in it with the help of systematic literature reviews.

Following this introduction, this paper is structured as follows: In Section 2, describes how the systematic review methodology has been planned, conducted and reported. In Section 3.2 there are the threats to validity of our study. Concluding remarks are made in Section 4.

2. THE SYSTEMATIC REVIEW

This study was undertaken as a systematic review based on the guidelines proposed by Kitchenham and Brereton [16] which consist of three main phases: (*i*) planning, (*ii*) conducting and (*iii*) reporting.

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.

As described before, the objective of this review is to find out **how ADM has been applied in the literature to assist engineers during the process of modernization of legacy systems**. In order to achieve such objective we formulated five research questions. $\mathbf{RQ_1}$**:** What are the focus area most discussed and least discussed in the literature regarding the ADM? Moreover, what types of contributions have been presented so far?; $\mathbf{RQ_2}$**:** 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?; $\mathbf{RQ_3}$**:** Which research types have been employed into the field herein?; $\mathbf{RQ_4}$**:** Which are the existing tool support for ADM? Moreover, how the tools perform the identification of what needs to be modernized?

To address $\mathbf{RQ}_1$, we read all primary studies 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. With respect to $\mathbf{RQ}_3$, we used and adapted the scheme proposed by Wieringa et al [34] in order to classify each primary study into a research type, see Section 2.2.3. Finally, to address $\mathbf{RQ}_4$ we read all primary studies in order to identifiy the tools used to modernize the legacy systems.

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. The search encompassed electronic databases which are deemed as the most relevant scientific sources [16] 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.

("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")

Figure 1: Search String.

Then, in order to determine which primary studies are relevant to answer our research questions, we applied a set of inclusion and exclusion criteria. Inclusion criteria applied were: (*i*) **The primary study presents at least one solution of modernization by means of ADM** - (*ii*) **Studies that explicitly present an ADM approach** - (*iii*) **The primary study presents at least one type of evaluation technique for ADM**.

Not all of these criteria must be present for every primary study. However, at least the former, i.e., (a), must be present. If all criteria were mandatory, the number of selected techniques would decrease significantly.

Exclusion criteria applied were: (*i*) **Papers which mentioned ADM and its metamodels only in the abstract** - (*ii*) **Introductory papers for books and workshops** and (*iii*) **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 three reviewers. The review was performed in August, 2013 by two M.Sc. and a Ph.D. student. 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 identified primary studies in the following digital libraries: IEEE, ACM, Scopus, Web of Science and Engeneering Village. We applied the search string given in Figure 1.

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, 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. [31] 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 [1].Thus, the aforementioned categories were adapted to specifics of our systematic study. The resultant classification schemes are as follows:

2.2.1 Focus Area

We used the keywording method described in [31] to identify the focus area of the identified studies. As result we got four ones - a brief description of each focus area follows: (*i*) **Approaches to modernize legacy systems to another platform/architecture (SOA, change programming language, web 2.0, mobile, etc):** 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; (*ii*) **Business Knowledge Extraction:** Describes primaries studies which address process, method or approach to extract business process of a legacy system; (*iii*) **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; and (*iv*) **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.

2.2.2 Contribution Type

During the reading of the primary studies was possible to identify a set of contribution types related to ADM and their metamodels. We identified five contribution types: (*i*) **Tool:** This contribution type refers to the primary studies that focus on providing tools to support the modernization of legacy system by using ADM and its metamodels, i.e., either in the form of a prototype or a tool that can be integrated with existing environments; (*ii*) **Process:** Similarly, it refers to contributions which specifically describe a process to assist the modernization of legacy system by means of ADM and its metamodels; (*iii*) **Model Transformation:** It refers to contributions which describe transformation among ADM's metamodels, e.g., primary studies that describe the use of language transformation such as Query/Views/Transformations (QVT)[1] or ATL Transformation Language (ATL)[2]; (*iv*) **Metamodel:** This contribution type 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; and (*v*) **Metrics:** This type of contributions focus on proposing or applying metrics to effectiveness of ADM and its metamodels.

2.2.3 Research Type

The research type reflects the research approach used in the primary study. We used and adapted the scheme proposed by Wieringa et al [34] herein. A brief description of research types are as follows: (*i*) **Validation research:** The main purpose of validation research is to examine a solution proposal that has not yet been practically applied. Validation research is conducted in a systematic way and may present any of these: prototypes, math analysis, etc; (*ii*) **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; (*iii*) **Conceptual proposal:** A conceptual proposal presents an arrangement to perceive things that already exist, in a novel way. However it does not precisely solve a particular problem. Conceptual proposals may include taxonomies, theoretical frameworks, etc; (*iv*) **Experience paper:** An experience paper reports on personal experience of the author from one or more real life projects. It usually elaborates on what was accomplished in the project as well as how it was actually done; and (*v*) **Opinion paper:** Opinion papers report on personal opinion of the author on suitability or unsuitability of a specific technique or tool. Similarly, these are sometimes used to share personal opinion describing as to how some technique or tool should have been developed, etc.

2.3 Reporting the Systematic Review

The focus of this section is to present broad overview of research within ADM and its metamodels 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 and its metamodels we plotted a bubble plot, which is depicted in Figure 2. Bubble plots are essentially two x-y scatter plots with bubbles in category intersections. The size of each bubble is determined by the number of primary studies that have been classified as belonging to the categories corresponding to the bubble coordinates. This visual summary provides a bird's-eye view that enables one to pinpoint which categories have been emphasized in past research along with gaps and opportunities for future research.

In Figure 2 the facets we used for organizing the map are the **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, **Model transformation** and **Tool** (to assist ADM's process) are also another field which have been researched. Maybe this happened once the majority of the primary studies found which describe a process to assist the modernization of legacy systems usually propose a set of

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

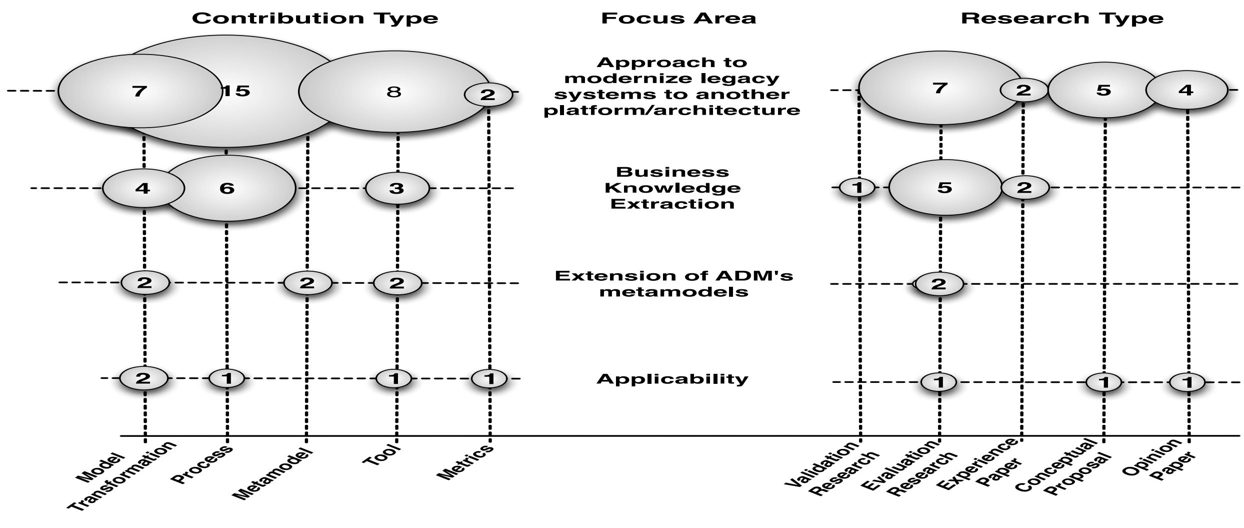
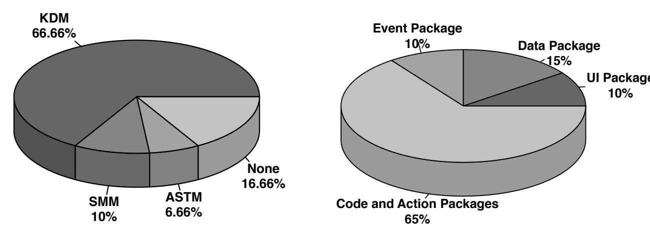


Figure 2: Map of research focus on ADM and its metamodels.

model transformation and a semi-automatic tool or a fully-automatic one. In contrast, 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 and its metamodels, 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 (i.e., 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 (i.e., where new or better primary studies are required). In other words, **Process** to assist software engineer during the modernization of legacy system to another platform/architecture, **Model transformation** and **Tool** have been covered by over 43%, 29% and 17%, respectively of the current research (see Figure 4a). On the other hand, **Metamodel** and **Metrics** have been addressed by a very small number of publications i.e., 3.92% and 5.88%, respectively (see Figure 4a).

As result of this analysis we answered partially the $\mathbf{RQ}_1$, i.e., we answered the types of contributions that have been presented so far in liteture 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 2.3.1 through 2.3.4. We organized each subsection in a way that it briefly describes the studies selected for each topic while highlighting the nature of research.

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



(a) (b)

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

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.

On the facet **Research Type**, shows the data we gathered related to the research type employed in the field of ADM ($\mathbf{RQ}_3$). As far as the research type is concerned, **Evaluation Research** are in vast majority, covering 15 primary stydies, see Figure 2 on the right side. 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. Finally, in Figure 4b shows the distribution by year of the accepted primary studies. As can be seen, the year 2012 had the biggest number of publications related to ADM and

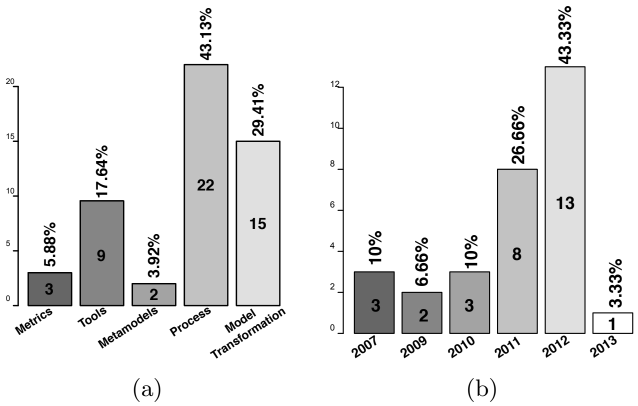


Figure 4: Frequency of studies in each category and Distribution of publication by years.

Table 1: The analyzed information of each tool.

Cat.	Name	Ref	Website
Modernize	CloudMIG	[11]	tinyurl.com/cloudmigxpress
	MARBLE	[22, 23, 27, 28]	No
	MIMOS	[28]	No
	PRECISO	[7]	No
	R2SOA	[14]	No
	WA2RIA	[32]	No
Abst.	COMO	[2]	No
	Gra2MoL	[15]	modelum.es/trac/gra2mol/
	MoDisco	[4]	eclipse.org/MoDisco/

its metamodels, i.e., 43.33%. Among the 30 primary studies included herein 8 were published in 2011, i.e., 26.66%. In 2007 and in 2010 were published 3 primary studies related to ADM and its metamodelo, representing 20% of all primary studies. Among the 30 primary studies any of them were published in 2008. In 2009 was identified a percentage of 6.66 of primary studies related to the review herein. Taking into account the search strings was applied in 2013, it may explain the low amount of primary studies published in 2013.

As depicted in second bar of the Figure 4a, we identified 9 tools. Among them six were developed to assist the modernization of legacy systems. The remainder were elaborated to transform the legacy sytem's source-code into instance of ADM's metamodels. In Table 1 has four columns to categorize, name and reference such tools. The column, Cat., stand for "Categories" of the tools. There are two categories: (*i*) **Modernization** which represents tool that assist to modernize legacy systems to another plataform/architecture, and (*ii*) **Abst.** ("Abstraction") which symbolizes tool that takes the legacy system's source-code and instantiate a representation of the ADM's metamodels, i.e., brings the source-code into another abstraction level. By using this table was possible to answer the fisrt part of the $\mathbf{RQ}_5$. As for answering the second part of it, a brief description about the identified tools are presented as follows.

CloudMIG. It provides a semi-automatically support for the migration of software systems to PaaS or IaaS-based clouds by using KDM. This tools identifies the possible services by applying a set of rules heuristics., such as: Distribute the five most frequently used services to own virtual machines or the server methods responsible for at least 10% of overall consumption of the CPU time shall be moved to client side components if they do not need access to the database. After, CloudMIG can generate considerable parts of a resource-efficient target architecture utilizing these rules heuristics [11].

MARBLE. It is a modernization tool to recover business processes from legacy systems based on KDM. It uses two process mining techniques, static and dynamic analysis of source code to identify business processes. The static one is based on a module that analyze the source code file (Java files in particular) and builds an abstract syntax tree of the source code and then create an instance of KDM. On the other hand, the dynamic one takes the information about the system execution into account to extract meaningful business knowledge.

MIMOS. Similar to MARBLE, MIMOS is a tool to recover business processes from legacy systems based on KDM [28]. This tool uses parser to identify the business processes, then a set of refactoring techniques is applied to KDM.

PRECISO. It is a tool for database modernisation through Web Services. Potential services can be simultaneously discovered along with database reverse engineering. A set of patterns are sought in recovered database schema [7].

R2SOA. It is a complete process to reengineer relational databases at a model level to integrate them into SOA contexts as a set of services. Firstly, database schema is reversed and a suitable model is built from the metadata extracted from the database catalog. Based on the structure of the database schema, a first service extraction can be undertaken, a set of model driven pattern matching is used, then CRUD operations are automatically included [14].

WA2RIA. It can modernize legacy Web Applications (WA) to Rich Internet Applications (RIAs) by using ADM and KDM. This tool uses MoDisco to get an instance of KDM, then a set of rules are applied to realize the modernization of the WA to RIA [32].

COMO. It provides an extension to KDM with the concepts of component and interface. By lifting the abstraction level, COMO can produce an adequate view of the entire system, and leverage the boundaries between components to better focus the subsequent modernization effort [2]. This tool neither offer a semi-automatically or automatically way to identify the components, i.e., the engineer has to identify the components.

Gra2MoL. It has been specifically designed to address the problem of extracting models from source code. Gra2MoL is a rule-based transformation language like existing model-to-model transformation languages, but with the fundamental difference that the source element of a rule is a grammar element instead of a source metamodel element [15]. Gra2MoL uses parsers to identify and to extract models from source code.

MoDisco. It aids to offer an open source generic and extensible MDD framework. It aims at providing the required capabilities for creating models and allowing there handling, analysis and computation [4].

2.3.1 *Approach to modernize legacy systems to another platform/architecture*

He we describes different studies regarding approaches to modernize legasy systems by means of ADM. Jorge Maratalla et al., propose **GAFEMO** [19] 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 [35] the autors 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 [18] 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 [14] 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 [10, 11] 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 [13] 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.

In [7, 25] is presented a reengineering process that uses ADM to recover and implement Web Services in automatic manner from relational databases. Pérez-Castillo et al., [29] 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 modelling 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 [12] 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.

Mainetti et al., [17] present an approach that allows developers to automatically modernize the client side of legacy systems. In this approach developers can refactor the Graphical User Interface (GUI) of legacy systems during the modernization, taking the opportunities offered by novel interaction paradigms, i.e., Rich Internet Application (RIA). Similarly, in [32] the authors present an approach for the definition of a systematic process for Web Applications (WA) to RIA modernization, by applying ADM principles, techniques and tools. The appoach 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., [3] 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 [15] 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.

2.3.2 Business Knowledge Extraction

Several papers have contributed to propose approaches to identify Business Knowledge by means of ADM. A brief description of these paper are follows. Pérez-Castillo et al., [22, 23, 27, 28] present an approach to recover business processes from legacy systems. More specifically, 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 [21] the authors report the results of a family of case studies that were performed to empirically validate this approach.

Normantas and Vasilecas [20] 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., [8] 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. The authors takes an MXML model and obtains an equivalent KDM model at the same abstraction level. The proposed set of rules consist of eight declarative transformation rules.

2.3.3 Extension of ADM's metamodels

We identified two papers that address how to perform extension of ADM's metamodels. We provided a brief summary of these paper are follows. In [2] 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 [26] propose an extension to the KDM that aims to represent all the information registered in a MXML model in the KDM model. They claimed that the impact of this extension on well-proven and KDM based tools is not problematic since it is carried out with the own extension mechanism of the KDM standard. Besides this fact, most elements of the event model are present in the core of KDM which is used for many tools.

2.3.4 Applicability

Similarly, we also identified a small number of papers that address just the applicability of the ADM and its metamodel. Pérez-Castillo et al., [24] present how to apply KDM to modernize legacy systems. Also in this paper 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

3. DISCUSSION

In this section we provide a summary of key findings of this study and present some directions of new research that we ascertained during the review, i.e., we pinpoint some open issues that still need to be researched in ADM. We also highlight the limitations of this mapping study that may represent threats to its validity.

3.1 Principle Findings

Recent proposals in ADM have focused mainly on providing approaches to modernize legacy system to another plataform/architecture resulting in appearance of several proposals, see Section 2.3.1. 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., [4, 15]) 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, we argue that 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 [30] the authors presented an approach that dynamically discovery knowledge of the targe 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 KDM specification 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 programmers to see where they are heading when modernizing their KDM model. Fowler et al. [9] advocated a specific notion of style for object-oriented programming through a catalog of 22 code smells, compounded by a catalog of 72 refactorings through which those smells can be removed from existing code. These catalogs proved very useful in bringing the concepts of refactoring and good object-oriented style to a wider audience and in providing programmers with guidelines on when to refactor and how best to refactor. 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 programmers to detect situations in the KDM model 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. [9] 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 engineer during the modernization process.

Finally, we also noticed that all researches have been conducted in an isolated manner. For instance, there are both researches which focus on identified SaaS [10, 11] and researches that aims to identify Web-Services [7, 25] but none research has been done in order to combine efforts, i.e., techniques.

3.2 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: The search for primary studies was conducted in several search engines, even though it is rather possible we have missed some primary studies. We mitigated this threat by selecting search engines which have been regarded as the most relevant scientific sources [16].

Reviewers reliability: All the reviewers of this study are researchers in the software reuse field. Thus, 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.

4. CONCLUDING REMARKS

Our long-term research goal is to find out how ADM has been applied in the literature to assist 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 [16]. Through an examination of 30 primary studies encompassing ADM, this review has presented 22 papers which describe processes, 15 that propose a set of model transformation, 9 that put forward tools, 3 which present metrics and 2 that illustrate same 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.

The initial significant contributions to ADM were presented in 2007 (i.e., [13,14,18]), in 2008 had no publications. However, in 2009 we identified two paper dealing with ADM (i.e., [7, 29]). Next, we had a rise from 2010 (3 papers) to 2011 (8 papers). After, in 2012 the rate of papers gradually rose to 13. Then, it dropped out in 2013. This may happed once we carried out the review in 2013. Among these paper were found that most of them have appeared in conferences and journal, while just a was reported in workshop. We claim that researchers can use this review as a basis for advancing the field, while practitioners can use it to identify process, tools, metrics, etc that are well-suited to their needs. This systematic review should serve not only academic researchers but also industrial professionals, aiming at adopting some process, tool, metrics, to modernize a legacy systems within their organizations.

In summary, this review shows that there are some researchers investigating related to ADM and its metamodels. As consequence, we were able to identify that there are a set of process which have been commonly used in literature to assist the software engineer to modernize legacy system by means of ADM. Afterwards, was also possible to identify that KDM is the ADM's metamodel more utilized into the literature. Similarly, the KDM's packages most used in the literature are Code and Action - the least used are Event and UI packages. As future work, other related systematic review are about to be concluded and the relationship among their results should be investigated aiming to characterise 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.

5. REFERENCES

[1] Anonymized. Anonymized. In *28th Annual ACM Symposium on Applied Computing*, 2013.

[2] L. Baresi and M. Miraz. A component-oriented metamodel for the modernization of software applications. In *Engineering of Complex Computer Systems, 16th IEEE Inter. Conf. on*, 2011.

[3] G. Boussaidi, A. Belle, S. Vaucher, and H. Mili. Reconstructing architectural views from legacy systems. In *Reverse Engineering, 19th Work. Conf. on*, 2012.

[4] H. Bruneliere, J. Cabot, F. Jouault, and F. Madiot. Modisco: a generic and extensible framework for model driven reverse engineering. In *IEEE/ACM Inter. Conf. on Automated software engineering*, 2010.

[5] G. Canfora, M. Di Penta, and L. Cerulo. Achievements and challenges in software reverse engineering. *Commun. ACM*, 54:142–151, Apr. 2011.

[6] E. J. Chikofsky and J. H. Cross II. Reverse Engineering and Design Recovery: A Taxonomy. *IEEE Software*, 7(1):13–17, 1990.

[7] R. P. del Castillo, I. García-Rodríguez, and I. Caballero. Preciso: A reengineering process and a tool for database modernisation through web services. In *ACM SAC*, 2009.

[8] M. Fernández-Ropero, R. Pérez-Castillo, B. Weber, and M. Piattini. Empirical assessment of business model transformations based on model simulation. In *5th Inter. Conf. on Theory and Practice of Model Transformations*, 2012.

[9] M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, Agosto 2000.

[10] S. Frey and W. Hasselbring. 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.

[11] S. Frey, W. Hasselbring, and B. Schnoor. Automatic conformance checking for migrating software systems to cloud infrastructures and platforms. *Journal of Software: Evolution and Process*, 2012.

[12] R. Fuentes-Fernandez, J. Pavon, and F. Garijo. A model-driven process for the modernization of component-based systems. *Science of Computer Programming*, 2012.

[13] I. Garcia-Rodriguez de Guzman. Pressweb: A process to reengineer legacy systems towards web services. In *Reverse Engineering, 14th Working Conf. on*, 2007.

[14] I. G.-R. d. Guzman, M. Polo, and M. Piattini. An adm approach to reengineer relational databases towards web services. In *14th Work. Conf. on Reverse Engineering*, 2007.

[15] J. Izquierdo and J. Molina. An architecture-driven modernization tool for calculating metrics. *Software, IEEE*, 2010.

[16] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering. *Information and Software Technology*, 2009.

[17] L. Mainetti, R. Paiano, and A. Pandurino. Migros: a

model-driven transformation approach of the user experience of legacy applications. In *12th Inter. Conf. on Web Engineering*, 2012.

[18] J.-N. Mazón and J. Trujillo. A model driven modernization approach for automatically deriving multidimensional models in data warehouses. In *26th Inter. Conf. on Conceptual modeling*, 2007.

[19] J. Moratalla, V. de Castro, M. Sanz, and E. Marcos. A gap-analysis-based framework for evolution and modernization: Modernization of domain management at red.es. In *SRII Global Conf.*, 2012.

[20] K. Normantas and O. Vasilecas. Extracting business rules from existing enterprise software system. In *Information and Software Technologies*, Communications in Computer and Information Science. Springer Berlin Heidelberg, 2012.

[21] R. Perez-Castillo, J. A. Cruz-Lemus, I. G.-R. de Guzman, and M. Piattini. A family of case studies on business process mining using marble. *Systems and Software*, 2012.

[22] R. Pérez-Castillo, B. de Guzmán, Weber, and A. S. Places. An empirical comparison of static and dynamic business process mining. In *ACM Symposium on Applied Computing*, 2011.

[23] R. Pérez-Castillo, I. G.-R. De Guzmán, and M. Piattini. Implementing business process recovery patterns through qvt transformations. In *Proceedings of the Third international conference on Theory and practice of model transformations*, pages 168–183. Springer-Verlag, 2010.

[24] R. Perez-Castillo, I. G.-R. de Guzman, and M. Piattini. Knowledge discovery metamodel-iso/iec 19506: A standard to modernize legacy systems. *Computer Standards & Interfaces*, 33(6):519 – 532, 2011.

[25] R. Pérez-Castillo, I. G. R. de Guzmán, and M. Piattini. Database schema elicitation to modernize relational databases. 2012.

[26] R. Pérez-Castillo, I. G.-R. de Guzmán, M. Piattini, and B. Weber. Integrating event logs into kdm repositories. In *27th Annual ACM Symposium on Applied Computing*, 2012.

[27] R. Perez-Castillo, M. Fernandez-Ropero, I. Garcia-Rodriguez de Guzman, and M. Piattini. Marble. a business process archeology tool. In *27th IEEE Inter. Conf. on Software Maintenance*, 2011.

[28] R. Perez-Castillo, I. Garcia-Rodriguez de Guzman, and M. Piattini. Mimos, system model-driven migration project. In *17th European Conf. on Software Maintenance and Reengineering*, 2013.

[29] R. Perez-Castillo, I. Garcia Rodriguez de Guzman, M. Piattini, and M. Piattini. On the use of adm to contextualize data on legacy source code for software modernization. In *16th Work. Conf. on Reverse Engineering*, 2009.

[30] R. Perez-Castillo, B. Weber, I.-R. de Guzman, and M. Piattini. Process mining through dynamic analysis for modernising legacy systems. *Software, IET*, 2011.

[31] K. Petersen, R. Feldt, and M. Mattsson. Systematic mapping studies in software engineering. In *12th Inter. Conf. on Evaluation and Assessment in Software Engineering*, 2008.

[32] R. Rodríguez-Echeverría, P. J. Clemente, J. C. Preciado, and F. Sanchez-Figueroa. Modernization of legacy web applications into rich internet applications. In *11th Inter. Conf. on Current Trends in Web Engineering*, 2012.

[33] S. Tilley and D. Smith. Perspectives on legacy system reengineering. 1995.

[34] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria. 2005.

[35] A. R. Yazdanshenas and L. Moonen. Crossing the boundaries while analyzing heterogeneous component-based software systems. In *27th IEEE Inter. Conf. on Software Maintenance*, 2011.