A Mapping Study on Architecture-Driven Modernization

Rafael S. Durelli[†‡], Daniel S. M. Santibáñez[*], Bruno Marinho[*], Raphael Honda[*],
Márcio E. Delamaro[†], Nicolas Anquetil[‡] and Valter Vieira de Camargo[*]
[*]Departamento de Computação, Universidade Federal de São Carlos,
Caixa Postal 676 – 13.565-905, São Carlos – SP – Brazil
Email: {daniel.santibanez, bruno.marinho, raphael.honda, valter}@dc.ufscar.br
[‡]RMoD Team - INRIA, Lille – France
Email: {Nicolas.Anquetil}@inria.fr
[†]Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo,
Av. Trabalhador São Carlense, 400, São Carlos – SP – Brazil
Email: {rdurelli, delamaro}@icmc.usp.br

***Abstract*—Background: Perhaps the most common of all software engineering activities is the modernization of software. Unfortunately, during such modernization often leaves behind artifacts that are difficult to understand for those other than its author. Thus, the Object Management Group (OMG) has defined standards in the modernization process, by creating the concept of Architecture-Driven Modernization (ADM). Nevertheless, to the best of our knowledge, there is no a systematic mapping study providing an overview of how researchers have been employing ADM. Thus, we assert that there is a need for a more systematic investigation of the topics encompassed by this research area. Objective: To describe a systematic mapping study on ADM, highlighting the main research thrusts in this field. Method: We undertook a systematic mapping study, emphasizing the most important electronic databases. Results: We identified 30 primary studies, which were classified by their contribution type, focus area, and research type. Conclusion: This systematic mapping can be seen as a valuable initial foray into ADM for those interested in doing research in this field. More specifically, our paper provides an overview of the current state of the art and future trends in software modernization area, which may serve as a road-map for researchers interested in coming up with new tools and processes to support the modernization of legacy systems.**

***Index Terms*—Systematic Mapping, Architecture-Driven Modernization, ADM, Knowledge Discovery Metamodel, KDM**

I. Introduction

Software systems cannot be simply discarded because they incorporate a lot of invaluable knowledge about their organizations. However, the structure of these software systems starts deteriorating when they undergo maintenance. These systems are considered legacy when their maintenance costs are raised to undesirable levels but they are still valuable assets to their organizations. Given the significant business value of these systems, they must be modernized.

In this context, the Object Management Group (OMG) has defined standards in the modernization process, creating the concept of Architecture-Driven Modernization (ADM). ADM follows the Model-Driven Development (MDD) [1] guidelines and comprises three major steps. First, reverse engineering is performed starting from the source-code. This step yields an instance model named Platform-Specific Model (PSM). Second, successive transformations are applied to this model, up to point at each of these transformations reaches an apt abstraction level. These transformations are represented using a model called Knowledge Discovery Metamodel (KDM). Using this model, several modernization activities, optimizations, and modifications can be performed in order to solve problems found in the legacy system. Third, a forward engineering step is carried out, resulting in a modernized version of the source code of the target system. The idea behind the KDM standard is to provide a common representation, thereby increasing portability by allowing the community to create parsers that turn any representation into KDM. As a result, as long as developers stick to such a standard, 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 gauge and report on the current state of the art and research trends in this field, we carried out a systematic mapping study on ADM. Our motivation to conduct such a systematic mapping study is to identify the topics that have been most investigated as well as the topics that have not received much attention. Although, ADM is a relatively new approach, OMG claims that it is a step towards combining two well-known research fields, (i) MDD and *(ii)* software reengineering. Since ADM has come a long way in the last few years and many efforts have emphasized the modernization of legacy systems through this approach, we assert that there is a need for a more systematic investigation of the topics encompassed by this research area. To the best of our knowledge, this is the first systematic mapping study on ADM, providing an initial foray into the current state-of-the-art of this field.

This paper is structured as follows. Section II describes how the systematic mapping methodology has been conducted. Section III presents the main findings of this study. In Section IV the threats to validity of this study are presented. Finally in Section V concluding remarks are made.

IEEE IRI 2014, August 13-15, 2014, San Francisco, California, USA
978-1-4799-5880-1/14/$31.00 ©2014 IEEE

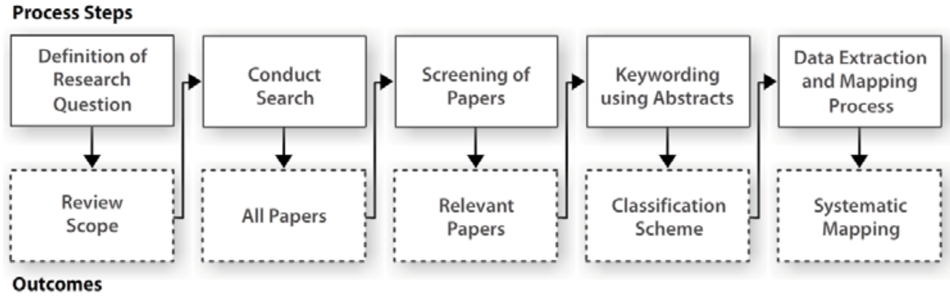

Figure 1. The systematic mapping process (Adapted from [2]).

II. Research Methodology

A mapping study provides a systematic and objective procedure for identifying the nature and extent of the available research relevant to answer a particular research question. Throughout the conduction of this systematic mapping, we followed the all guidelines provided by Petersen et al. [2]. Each step produces an intermediate outcome, the concluding result being the mapping study as shown in Figure 1.

Furthermore, in this paper we have used Visual Text Mining (VTM) technique to support the studies selection [3]. 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 we carried out this mapping study.

A. Search Strategy

The review protocol is defined in this step. This protocol contains: *(i)* the research questions (RQs) and *(ii)* the search string.

RQs must embody the mapping study purpose. Thus, given that we set out to give an overview of the current state of the art and research in the ADM field, we formulated three research questions:

- **RQ**$_1$ - Given ADM's standards metamodels, which one has been more used in the literature? In addition, given the identified metamodel, what are the most and least used packages?
- **RQ**$_2$ - What types of studies have been published in the area?
- **RQ**$_3$ - What are the most and least discussed focus areas in the ADM literature? Moreover, what types of contributions have been presented so far?

Afterwards, the search string was defined. The search string was created based upon a set of keywords. Figure 2 shows the search string we used to carry out our systematic mapping.

B. Data Source and Study Selection

The search encompassed electronic databases that are deemed as the most relevant scientific sources [4] 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*. It is worth mentioning that since the features provided by these databases, as well as the exact syntax of the search strings to be applied, vary from one database to the other, the string given in Figure 2 was actually used to construct a semantically equivalent string tailored 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 2. Search string used in our systematic mapping.

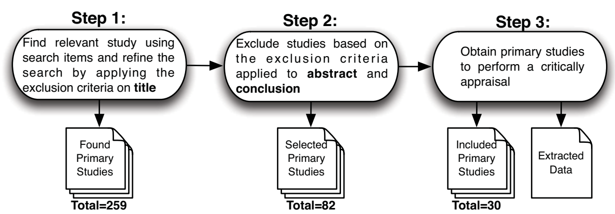

Figure 3. Study Selection Summary.

The search string given in Figure 2 was applied in the digital libraries. An overview of results acquired from these digital libraries is depicted in Figure 3. Moreover, it presents the amount of studies remaining after each step.

In order to determine which primary studies are relevant to answer our research questions, we applied a set of inclusion and exclusion criteria. The inclusion criteria we applied were the following:

- The primary study presents at least one modernization approach that employs ADM;
- The primary study describes an empirical evaluation of an ADM-based approach.

and the following exclusion criteria:

- Papers that mention ADM and its related metamodels only in the abstract;
- Introductory papers for books and workshops;
- The primary study is a short paper (containing up to three pages).

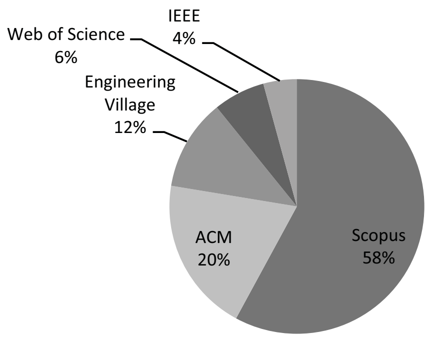

Figure 4. Distribution of primary studies by electronic database.

As can be seen in Figure 4 Scopus was the digital library that returned most primary studies 58% (150). ACM, Engineering Village, Web of Science and IEEE returned 20% (51), 12% (30), 6% (17), and 4% (11), respectively. We surmise that this occurred because Scopus indexes studies from others libraries, such as IEEE and ACM. Summarizing, we obtained 259 primary studies in the first step. After the first step (see Figure 3), 82 papers were selected. We limited the publication venues to international journals and conferences. We applied the aforementioned exclusion criteria to the abstract and conclusion of each primary study. After this step, 229 papers were excluded. So we end up analyzing 30 primary studies.

C. Defining a Classification Scheme

We applied the classification schemes proposed by Petersen et al. [2] and classified the publications into categories from three perspectives: (*i*) **Focus Area**, (*ii*) **Contribution Type** and (*iii*) **Research Type**. The resulting classification schemes are described in the following subsections.

1) Focus Areas: After reading through the primary studies, five focus areas were identified. The first one is **"Software Modernization"**. This focus area is related to primary studies that describe approaches that employ ADM to fully modernize legacy systems either to another platform or architecture. The second focus area is related to **"Business Knowledge Extraction"**, which describes primaries studies on processes, methods, or approaches to extract business-related information of legacy systems. The third one is **"Concern Extraction"** and it comprises primaries studies report on processes, methods, or approaches to extract crosscutting concerns (CC) of legacy systems. The forth one, **"Extension of ADM's Metamodels"**, is concerned with grouping studies taht present approaches, methods, or processes to extend one of the ADM's metamodels. Finally, the last focus area is **"Applicability"** which includes papers that mainly focus on presenting evidence related to applying ADM and its metamodels in practice. In other words, papers that enable researchers and practitioners to get a better understanding and utilization of ADM and its metamodels.

2) Contribution Type: five contribution types were identified. **"Tools"** groups primary studies that focus on providing tools to support the modernization of legacy system by using ADM. The second contribution type is **"Process"**, which refers to primary studies that describe processes to assist the modernization of legacy system by means of ADM. **"Model Transformation"** contains primary studies that describe the use of language transformation such as Query/Views/Transformations (QVT)[1] or ATL Transformation Language (ATL)[2] to realize transformation among the ADM's metamodels. The fourth contribution type is **"Metamodel"**. Studies in this category create or extend the ADM's metamodels to deal with a specific problem as, for instance, providing a KDM light-weight extension in order to either represent the aspect oriented paradigm or supports a component-oriented decomposition. The last contribution type is **"Metrics"** and describes papers that focus on proposing or applying metrics to gauge the effectiveness of ADM and its metamodels.

3) Research Type: The research type reflects the research approach used in the primary study. The research type categories are based on the scheme proposed by Wieringa et al. [5] (RQ_3), as follows: Studies in the **"Validation Research"** aims to examine a solution proposal that has not yet been practically applied. It is conducted in a systematic way and may present any of these: prototypes, math analysis, etc. **"Evaluation Research"** in contrast to validation research, evaluation research aims at examining a solution that has already been practically applied. Studies in this category investigate the practical implementation of the proposed solution and usually present results using empirical strategies (e.g., experiments and study cases). **"Conceptual Proposal"** presents an arrangement to perceive things that already exist, in a novel way. **"Experience paper"** reports on personal experience of the author from one or more real life projects. **"Opinion papers"** report on the personal opinion of the author on suitability or unsuitability of a specific technique or tool.

D. Data Extraction and Synthesis

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 a summary of the primary study, highlighting how ADM and its metamodels were used, date when the data extraction took place, title, authors, venue, and a summary of the study's conclusion. During the extraction process, information about each primary study was independently gathered by all reviewers. The review was performed in November, 2013 by three Masters students, a PhD student, and three domain experts.

E. 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 [3].

Figure 5 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

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

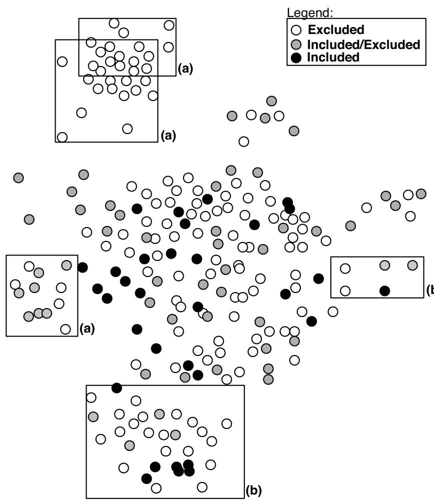

Figure 5. Document map colored with the history of the inclusions and exclusions of the studies.

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 5 just five clusters generated by PEx are depicted. Examples of pure clusters (all excluded) are identified in Figure 5 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.

F. Mapping and discussion of research questions

The focus of this section is to present a broad overview of the research on ADM. Apart from creating this overview, we also used the information drawn from the primary studies to answer the research questions.

Instead of using frequency tables we have decided to produce a bubble plot to report the frequencies and distribution of the selected studies according to their categories and publication date, thereby providing a map of research related to ADM. Our resulting map is shown in Figure 7. 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. Figure 7 has three facets: **Contribution Type**, **Focus Area** and **Research Type**. It is worth highlighting that certain primary studies were grouped in more than one category, affecting the frequency count; i.e., the sum of the frequencies shown in each facet can be greater than the total of selected studies presented earlier (30).

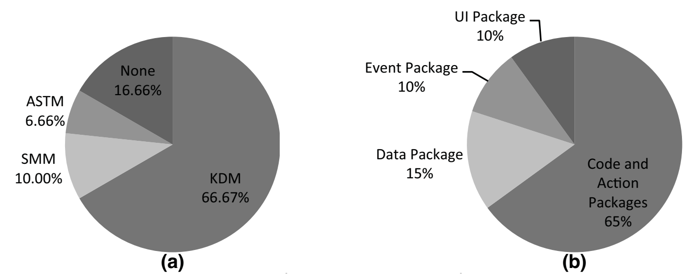

Figure 6. Frequency of ADM's metamodels used in the literature and the most and least used packages.

As for answering the first part of **RQ**$_1$ we analyzed all primary studies, focusing on investigating which ADM standard metamodels have more been used in the literature. In Figure 6(a) is depicted a pie chart wherein we plotted the collected data. As can be seen in this figure, KDM seems to be the most used metamodel (66%). A small percentage of primary studies have reported on the use of SMM (10%). While ASTM has been used by a rather small amount of studies (6.66%). We found that 16.66% of the primary studies does not explicitly mention which metamodel has been used during the modernization process. In order to answer the second part of **RQ**$_1$ we investigated which are the most and least used packages within the KDM. In Figure 6(b) it is fairly evident that the packages Code and Action are the most used in the literature (65%). We surmise that the reason for this is twofold: (*i*) these packages are often used to represent the source-code of system and since most of the primary studies use 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 databases. As shown in Figure 6(b), the least used packages are Event and UI.

By observing Figure 7 (right side) it is possible to answer **RQ**$_2$. The vast majority of the primary studies were classified as **Evaluation Research**, approximately 49%. A small percentage of publications is concerned with **Validation Research** (3.12%). While 30% of the selected studies fall into **Experience Paper** and **Opinion paper**. Finally, **Conceptual Proposals** account for 18% of the selected studies.

In Figure 7 (left side), it can be seen that the majority

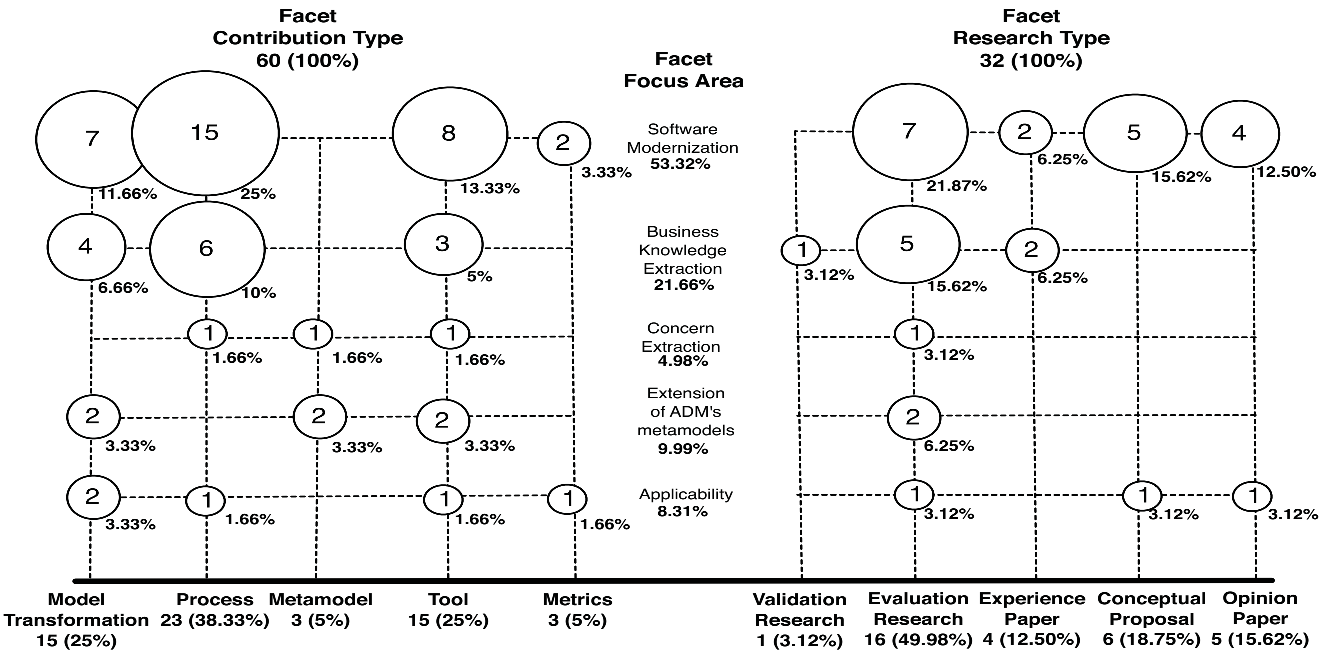

Figure 7. Overview of research on ADM and its metamodels.

of the research papers focus on processes to assist software engineers during the modernization of legacy system. Model transformations and Tools are fields that have also been researched. We believe that these fields have drawn a lot of attention from researchers because most primary studies that describe modernization processes also propose a set of model transformations and a tool that fully or partially automates the proposed process. On the other hand, the contribution type with less studies are **Metamodel** and **Metrics**. Thus, it is argued that primary studies that describe processes to assist the modernization of legacy systems by means of ADM, papers that show a set of rules to be applied during model transformation among the ADM's metamodels (KDM, SMM and ASTM), and papers that devise tools to assist ADM's process can be considered as evidence clusters. In other words, where there may be scope for more complete literature reviews to be undertaken. Whereas metamodels (i.e., papers that explain how to extend ADM's metamodels) and metrics (i.e., papers that describe how to apply metrics in ADM's metamodel) can be regarded as gaps, thus new or better primary studies are required.

In terms of focus area, Figure 7 (middle) shows that previous research has turned much attention to presenting **Software Modernization** (i.e., 53.32%). **Business Knowledge Extraction** has also been significantly covered, 21.66%. While **Concern Extraction**, **Extension of ADM's metamodels** and **Applicability** have been presented collectively by rather small percentage of 25%. As result of this analysis we partially answered $\mathbf{RQ}_3$. We highlighted the main types of contributions that have been proposed in the ADM literature so far. We organized the other part of the analysis related to $\mathbf{RQ}_3$, i.e., (a discussion about the focus area regarding the ADM) in subsections. Each subsection briefly describes the studies selected for each **Focus Area** while highlighting the extent and nature of research.

*1) **Software Modernization**:* Jorge Maratalla et al., propose **GAFEMO** [6], which aims to modernize a legacy systems to the 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 representations of it. Afterwards, a set of rules are applied in this model to create the services.

In [7] 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 [8] is defined an approach that is focused on the analysis of legacy systems to discover and create functionalities to be exposed as services using Web Services by means of ADM. 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 [9], [10] 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 [11] the authors propose an approach that uses ADM which is focused on the analysis of legacy systems to discover and create functionalities to be exposed as services using Web Services.

Pérez-Castillo et al., [12]–[14] present approaches to modernize legacy systems together with the legacy relational database. This approach recovers the code-to-data linkages and obtains three kinds of models according to the ADM approach: (*i*) The KDM Code Model, which represents the inventory of legacy source code. It has also the points that link the SQL Sentence Models and Database Schema Models. (*ii*) The SQL Sentence Model for modeling a certain SQL query that was embedded in legacy source code. (*iii*) The Database Schema Model, which represents the specific database fragment derived by an SQL Sentence Model. In [15] presents the XIRUP modernization methodology, which proposes a highly iterative process. This 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 Graphical User Interface (GUI) of legacy systems during the modernization, taking the opportunities offered by novel interaction paradigms, i.e., Rich Internet Application (RIA).

In [17] the authors present an approach for the definition of a systematic process for Web Applications (WA) to RIA modernization, by applying ADM principles. The approach presented by the authors consists on generating a RIA client from the legacy WA presentation and navigation layers and its corresponding service-oriented connection layer with the underlying business logic at server side. Boussaidi et al., [18] propose an approach that makes use of the KDM to reconstruct and document software architectural views of the legacy system. They consider an architectural view to be a way of partitioning a system using a specific set of KDM relevant concepts and relations and they propose clustering algorithms that target specific views mainly a layered view that we call horizontal view and a feature based view that we call vertical view. In [1] ADM is used into practice by building a modernization tool to generate metric reports of legacy Oracle Forms applications to assess migration efforts. The authors devised an extractor that generates KDM models from PL-SQL code (PL/SQL-to-KDM) and a metrics report generator for these KDM models.

2) ***Business Knowledge Extraction****:* Pérez-Castillo et al., [19]–[22] present an approach to recover business processes from legacy systems. This approach is based on a set of transformation: (*i*) transformation obtains PSM models from each legacy software artifact using a specific metamodel for each artifact; the traditional reverse engineering techniques such as static analysis, dynamic analysis, and formal concept analysis 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 [23] the authors report the results of a family of case studies that were performed to empirically validate this approach. Pérez-Castillo et al., also provides in [24] 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 [22] 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.

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

3) ***Concern Extracting****:* Santibáñez et al., [27] propose an approach called CCKDM for identifying crosscutting concerns by means a combination of a concern library and a K-means clustering algorithm. The input of the approach is a KDM model instance and the result is the same KDM model with annotated concerns. According to the authors this is the first work in concern mining area that use a standardized model in the context of ADM to perform search of concerns. They also believe that ADM standards will be widely used in a near future because is an OMG initiative.

4) ***Extension of ADM's metamodels****:* We identified three papers that address how to perform extension of ADM's metamodels. We provided a brief summary of these paper are follows.

In [28] 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 [24] propose an extension to the KDM that aims to represent all the information registered in a MXML model in the KDM model. In [29] the author proposes an extension of the KDM to represent all elements of the Aspect Oriented Paradigm, i.e., aspect, advice, point-cut, can be represented using the KDM. All these three paper have in common is that the authors claimed the impact of these extensions on well-proven and KDM based tools is not problematic since they are performed with the own extension mechanism of the KDM standard.

*5) **Applicability**:* Pérez-Castillo et al., [21], [24], [30] present how to apply KDM to modernize legacy systems. Also, the authors described each layer of the metamodel KDM, they also presented a set of example of how to use ADM and KDM during the modernization of a legacy systems. The authors claim that the paper enables researchers and practitioners to get a better understanding KDM.

III. Main Findings and Open Issues

Recent proposals in ADM have focused mainly on providing approaches to modernize legacy system to another platform/architecture. However, if we look at overall problem of the integration of modernization into an ADM context, there is still room for improvement. For instance, in order to integrate ADM's metamodels into larger context, the area of discovering knowledge, i.e., parsers needs more attention along with solution to verification of models. Few efforts (e.g., [1], [31]) have dealt with devising parsers to represent instances of KDM, but even these parsers provide limited infrastructure to represent all KDM's layers. Thus, new efforts must be conducted to create a more effective parsers able to represent all KDM's layer. Besides, the processes to discovery of knowledge are often mostly static in a sense that these parsers are unable to obtain knowledge during the executing of the target legacy system. Hence, further research is required to support the discovery of knowledge dynamically.

Another issue is that although KDM had been created to support modernization of legacy systems, the original version of the KDM does not contains metaclasses suitable for representing, for instance, Aspect Oriented Programming concepts; making it difficult to conduct a modernization process whose goal is to remodularize crosscutting concerns. In order to overcome the aforementioned limitation, in [32] we devised a heavyweight extension for KDM called KDM-OA. The goal is to create an extension that allows representing as high-level as low-level AO details, but still respecting the language and platform independence offered by KDM. As result it is possible to apply modernization based on Crosscutting Framework Families [33].

Also, we observed that there are three main hurdles that demand more research so that modernization techniques can be used in the ADM approach in an effective way. The first hurdle is the present lack of a fully developed idea of "good" KDM style. This is an important issue, for a clear notion of style is a fundamental prerequisite for the use of modernization, enabling software engineers to see where they are heading when modernizing their legacy system with KDM. Fowler et al. [34] 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. The second one – both a cause and a consequence of the first – is the lack of a KDM equivalent of such catalogues. We assume that the process of modernization by using KDM would equally benefit from KDM specific catalogues of smells and refactorings, helping software engineers to detect situations where the KDM could be improved, guiding them through the corresponding transformation processes. The third hurdle is the absence of a tool that supports refactoring by using KDM specification. The catalogue presented by Fowler et al. [34] provides a basis upon which developers can rely on to build tool support for object-oriented refactoring: a similar catalogue for the KDM specification is likely to bring similar benefits to assist software engineers during the modernization process.

IV. 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 because we selected studies without assigning any scores.

Missing important primary studies: We conducted the review in several search engines. Nevertheless, it is possible that we some primary studies were left out of our selection. We mitigated this threat by selecting search engines that have been regarded as the most relevant scientific sources [4].

Reviewers reliability: The reviewers of this study are researchers in the software reuse field. So, we are not aware of any bias we may have introduced during the analyses.

Data extraction: Another threat for this review refers to how data was 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 and technical reports. In the event of a disagreement between the two primary reviewers, a third reviewer acted as an arbitrator to ensure that full agreement was reached.

V. Concluding Remarks

Research in the area of ADM can lead to advances in modernization of software systems, resulting in software systems that are more maintainable, extensible, and reusable. To get an overview of the current research in this area we carried out a systematic mapping. After examining 30 primary studies, we answered three research questions.

We have found that the most used ADM's standard metamodel is KDM, which is used in approximately 66% of the primary studies. Also, we found that the most used packages are Code and Action, which was used in around 65% of the primary studies. We found that most papers in this area fall into the **Evaluation Research** category, 49.98% of the selected studies. A small percentage of publications is concerned with **Validation Research** (3.12%) Papers that fall into the **Experience** and **Opinion** categories account for 30% of all selected papers, and **Conceptual Proposals** account for 18%. Concerning the third research question, we found that the main contributions types are as follows: (1) Process, (2) Model Transformation, (3) Tolls, (4) Metamodels and (5) Metrics.

Another contribution of this paper is the map we created. By observing it is possible to ascertain the extent and form of literature related to ADM, thereby identifying which categories have been emphasized in past research, gaps, and possibilities for future research. Furthermore, it provides additional insight into the frequencies of publication over time.

We confined our analysis mainly to the extent of the evidence available, rather than the content. Thus, as a longer-term future work, we intend to carry out systematic reviews in order to pinpoint the state of evidence in the most prominent categories.

ACKNOWLEDGMENT

Rafael S. Durelli would like to thank the financial support provided by FAPESP, process number 2012/05168-4. We also thanks Vinicius Durelli for his proof-reading of this paper.

REFERENCES

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

[2] 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.

[3] V. Malheiros, E. Hohn, R. Pinho, M. Mendonca, and J. C. Maldonado, "A visual text mining approach for systematic reviews," in *First Inter. Symposium on Empirical Software Engineering and Measurement*, 2007.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] R. Pérez-Castillo, I. G.-R. De Guzmán, and M. Piattini, "Implementing business process recovery patterns through qvt transformations," in *Inter. Conf. on Theory and practice of model transformations*. Springer-Verlag, 2010, pp. 168–183.

[23] 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.

[24] 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.

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

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

[27] D. Santibáñez, R. S. Durelli, B. Marinho, and V. V. de Camargo, "A Combined Approach for Concern Identification in KDM models," in *Latin-American Workshop on Aspect-Oriented Software Development*, ser. LAWASP '7, 2013.

[28] 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.

[29] P. M. Shahshahani, "Extending the knowledge discovery metamodel to support aspect- oriented programming," 2011.

[30] 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*, vol. 33, no. 6, pp. 519 – 532, 2011.

[31] 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.

[32] B. M. Santos, R. R. Honda, V. V. Camargo, and R. Durelli, "KDM-AO: An Aspect-Oriented Extension of the Knowledge Discovery Metamodel ," in *28th Brazilian Symposium on Software Engineering (SBES),*, 2014.

[33] V. V. Camargo and P. C. Masiero, "An approach to design crosscutting framework families," in *Proceedings of the 2008 AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software*. ACM, 2008.

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