A Systematic Review on Architecture-Driven Modernization

Rafael S. Durelli
ICMC - USP
São Carlos, SP, Brazil
rdurelli@icmc.usp.br

Daniel S. M. Santibáñez
DC - UFSCar
São Carlos, SP, Brazil
daniel.santibanez@dc.ufscar.br

Nicolas Anquetil
RMoD Team - INRIA
Lille, France
Nicolas.Anquetil@inria.fr

Márcio E. Delamaro
ICMC - USP
São Carlos, SP, Brazil
delamaro@icmc.usp.br

Valter Vieira de Camargo
DC - UFSCar
São Carlos, SP, Brazil
valter@dc.ufscar.br

ABSTRACT

Background: Software modernization is critical for organizations that need cost-effective solutions to deal with the rapid obsolescence of software and the increasing demand for new functionality. In this context, the Object Management Group (OMG) promotes the Model Driven Development (MDD) concept and proposes the Architecture Driven Modernization (ADM) approach for model-based platform migration, which contains a set of standard representations of views on existing software systems. Objectives: To conduct a systematic review study of the literature describing research into Architecture Driven Modernization. Research method: We undertook a systematic review study of the literature based upon searching of major electronic databases. Results: We selected **x** and classified by their contribution, Architecture Driven Modernization process implementation, type and date of publication. Conclusion: As a result, the review identified that there are still gaps in the modernization process. Furthermore, the results can provide insights for new research in the modernization area for investigating and defining new tools/process to assist the modernization of system

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Miscellaneus

Keywords

Systematic Review, Concern Mining, Aspect Mining, Crosscutting Concerns.

1. INTRODUCTION

2. BACKGROUND AND MOTIVATION

In this section we provide a brief background to Architecture-Driven Modernization (ADM) presenting the core ideas. Furthermore, this section describes the ADM standards, e.g., Knowledge Discovery Metamodel (KDM), Abstract Syntax Tree Metamodel (ASTM) and Software Metrics Metamodel (SMM). Finally, we justify the need for a systematic review study in this research field.

2.1 Architecture-Driven Modernization

Nowadays, researchers have been shifted from the typical refactoring process to the so-called Architecture-Driven Modernization (ADM). ADM is the concept of modernizing existing systems with a focus on all aspects of the current systems architecture and the ability to transform current architectures to target architectures by using all principles of MDA [**?**, p. 60]. Figure 1 shows the ADM modernization domain model where the left side of the horseshoe is the current state of a business architecture "as-is" and the right side is what we want to get after the modernization "to-be".

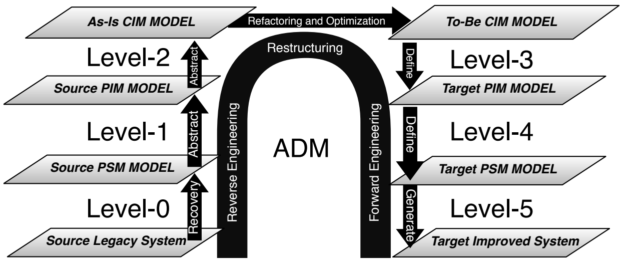

Figure 1: Modernization domain model (Adapted from Ulrich and Newcomb [**?**])

As can be seen in Figure 1 the horseshoe reengineering model has been adapted to ADM and it is nowadays known as horseshoe modernization model. As ADM uses the principles of MDA three kinds of models in the horseshoe are used, they are: (*i*) PIM - **P**lataform **I**ndependent **M**odel which represents a view of the system from the platform independent viewpoint at an intermediate abstraction level, (*ii*) PSM - **P**lataform **S**pecific **M**odel which constitutes a view of the system from the platform specific viewpoint at a low abstraction level, and (*iii*) CIM - **C**omputational **I**ndependent **M**odel that represents a view of the system from the computational independent viewpoint at a high abstraction level. These models are used in the steps of the ADM process, i.e., Reverse Engineering, Restructuring, and Forward Engineering. In the first step, a reverse engineering is performed starting from the artifacts of the legacy system (source code,

SAC'13 March 18-22, 2013, Coimbra, Portugal.
Copyright 2013 ACM 978-1-4503-1656-9/13/03 ...$10.00.

database, configuration files, etc) and a set of PSM are created. Next, refactoring and restructuring techniques can be applied on these models in order to solve problems found in the legacy system. Therefore, this step consist of a set of transformation from the input model ("as-is") to obtain a target model ("to-be"). Finally, a forward engineering is carried out and the source code of the modernized target system is generated again.

In order to perform such steps, ADM introduces several modernization standards: Abstract Syntax Tree Metamodel (ASTM), Knowledge Discovery Metamodel (KDM), Structured Metrics Metamodel (SMM), etc. The next subsections present more information about these standards:

2.1.1 Abstract Syntax Tree Metamodel - ASTM

Abstract Syntax Tree Metamodel (ASTM) was established to represent software at a very granular level of procedural logic, data definition, and workflow composition. ASTM can provide this granular level of information to KDM (see Section 2.1.2) to augment the KDM view of a system. As a standard, ASTM can stand alone and supports tools geared at the complete, functionally equivalent refactoring and transformation of a system from one platform and language environment to a target platform and language environment. ASTM is a model that represent software artifacts using data structures that represent the types of language constructs, its compositional relationships to other language constructs, and a set of direct and derived properties associated with each language construct. The ASTM is derived by analyzing software artifacts and provides a way to create a representation of those software artifacts.

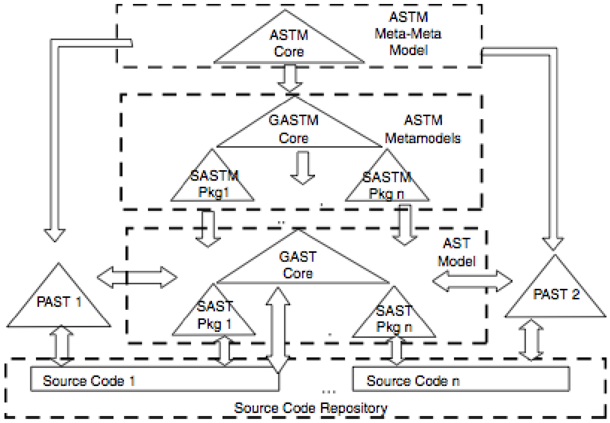

Figure 2: ASTM modeling framework

ASTM aims to facilitate exchanging software models in standard formats among tools. This ability of exchanging software models between tools is thanks to its attributes, they are: (i) ASTM is language and platform independent, but can be extended as needed, (ii) ASTM uses XMI formats for tool-based metadata exchange, iii Generic Abstract Syntax Tree Metamodel (GASTM) represents a generic set of language modeling elements common across numerous languages. Language Specific Abstract Syntax Tree Metamodel (SASTM) represents particular languages such as Ada, C, FORTRAN, and Java, iv Proprietary Abstract Syntax Tree Metamodel (PASTM) expresses ASTs for languages such as Ada, C, COBOL, etc., modeled in formats inconsistent with MOF, the GSATM, or SASTM. Figure 2 represents the structure of the ASTM, including the SASTM, GASTM, and PASTM.

2.1.2 Knowledge Discovery Metamodel - KDM

Knowledge Discovery Metamodel (KDM) is the key within set of standards [?]. KDM allows standardized representation of knowledge extracted from legacy systems by means of reverse engineering. KDM provides a common repository structure that makes possible the exchange of information about existing software assets in legacy systems. This information is currently represented and stored independently by heterogeneous tools focused on different software assets [?, p. 32]. Figure 3 shows each of the varying views of the existing IT architecture represented by the KDM. For example, the build view, depicts system artifacts from a source, executable, and library viewpoint. Other perspectives include design, conceptual, data, and scenario views.

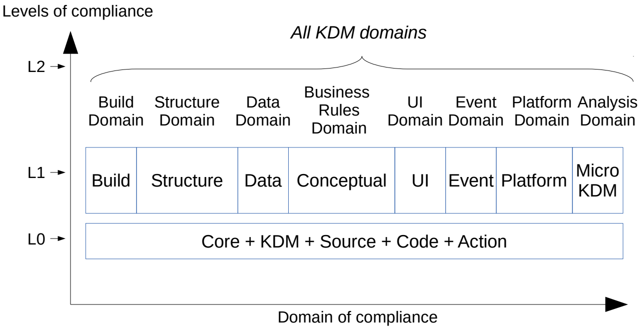

Figure 3: KDM domains of artifact representation (Adapted from Ulrich and Newcomb [?])

The Level 0 (L0) encompasses the Infrastructure and Program Elements Layer. Infrastructure Layer consists of the Core, kdm, and Source packages which provide a small common core for all other packages. Program Elements Layer consists of the Code and Action packages providing programming elements such as data types, data items, classes, procedures, macros, prototypes, templates and captures the low level behavior elements of applications, including detailed control and data flow between statements. The Level 1 (L1) cover the Resource Layer which represents the operational environment of the existing software system. For example, the knowledge related to events and state-transition, the knowledge related to the user interfaces of the existing software system and the knowledge related to persistent data, such as indexed files, relational databases, and other kinds of data storage. The Level 2 (L2) cover the Abstraction Layer which represents domain and application abstractions.

As we stated earlier, herein we are only interested in the Program Element Layer - more specifically in the Code Package, which represents the code elements of a program and their associations. Therefore, it is important to dig a little deeper in this metamodel because it is mainly used by our approach in order to identify concerns.

In a given KDM instance, each instance of the code metamodel element represents some programming language construct, determined by the programming language of the existing software system. Each instance of a code meta-model element corresponds to a certain region of the source code in one of the artifacts of the existing software system. In addition, the Code package consists of 24 classes and contains all the abstract elements for modeling the static structure of the source code. However, we are particularly interested in some of them. In Figure 4 is depicted a chunk of the Code package. It worth to notice that the more important metaclasses used herein are highlighted.

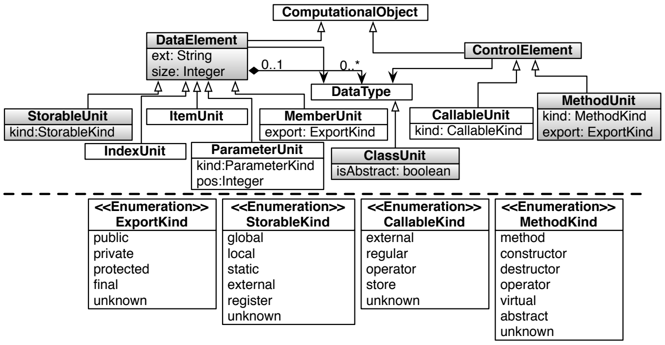

Figure 4: Chunk of the Code Package (OMG Group [?])

As can be seen in Figure 4 the root metaclass is *ComputationalObject* which has two sub-metaclasses, i.e., *DataElement* and *ControlElement*. The former sub-metaclass, *DataElement*, is a generic modeling element that defines the common properties of several concrete classes that represent the named data items of existing software systems, for example, global and local variables, record files, and formal parameters. *DataElement* has five sub-metaclasses - *StorableUnit*, *IndexUnit*, *ItemUnit*, *ParameterUnit* and *MemberUnit*. *StorableUnit* is a concrete sub-metaclass of the *StorableElement* meta-class that represents variables of the existing software system. *IndexUnit* class is a concrete subclass of the *DataElement* class that represents an index of an array datatype. Instances of *ItemUnit* class are endpoints of KDM data relations which describes access to complex datatypes. *ParameterUnit* class is a concrete subclass of the *DataElement* class that represents a formal parameter; for example, a formal parameter of a procedure. *MemberUnit* class is a concrete subclass of the *DataElement* class that represents a member of a class type. Finally, the latter, *ControlElement* is a sub-metaclass that contains two sub-metaclasses - *MethodUnit* and *CallableUnit*. *MethodUnit* element represents member functions owned by a *ClassUnit*, including user-defined operators, constructors and destructors. The *CallableUnit* represents a basic stand-alone element that can be called, such as a procedure or a function. As can be seen below the dashed line in Figure 4 there are also the following enumerations: "*ExportKind*", "*StorableKind*", "*CallableKind*", "*MethodKind*", which are sets os literals used as properties of the metaclasses.

2.1.3 Structured Metrics Metamodel - SMM

SMM (Structured Metrics Metamodel) provides a standard format to define metrics and to represent the measurement results. The main terms in SMM are measures and measurements. Measures are models which describes a way to calculate properties of software. For measurements can understand models containing results of applications of the metrics.

To apply SMM models in KDM models, a tool is needed to automate the process. The process occurs as illustrated in Figure X, the tool must receive an SMM model and a KDM model as input (KDM represents the software to be measured), then, after the measurements, other SMM model is generated as output and this model contains the results of the process.

There are some advantages in using SMM. It is advantageous because it is a standard, so, metrics defined in accordance with this metamodel can be tested, reused and shared. Furthermore, the use of a standard facilitates the development of compatible tools for automating the measurement process.

The SMM metrics can be of various types; direct, collective, binaries, etc. Direct metrics perform measurements of an index in a model. Collective metrics may measure multiple components and perform an operation with the result, for example; sum, average and others.

2.2 Motivation

Systematic review studies belong to Evidence-Based Software Engineering (EBSE) paradigm [36]. They provide new, empirical and systematic methods of research. Although several studies have been reported in the broader context of Architecture Driven Modernization (ADM) [9,11,32,33], to the best of our knowledge any systematic review has been conducted in this field. 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. As result, this paper is meant to furnish to ADM through a systematic review and evidence-based approach. Also we argue that this paper may help researchers in the field of modernization of legacy systems, once the paper provides an overview of the current state-of-the-art of the ADM. Furthermore, it may serve as a first step towards more thorough examination of the topics addressed in it with the help of systematic literature reviews.

3. THE SYSTEMATIC REVIEW

This study has been undertaken as a systematic review based on the guidelines proposed by Kitchenham and Brereton [15]. According to them, in order to conduct a systematic review, it is advisable to follow three main phases: (*i*) planning the review, (*ii*) conducting the review and (*iii*) reporting the review. In Figure 5 depicts these three phases that we carried out herein. Furthermore, in this paper we have used Visual Text Mining (VTM) technique to support the studies selection [17]. VTM uses text mining algorithms and methods combined with interactive visualisations. Therefore, it can help the user making sense of a collection of primary studies, without actually reading all of them. In this case the studies were reading partially or full. The following sections present details on how each phase was carried out.

3.1 Planning the Systematic Review

In this phase we have 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.

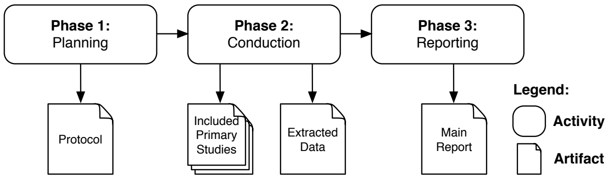

Figure 5: Systematic review process (Adapted from Kitchenham [15]).

Research questions must embody the review study purpose. Moreover, these questions reflect the general scope of the review study. The scope is comprised of population (i.e., population group observed by the intervention), intervention (i.e., what is going to be observed in the context of the planned review study), and outcomes of relevance (i.e., the results of the intervention). Furthermore, during the conduction of this step, it was also necessary to establish the scope of the review study. According to the systematic review process [15], the scope has to be established using the PICO criteria. Thus, herein our **Population** is published scientific literature reporting on the use of Architecture-Driven Modernization and its metamodels. The **Intervention** is published scientific literature interested with Architecture-Driven Modernization and its metamodels. The **Comparison** is not applied herein. Finally, the **Outcomes of relevance** is an overview of the studies that have been conducted in the field of Architecture-Driven Modernization and its metamodels, emphasizing primary studies that report on the process used in the research area, from observing such an aggregated data set, we also intend to provide insight into the frequencies of publication over time to inspect trends.

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 objectives we worked out four research questions. The questions are:

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?

RQ$_2$: Given the ADM's standards metamodels, which one has been more used in the literature?

RQ$_3$: Which research types have been employed into the field herein?

RQ$_4$: Which avenues are often used to publish research related to ADM?

To address **RQ$_1$**, we read all primary studies in order to identify the topic of each study. Next, we arrange all studies according to them topics. Whether any kind of disagreement related to the topic that the article meets, the article was marked and was discussed with everyone involved in the review in order to clarify which topic it belongs. With respect to **RQ$_2$**, we also read all primary studies and identified which metamodel was used in the study. Concerning to **RQ$_3$**, we gathered all references of each primary studies herein. Afterwards, we organized them among journal, conference, and workshop. Finally, with respect to **RQ$_4$**, we used and adapted the scheme proposed by Wieringa et al [36] in order to classify each primary studies into a research type, see Section 3.2.3.

Afterwards, we have defined the search string and chosen the electronic databases. The search string was created based upon the following keywords: *Architecture Driven Modernization, Architecture-Driven Modernization, ADM, Object Management Group, OMG, Legacy Systems, Knowledge Discovery Metamodel, KDM, Abstract Syntax Tree Metamodel, ASTM, Structured Metrics Metamodel - SMM, Model-Driven Development*. A sophisticated search string was constructed using boolean operators i.e., *AND*, *OR* and *NOT*. Figure 6 shows the search string elaborated. The search have encompassed electronic databases which are deemed as the most relevant scientific sources [6] and therefore likely to contain important primary studies. We have used the search string on the following electronic databases: *ACM*, *IEEEXPLORE*, *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 6 was actually used to construct a semantically equivalent string specific to each database.

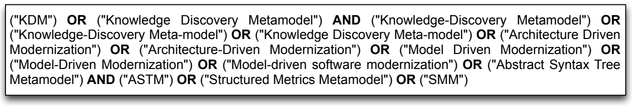
("KDM") **OR** ("Knowledge Discovery Metamodel") **AND** ("Knowledge-Discovery Metamodel") **OR** ("Knowledge-Discovery Meta-model") **OR** ("Knowledge Discovery Meta-model") **OR** ("Architecture Driven Modernization") **OR** ("Architecture-Driven Modernization") **OR** ("Model Driven Modernization")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

Figure 6: Search String.

Then, in order to determine which primary studies are relevant to answer our research questions, we have applied a set of inclusion and exclusion criteria. Inclusion criteria devised and applied are:

(a) **The primary study presents at least one solution of modernization by means of ADM:** the paper provides evidences that the ADM assists the software engineer during the modernization/refactoring of legacy system.

(b) **Studies that explicitly present an ADM approach:** the paper provides an approach to assist the software engineer to modernize the legacy system.

(c) **The primary study presents at least one type of evaluation technique for ADM:** without the results of the evaluation we would not be able to make comparisons desired. In other words, the paper must clearly present which assessment techniques have been employed to evaluate the ADM process, i.e., case study, experiment, survey, etc.

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 devised and applied are:

(a) **Papers which mentioned ADM and its metamodels in the abstract only:** this was required because we found many studies that mentioned ADM and its metamodels in their opening sentences as a principal concept, however, the studies did not really address it.

(b) **Papers that present only recommendations, guidelines or principles:** usually this kind of papers do not present practical approach to modernize legacy systems.

(c) **Introductory papers for books and workshops.**

(d) **The primary study is a short paper:** papers with two pages or less were not considered herein, since we considered that this kind of study do not own sufficient information.

(e) **Papers from industrial conferences, and non English publications.**

We devised 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 are 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; the achieved results were crossed and then validated. All the results of the search process are documented in the web material[1]. Therefore, it is clear to others how thorough the search was, and how they can find the same documents.

3.2 Conducting the Systematic Review

In this phase, firstly we identified primary studies in the digital libraries, i.e., IEEE, ACM, Scopus, Web of Science and Engeneering Village. We applied the search string given in Figure 6. Thus, an overview of results acquired from these digital libraries is depicted in Table 1.

Table 1: 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

As can be seen in Table 1 the digital libraries Scopus has returned more primary studies than the others (150), i.e., ACM, Engeneering Village, Web of Science and IEEE returned 51, 30, 17 and 11, respectively. Possibly, this came about because Scopus indexes studies of others libraries, such as IEEE and ACM. Summing up, we have gotten 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 means of 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 have gotten 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 (see Section 3.1).

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 and systematic mapping [5,19]. Although, this classification schemes has been proposed to be applied during a systematic mapping, we claim that it can be used during the systematic review process as well. Thus, the aforementioned categories were adapted to specifics of our systematic study. The resultant classification schemes are as follows:

3.2.1 Focus Area

We have used the keywording method described in [31] to identify the focus area of the studies identified. As result we got four ones - a brief description of each focus area follows:

- **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.
- **Business Knowledge Extraction:** Describes primaries studies which address process, method or approach to extract business process of a legacy system.
- **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, e.g., extension of KDM, SMM, ASTM, etc.
- **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 of ADM and its metamodels, e.g., papers that show how to use the metamodels (KDM, SMM and ASTM) and provide a basis for the further research, etc.

3.2.2 Contribution Type

During the reading of the primary studies was possible to identify a set of contribution type related to ADM and its metamodels. More precisely we identified five contribution type:

- **Tool:** This contribution type refers to the primary studies found that focus on providing tool in order 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 environment.
- **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.

[1] http://tinyurl.com/99spmaz

- **Model Transformation:** It refers to contributions which specifically describe transformation among ADM's metamodels, e.g., primary studies that describe the use of language transformation such as either Query/Views/-Transformations (QVT)[2] or ATL Transformation Language (ATL)[3].

- **Metamodel:** This contribution type describes primary studies which either created or extended 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.

- **Metrics:** This is type of contributions focus on proposing or applying metrics to effectiveness of ADM and its metamodels

3.2.3 Research Type

The research type reflects the research approach used in the primary study. Therefore, we used and adapted the scheme proposed by Wieringa et al [36]. A brief description of research types are as follows:

- **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,mathematical analysis, etc.

- **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.

- **Conceptual proposal:** A conceptual proposal presents an arrangement to see 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.

- **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.

- **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.

3.3 Validation

In validation phase an approach that uses VTM technique and the associated tool - Projection Explorer (PEx) - were applied to support the inclusion and exclusion decisions [17].

Figure 7 presents a document map generated using PEx. This map is composed of 259 primary studies analysed in this review, highlighting them using different shades of gray to differentiate in which of the stages a study was removed from the review. White points are studies excluded in first stage, gray points are the studies excluded in second stage and the black points are the included. The exploration of a document map is conducted in two steps: (*i*) firstly, a clustering algorithm is applied to the document map, creating groups of highly related documents; (*ii*) secondly, the resulting clusters are analysed in terms of: **Pure Clusters** - all documents belonging to a cluster have the same classification (all included or excluded, regardless of exclusion stage). Normally, in this case do not need to be reviewed; and **Mixed Clusters** - which represent documents with different classification on the same cluster. These cases are hints to the reviewer, and the estuaries grouped should be reviewed following the traditional method. To facility the visualisation, in Figure 7 just five clusters generated by PEx are depicted. Examples of pure clusters (all excluded) are identified in Figure 7 using label "(a)" and therefore do not needed to be reviewed. Mixed clusters (clusters containing black (included) and white or gray (excluded) studies) are identified using label "(b)" and they were reviewed by the authors of this paper. At the end, we kept the initial classifications conducted manually, but this technique contributed to a review of studies that could have been wrongly excluded or included previously.

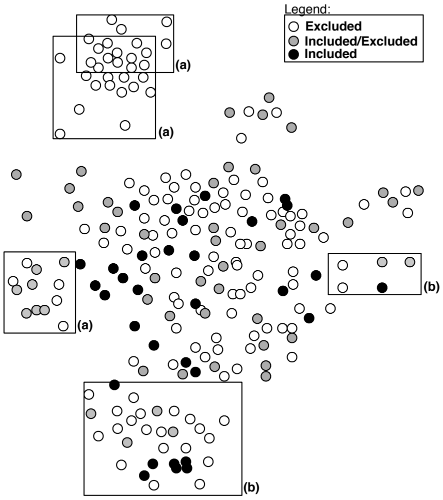

Figure 7: Document map colored with the history of the inclusions and exclusions of the studies.

3.4 Reporting the Systematic Review

The focus of this section is to present the broad overview of research within ADM and its metamodels we have acquired after conducting the systematic review. Moreover, we used information drawn from this overview to answer this review study's research questions. Notice that we present our results in twofold: (1) what are the focus area most discussed and least discussed in the literature regarding the ADM and its metamodels, and contribution types ($\mathbf{RQ}_1$), research type ($\mathbf{RQ}_3$), among metamodels which is more used to assist the modernization ($\mathbf{RQ}_2$) and (2) avenues used for publishing the related research ($\mathbf{RQ}_4$).

Aiming to show a part of the first dimension of our results and also the frequencies of all publication related to ADM

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

[3] www.eclipse.org/atl/

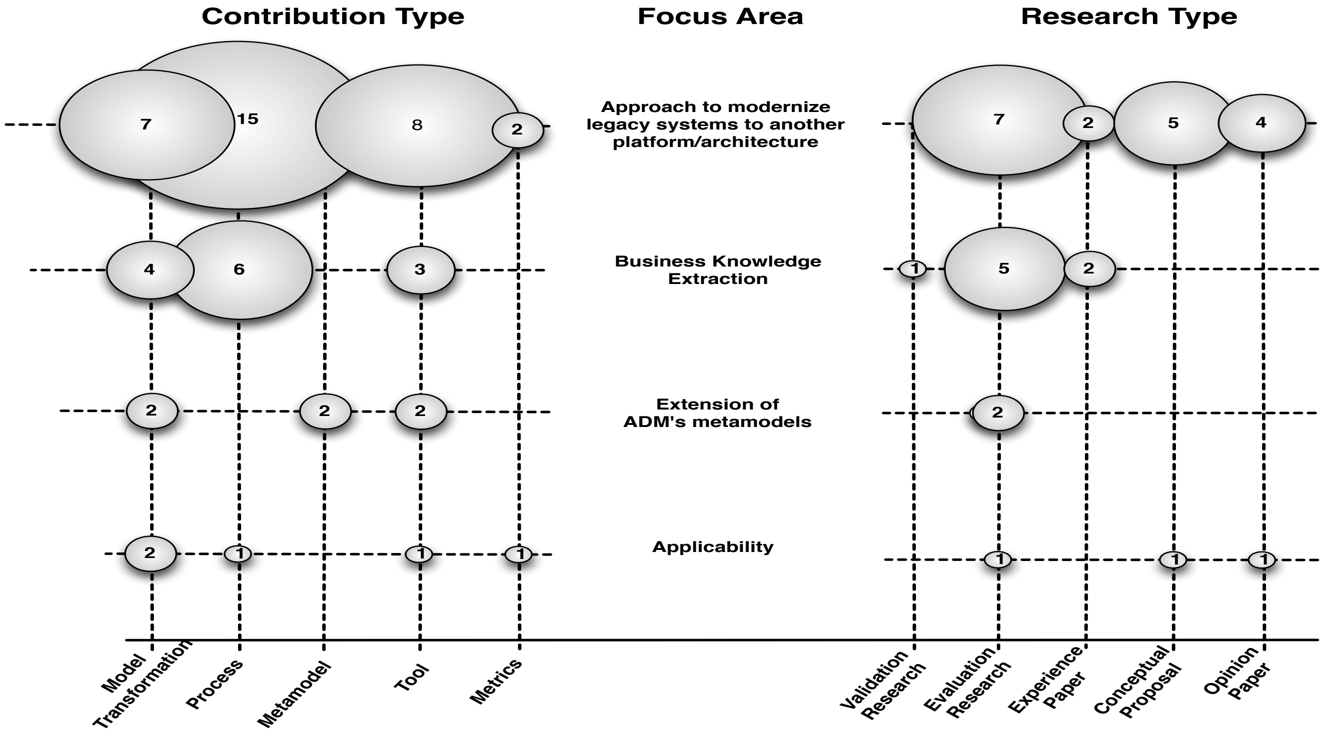

Figure 8: Map of research focus on ADM and its metamodels.

and its metamodels we plotted a bubble plot, which is depicted in Figure 8. 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 8 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 in Table 1 (30).

It is fairly evident from observing Figure 8 that 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 came about once the majority of the primary studies found which describe a process to assist the modernization of legacy systems usually propose a set of 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 39%, 26% and 25%, respectively of the current research (see Figure 9). On the other hand, **Metamodel** and **Metrics** have been addressed by a very small number of publications i.e., 3.57% and 5.35%, respectively (see Figure 9).

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 3.4.1-3.4.4. We have 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 $\mathbf{RQ}_2$ we analyzed individually all identified primary studies focus on gather which ADM standard metamodels have more been used in the literature. In Figure 10 is depicted a pie chart wherein we have 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

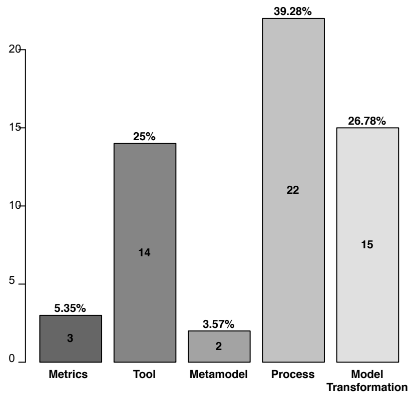

Figure 9: Frequency of studies in each category.

6.66%. Finally, we found out 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.

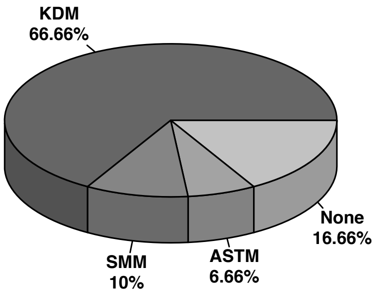

Figure 10: Frequency of ADM's metamodels used in literature.

In Figure 11 shows the data we gathered related to the research type employed in the field of ADM. As far as the research type is concerned, **Evaluation Research** are in vast majority, covering over 48%, see Figure 11. A small percentage of publications have reported on **Validation Research** and **Experience Paper**, i.e., 3.22% and 12.90%, respectively. While **Conceptual Proposal** and **Opinion Paper** have been presented collectively by rather percentage of 35.47%.

Aiming to answer $\mathbf{RQ}_4$ we collected the name of the conference, name of the journal and the name of the workshop which the primary studies were published, Table 2 covers the second dimension of our results and shows all avenues identified during the conducting of this review. It is fairly evident from observing the Table 2 that the majority of primary studies were published in conference. Therefore, as result the majority of the data related to the systematic review herein were extracted from conference, i.e., among the identified primary studies (30) 23, i.e., 76.66% studies were published in conference. Even though research has appeared on different conference we can see that in Table 2 that we found out six journals during the review, which represent 20%, i.e., 6 of the primary studies identified and gathered. The table also reveals that there is only one publication related to ADM and its metamodels that has appeared in a workshop so far.

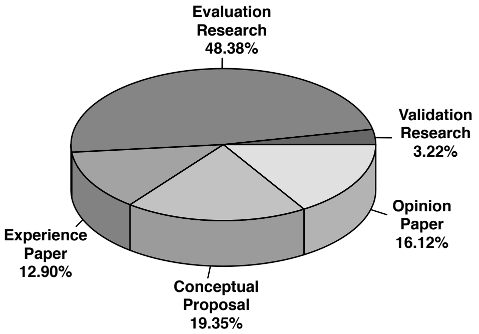

Figure 11: Frequency of research type.

Finally, in Figure 12 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 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 also 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.

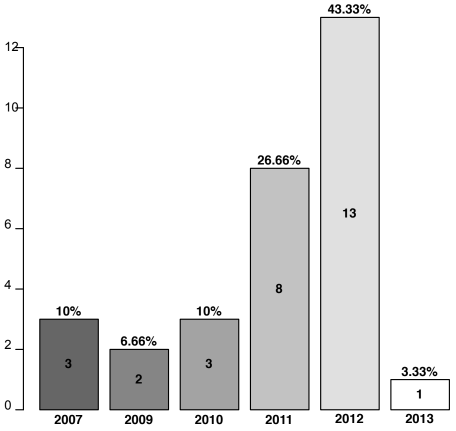

Figure 12: Distribution of publication by years.

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

In this section we briefly describes different studies regarding to propose approaches to modernize legasy systems by means of ADM.

Jorge Maratalla et al., [20] propose **GAFEMO** (Gap Analysis Framework That Integrating the Principles of Architectural Driven Modernization) which aims to modernize a legacy systems using a model-driven and service-oriented approach taking advantage of the features provided by gap-analysis techniques. To evaluate **GAFEMO** the authors

Table 2: Overview of publication avenues for selected primary studies.

No.	Avenue	Quant	Primary Stydy
Journal			
1	IET Software	1	[30]
2	Journal of Systems and Software	1	[32]
3	Journal of Software: Evolution and Process	1	[9]
4	Science of Computer Programming	1	[11]
5	IEEE Software	1	[14]
6	Computer Standards & Interfaces	1	[33]
Conference			
7	SRII Global Conf.	1	[20]. [27]
8	ACM Symposium on Applied Computing (ACM SAC)	3	[25], [26], [4]
9	International Conf. on Software Maintenance (ICSM)	2	[37]
10	International Conf. on Engineering of Complex Computer Systems (ICECCS)	1	[1]
11	International Conf. on Conceptual Modeling (ER)	1	[18]
12	Working Conf. on Reverse Engineering (WCRE)	4	[13], [29], [12], [2]
13	European Conf. on Software Maintenance and Reengineering (CSMR)	2	[8], [28]
14	International Conf. on Enterprise Information Systems (ICEIS)	1	[23]
15	International Conf. on Information Systems and Technologies (ICIST)	1	[22]
16	Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium (TMS)	1	[35]
17	International Conf. on Model Transformation (ICMT)	2	[24], [7]
18	International Conf. on Automated Software Engineering	1	[3]
19	International Conf. on Computer Systems and Technologies (CompSysTech)	1	[21]
20	International Conf. on Web Engineering (ICWE)	2	[16], [34]
Workshop			
21	Workshop Software-Reengineering (WSR)	1	[10]

have successfully applied it to a real case study from the public administration.

Yazdanshenas and Moonen [37] present a method that combines model-driven engineering with program analysis techniques to support analysis across the components of a component-based system. They build 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 our analysis. Finally, they also have implemented and evaluated our approach by building a prototype tool which has been successfully used to track information flow in a component-based system using program slicing.

In [18] the authors propose a modernization approach for the development/modernization of Data warehouses. The approach is based on ADM in order to analyze the operational data sources and obtain a PSM whose elements are marked with multidimensional concepts.

Guzmán and Piattini [13] 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. Their proposal is particularly focused on relational databases as a very extended and valuable type of legacy system.

Frey and Hasselbring [8, 9] propose 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.

Pérez-Castillo et al., [23] proposes the elicitation of database schemas, a reverse engineering technique that follows the ADM process to recover a minimal relational schema from the queries embedded in the legacy source code. In [4] 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] also put forward 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 [11] 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., [16] present an approach that allows developers to automatically modernize the client side of legacy systems. In this approach developers can refactor the User Experience of legacy systems during the modernization journey, taking the opportunities offered by novel interaction paradigms, i.e., Rich Internet Application (RIA). Similarly, in [34] the authors present an approach for the definition of a systematic process for Web Applications (WA) to Rich Internet Application (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.

In [12] the authors propose an approach that uses ADM and its metamodels which is focused on the analysis of legacy systems to discover and create functionalities to be exposed as services using Web Services. Boussaidi et al., [2] 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 [14] 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.

3.4.2 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., [26–28] present MARBLE, a modernization tool to recover business processes from legacy systems, as well as two process mining techniques within MARBLE based on static, i.e., only considers legacy source code's elements from a syntactical viewpoint and dynamic analysis of source code, i.e., it considers information derived by system execution. In [32] the authors report the results of a family of case studies that were performed to empirically validate MARBLE using analysis and meta-analysis techniques. The family of case studies demonstrates that the technique is feasible in terms of effectiveness and efficiency. Pérez-Castillo et al., also provides in [30] a semi-automatic technique based on dynamic analysis, combined with static analysis to instrument the source code for obtaining event log models. A set of model transformation to transform the event log into another model following the KDM to depict legacy system, concerning its runtime viewpoint, which can be used in any software modernisation project. In [24] Pérez-Castillo et al., explain the KDM2BPMN model transformation within MARBLE, an ADM-based framework to rebuilt business processes embedded in legacy systems in order to facilitate and improve the evolutionary maintenance. More specifically, MARBLE defines four abstraction levels (from L0 to L3) that represent four different kinds of models as well as three model transformations between the levels

Normantas and Vasilecas [22] present an approach that facilitates software comprehension by enabling traceability of implementation of business rules and business scenarios in the 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., [7] 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. In other words, 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. The authors carried out an experiment and found out that the set of rules are able to obtain KDM event models from MXML logs in a scalable and suitable way.

3.4.3 Extension of ADM's metamodels

We have identified two papers that address how to perform extension of ADM's metamodels. However, the identified paper related to this focus area are well explained. We provided a brief summary of these paper are follows.

In [1] proposes 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 [25] propose an extension to the KDM. This extension 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.

3.4.4 Applicability

Similarly, we also have identified a small number of papers that address just the applicability of the ADM and its metamodel. A summary of these paper are follows.

Normantas et al., [21] describe all concepts related to the metamodel KDM. The authors claim that the paper enables researchers and practitioners to get a better understanding KDM. This paper also provide a reference for the researchers and practitioners that are interested in using the KDM for the purpose of modernization, and would establish a basis for the further research.

Pérez-Castillo et al., [33] present how to use KDM to modernize legacy systems, making them more agile, preserving the embedded business knowledge and reducing maintenance costs. 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.

4. THREATS TO VALIDITY

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. Nevertheless, this threat was mitigated by selecting search engines which have been regarded as the most relevant scientific sources [6] and therefore prone to contain the majority of the important studies.

Reviewers reliability. All the reviewers of this study are researchers in the software reuse field, focused on the aspect-oriented programming, software testing and software product line, and none of the techniques and tools developed by us. Therefore, 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. Therefore, in order to ensure the validity, multiple sources of data were analyzed, i.e. papers, technical reports, white papers. Furthermore, in the event of a disagreement between the two primary reviewers, a third reviewer acted as an arbitrator to ensure full agreement was reached.

5. 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 [6]. Through a examination of 30 primary studies encompassing ADM, this review has presented 22 papers which process, 15 that propose a set of model transformation, 14 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 8 can be greater than the total of selected studies presented in Table 1 (30). 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, etc to modernize a legacy systems within their organizations.

The initial significant contributions to ADM were presented in 2007 (i.e., [12,13,18]), in 2008 had no publications. However, in 2009 we identified two paper dealing with ADM (i.e., [4, 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.

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 int the literature.

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.

6. ACKNOWLEDGMENTS

Rafael Durelli would like to thank the financial support provided by FAPESP, process number 2012/05168-4. Daniel Santibáñez would like to thank the financial support provide by CAPES. Valter Camargo would like to thank CNPQ, process number 560241/2010-0. We also thank Thiago Gottardi for his proofreading of this paper.

7. REFERENCES

[1] L. Baresi and M. Miraz. A component-oriented metamodel for the modernization of software applications. In *Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on*, pages 179–187, 2011.

[2] G. Boussaidi, A. Belle, S. Vaucher, and H. Mili. Reconstructing architectural views from legacy systems. In *Reverse Engineering (WCRE), 2012 19th Working Conference on*, pages 345–354, 2012.

[3] H. Bruneliere, J. Cabot, F. Jouault, and F. Madiot. Modisco: a generic and extensible framework for model driven reverse engineering. In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, ASE '10, pages 173–174. ACM, 2010.

[4] R. P. del Castillo, I. García-Rodríguez, and I. Caballero. Preciso: a reengineering process and a tool for database modernisation through web services. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 2126–2133, New York, NY, USA, 2009. ACM.

[5] R. S. Durelli, D. S. M. Santibáñez, N. Anquetil, M. E. Delamaro, and V. V. de Camargo. A systematic review on mining techniques for crosscutting concerns. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1080–1087, New York, NY, USA, 2013. ACM.

[6] T. Dyba, T. Dingsoyr, and G. K. Hanssen. Applying systematic reviews to diverse study types. In *International Symposium on Empirical Software Engineering and Measurement*, ESEM '07, pages 225–234, Washington, DC, USA, 2007. IEEE Computer Society.

[7] M. Fernández-Ropero, R. Pérez-Castillo, B. Weber, and M. Piattini. Empirical assessment of business model transformations based on model simulation. In *Proceedings of the 5th international conference on Theory and Practice of Model Transformations*, ICMT'12, pages 137–151, Berlin, Heidelberg, 2012. Springer-Verlag.

[8] 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 (CSMR), 2011 15th European Conference on*, pages 269–278, 2011.

[9] 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.

[10] S. Frey, A. van Hoorn, R. Jung, B. Kiel, and W. Hasselbring. Mamba: Model-based software analysis utilizing omg's smm. In *Workshop Software-Reengineering (WSR '12)*, pages 37–38, May 2012.

[11] R. Fuentes-Fernandez, J. Pavon, and F. Garijo. A model-driven process for the modernization of component-based systems. *Science of Computer Programming*, 77(3):247 – 269, 2012.

[12] I. Garcia-Rodriguez de Guzman. Pressweb: A process to reengineer legacy systems towards web services. In *Reverse Engineering, 2007. WCRE 2007. 14th Working Conference on*, pages 285–288, 2007.

[13] I. G.-R. d. Guzman, M. Polo, and M. Piattini. An adm approach to reengineer relational databases towards web services. In *Proceedings of the 14th Working Conference on Reverse Engineering*, WCRE '07, pages 90–99, Washington, DC, USA, 2007. IEEE Computer Society.

[14] J. Izquierdo and J. Molina. An architecture-driven modernization tool for calculating metrics. *Software, IEEE*, 27(4):37–43, 2010.

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

and Software Technology, 51:7–15, January 2009.

[16] L. Mainetti, R. Paiano, and A. Pandurino. Migros: a model-driven transformation approach of the user experience of legacy applications. In *Proceedings of the 12th international conference on Web Engineering*, ICWE'12, pages 490–493, Berlin, Heidelberg, 2012. Springer-Verlag.

[17] V. Malheiros, E. Hohn, R. Pinho, M. Mendonca, and J. C. Maldonado. A visual text mining approach for systematic reviews. In *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement*, ESEM '07, pages 245–254, Washington, DC, USA, 2007. IEEE Computer Society.

[18] J.-N. Mazón and J. Trujillo. A model driven modernization approach for automatically deriving multidimensional models in data warehouses. In *Proceedings of the 26th international conference on Conceptual modeling*, ER'07, pages 56–71, Berlin, Heidelberg, 2007. Springer-Verlag.

[19] A. Mehmood and D. N. A. Jawawi. Aspect-oriented model-driven code generation: A systematic mapping study. *Inf. Softw. Technol.*, pages 395–411, 2013.

[20] 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 Conference (SRII), 2012 Annual*, pages 343–352, 2012.

[21] K. Normantas, S. Sosunovas, and O. Vasilecas. An overview of the knowledge discovery meta-model. In *Proceedings of the 13th International Conference on Computer Systems and Technologies*, pages 52–57, New York, NY, USA, 2012. ACM.

[22] K. Normantas and O. Vasilecas. Extracting business rules from existing enterprise software system. In *Information and Software Technologies*, Communications in Computer and Information Science, pages 482–496. Springer Berlin Heidelberg, 2012.

[23] R. Pérez-Castillo, I. G. R. de Guzmán, D. Caivano, and M. Piattini. Database schema elicitation to modernize relational databases. In *ICEIS*, pages 126–132, 2012.

[24] 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.

[25] R. Pérez-Castillo, I. G.-R. de Guzmán, M. Piattini, and B. Weber. Integrating event logs into kdm repositories. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1095–1102, New York, NY, USA, 2012. ACM.

[26] R. Pérez-Castillo, I. G.-R. de Guzmán, M. Piattini, B. Weber, and A. S. Places. An empirical comparison of static and dynamic business process mining. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 272–279, New York, NY, USA, 2011. ACM.

[27] R. Perez-Castillo, M. Fernandez-Ropero, I. Garcia-Rodriguez de Guzman, and M. Piattini. Marble. a business process archeology tool. In *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*, pages 578–581, 2011.

[28] R. Perez-Castillo, I. Garcia-Rodriguez de Guzman, and M. Piattini. Mimos, system model-driven migration project. In *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*, pages 445–448, 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 *Reverse Engineering, 2009. WCRE '09. 16th Working Conference on*, pages 128–132, 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*, 5(3):304–319, 2011.

[31] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pages 68–77, 2008.

[32] R. PÃ©rez-Castillo, J. A. Cruz-Lemus, I. G.-R. de GuzmÃ¡n, and M. Piattini. A family of case studies on business process mining using marble. *Systems and Software*, 85(6):1370 – 1385, 2012.

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

[34] R. Rodríguez-Echeverría, J. M. Conejero, P. J. Clemente, J. C. Preciado, and F. S$#225;nchez-Figueroa. Modernization of legacy web applications into rich internet applications. In *Proceedings of the 11th international conference on Current Trends in Web Engineering*, pages 236–250, Berlin, Heidelberg, 2012. Springer-Verlag.

[35] Z. Tu, G. Zacharewicz, and D. Chen. Harmonized and reversible development framework for hla based interoperable application. In *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, pages 51–58, San Diego, CA, USA, 2011.

[36] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. pages 102–107, 2005.

[37] A. R. Yazdanshenas and L. Moonen. Crossing the boundaries while analyzing heterogeneous component-based software systems. In *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*, pages 193–202, 2011.