Entity-relationship Modeling Tools and DSLs: Is It Still Possible to Advance the State of the Art from Observations in Practice?

Jonnathan Lopes[a], Maicon Bernardino[b], Fábio Basso[c] and Elder Rodrigues[d]

Postgraduate Program in Software Engineering (PPGES),
Laboratory of Empirical Studies in Software Engineering (LESSE),
Federal University of Pampa (UNIPAMPA), Av. Tiarajú, 810, Ibirapuitã, CEP 97546-550, Alegrete, RS, Brazil

Keywords: Multivocal Literature Mapping, Domain Specific Language, Conceptual Data Model, Conceptual Model, Database Modeling.

Abstract: The variety of database system technologies that became available in recent years makes difficult the selection of tools for modeling entity-relationship (ER). The published mapping studies on this topic date back to 2000, thus outdated and limited to guide designers towards the recent innovations selection for the design and implementation of databases. In this sense, we contribute with an overview of the recent innovations through a systematic literature mapping complemented by research in the gray literature. This paper scopes ten (10) primary studies focused on Domain-Specific Languages (DSL) and identifies fifty-five (55) tools already applied in industry and academia for ER modeling at the conceptual, logical, and physical level. Hence, as a significant increment to existing mapping studies, this presents the state-of-the-art and practice for ER modeling, including its characterization and research gaps.

1 INTRODUCTION

In the software development process, data models can undergo different transformations. The three-level modeling approach to database (DB) design (ANSI, 1975) suggests separating data models into different degrees of abstraction, known as conceptual, logical, and physical models. The conceptual modeling of database systems requires the highest level of abstraction, and the most widespread technique for its development is the conceptual data modeling proposal conceived by Chen (1976). This proposal, influenced by the three-level approach, was so well accepted that it was considered a definitive reference for DB modeling (Cougo, 2013).

DB modeling is a relevant part of the software development lifecycle, which generally some Domain-Specific Language (DSL) supported (Voelter, 2009). According to Van Deursen et al. (2000), a DSL is an executable programming language or specification language that offers expressive power focused, through appropriate notations and abstractions, usually restricted to a domain-specific problem. Like other languages, a DSL must present a set of sentences well defined by syntax and semantics themselves. For Fowler (2010), a DSL is a computer programming language with limited expressiveness and focused on a particular domain. In addition to code-first approaches, the adoption of DSLs in software processes promotes a significant increase in productivity, quality, ease of use, and flexibility (Mernik et al., 2005; Vara et al., 2014). The great advantage of using DSLs is an abstraction, which takes a less complex level of knowledge needed for their development. In this way, domain experts can understand, validate and modify the code through model transformations and refinements (Van Deursen et al., 2000).

According to Mernik et al. (2005), we can classify DSLs into three dimensions: origin (internal or external), appearance (textual, graphical, tabular, or symbolic), and implementation (run mode). In general, the principal consideration for creating a DSL should be where it comes from, as each approach has specific advantages and disadvantages (Fowler, 2010). On the one hand, we can design an internal DSL around the grammatical rules of an existing language, which can be a General Purpose Language (GPL) or another DSL. On the other hand, an exter-

[a] https://orcid.org/0000-0001-9402-801X
[b] https://orcid.org/0000-0003-2776-8020
[c] https://orcid.org/0000-0003-4275-0638
[d] https://orcid.org/0000-0003-3431-2814

Lopes, J., Bernardino, M., Basso, F. and Rodrigues, E.
Entity-relationship Modeling Tools and DSLs: Is It Still Possible to Advance the State of the Art from Observations in Practice?.
DOI: 10.5220/0011044500003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 1*, pages 179-186
ISBN: 978-989-758-569-2; ISSN: 2184-4992

nal DSL is a language with syntax owned and depends on its infrastructure for lexical, syntactic, and semantic analysis, interpretation, compilation, optimization, and code generation (Mernik et al., 2005). As for execution, conceptually, this dimension ranges from the specification of graphic and textual notations of languages applied on specific domains to implemented solutions.

The objective of this study is to map existing DSLs to DB modeling at any of the three model levels. To meet this purpose, we carried out a Multivocal Literature Mapping (MLM) to find primary studies and tools that present approaches, methods, or techniques for the DB models creation.

We organized this study as follows. Section 2 presents the discussion of related works. The protocol used we detailed in Section 3. Its execution is described in Section 4, while the obtained results are analyzed and discussed in Section 5. Section 6 describes the threats to the study's validity. Section 7 points out possible gaps for future research and, finally, Section 8 presents the final remarks.

2 RELATED WORK

Before performing this study, we performed an exploratory ad hoc[1] search in the literature for studies that proposed research related to approaches to modeling entity-relationship (ER) in DB, with a focus on primary studies and mappings.

Gregersen and Jensen (1999) performed a survey and analyzed the design properties applied to specify temporal attributes in ER modeling approaches. The aiming study presents the new concepts used to facilitate temporal modeling in 19 different proposals. These proposals are based on models, namely: ER, EER, OO, and ERC+. As a result, the different mapping models for relational models are highlighted, used in the database implementation platform, which has algorithms that optimize the definition of attributes with temporal value. In addition, we evaluated ten (10) models against various design properties to verify their completeness. The authors conclude that while the models collectively have adequate design coverage, none satisfy all aspects of the evaluated modeling. It denotes that the temporal aspects modeling in data models is a motivating activity and a promising research area.

Siau (2012) conducted a literature review on relational and semantic approaches to ER modeling. He shows that most studies point out that semantic models are better than relational models and that, to a lesser degree, other studies do just the opposite. Early studies argue that, in general, semantic models are easier to use for end-users as they offer notations with a higher level of abstraction for domain modeling. More recent studies that oppose the first observation, on the other hand, commonly assert that the concepts of relationships between entities in relational models are more understandable in making their relationship identifiers explicit. In addition, it discusses some works that claim there are no significant differences between the two approaches studied, citing, for example, the author carried out a controlled experiment using the same SQL statement, i.e. the subjects did not show any difference significant when they performed their modeling using the different approaches.

We can mention three other relevant non-systematic mappings of ER approaches. The unstructured literature review presented in Ma and Yan (2010) searches contributions to uncertain information in conceptual modeling, encompassing complex objects and inaccurate data. In Lorenz et al. (2016), the authors present an unstructured mapping literature study comprehending industry and academic best practices in ER projects.

In Torres et al. (2017), the authors describe a mapping study to characterize nine (9) platforms that support Object-Relational Mapping Solutions (ORMS) that includes examples in UML and Barker notations. These are complementary studies and do not overlap with the research questions proposed in this research. Furthermore, unlike ours, the studies aforementioned are awkward to replicate due to the lack of a protocol.

3 MULTIVOCAL LITERATURE MAPPING

A Multivocal Literature Mapping (MLM) is a form of Systematic Literature Mapping (SLM) that includes gray literature. MLMs are helpful for researchers and practitioners as they provide comprehensive insights into the state-of-the-art and practice in a particular area. In this study, we conducted an MLM[2] using the SLM process defined by Petersen et al. (2008) and the guidelines proposed by Garousi, V. and Felderer, M. and Mäntylä (2019) to search the gray literature.

To this end, we defined the following Research Questions (RQs): **RQ1.** What is the state-of-the-

[1] Ad hoc search is a customized search, using a tailor-made method, and structured to meet specific needs without necessarily a high degree of scientific rigor.

[2] Data from the experimental package available in the Zenodo DOI repository: 10.5281/zenodo.5724135

art in developing DSLs for transforming ER models? *e.g.* modeling or manipulation, and transformation. **RQ1.1.** What are refinement methodologies, techniques, and proposals (automated design) based on data models presented in the primary studies? *e.g.* normalization. **RQ1.2.** What tools support the development of these DSLs? **RQ2** What evaluation methods are used in primary studies? **RQ2.1.** What are the positive and negative points observed in carrying out the studies? **RQ2.2.** What challenges do the primary studies point out? **RQ3.** What are the tools for conceptual modeling of databases? **RQ3.1.** What notations or languages do these tools use? **RQ3.2.** What modeling levels (Conceptual, Logical, Physical) do these tools support?

Some of the RQs (**RQ1**, ..., **RQ2.2**) were essentially answered through academic studies, thus seeking to understand the state-of-the-art, while others (**RQ3**, ..., **RQ3.2**) reflected the investigation of the state-of-the-practice and therefore, we used gray literature for a better conclusion.

For the SLM, it was necessary to access Digital Libraries (DLs) to obtain scientific studies. For the research conducted, the DLs consulted should include three mandatory requirements: have a web-based search engine; have a keyword search engine, and; cover primary studies in the field of Computer Science. Therefore, we carried out the search in five DLs: (i) ACM DL; (ii) IEEE Xplore; (iii) ScienceDirect; (iv) Scopus; (v) SpringerLink. Afterward, we established the most expressive terms and synonyms for the research object. The combination generated the generic string in Figure 1.

(DSL **OR** Domain Specific Language **OR** Domain-Specific Language **OR** Domain-Specific-Language **OR** DSML **OR** Domain Specific Modeling Language **OR** Domain-Specific Modeling Language **OR** Domain-Specific-Modeling-Language **OR** Query Language) **AND** (ER **OR** Entity-Relationship **OR** Enhanced Entity-Relationship **OR** Extended Entity-Relationship **OR** Database)

Figure 1: Search string.

We elaborated selection criteria for primary studies to include the most relevant works. We divided the criteria into:

(i) *Inclusion Criteria (IC)*: (**IC1.**) The study proposes some technique, method, approach or tool for the representation and transformation of DB models using DSL. (ii) *Exclusion Criteria (EC)*: (**EC1.**) Study with less than four pages; (**EC2.**) Study not written in English; (**EC3.**) Duplicate study; (**EC4.**) Study not provide full access to its content; (**EC5.**) Study not meet IC1.

In addition, we established a set of seven (7) quality assessment criteria (QCs). The purpose of the quality assessment was both to qualify the relevance of the studies and assist in the interpretation of the results, as well as to limit the SLM bias (Dybå and Dingsøyr, 2008). In addition, we gave a score to the QCs: (i) **T:Total**, fully considering the evaluated QC; (ii) **P:Partial**, partially considering the evaluated QC and according to its weight; (iii) **N:Negative**, does not include the evaluated QC at all.

The maximum possible score, evaluating all criteria, was ten (10.0) and the minimum zero (0). Each QC had a specific weight (1 → 1.5 → 2) depending on its importance considered for this study. We considered aspects relevant to the research these QCs based on, namely **report** (QC1, QC4, QC5), **rigor** (QC2, QC3), **credibility** (QC2, QC3) and **relevance** (QC1, QC6, QC7). Table 1 presents the list of QCs and their respective weights.

Table 1: Quality Assessment Criteria

QC	Description	Weight
QC1	Does the study make any contribution to the field of DB modeling?	1.5
QC2	Does the study present techniques or proposals for refinement based on data models ?	1.5
QC3	Does the study apply any form of empirical evaluation?	1.5
QC4	Does the study describe the characteristics of the DSL creation process?	1.5
QC5	Does the study characterize the model transformation activities for different DB technologies?	2.0
CQ6	Does the study have positive and negative points observed in its execution?	1.0
QC7	Does the study point out challenges arising from its implementation?	1.0

Once we carry out the quality assessment, it is necessary to perform the data extraction step to interpret the results. Thus, a data extraction form was created, consisting of a series of items, such as the origin of the solution, the year of publication, the objective, the evaluation of the proposal, the positive and negative points observed, among other data.

We used the Google search engine to seek database modeling tools in gray literature. For this, we defined combinations of keywords from the search string and adapted them to the context of the search on the web: (i) ERD Modelling Tool; (ii) ERD Design Tool; (iii) Conceptual Design of Database; (iv) Conceptual Modelling of Database; (v) Conceptual Modelling; (vi) Database Modelling Tool; (vii) Database Design Tool. The verification of results was limited to 10 (ten) pages in the search engine for each keyword.

The acceptance of tools into MLM was subject to a selection process based on certain requirements. First, the tool should provide access for some form of use (*e.g.* trial version), it would need to support some level of DB modeling and it would need to have interface in English or Portuguese (our native language). Afterwards, the included tools should have the notations and modeling levels extracted, along with other relevant information, and then categorized.

4 MLM EXECUTION

For the execution of the SLM, it was necessary to adapt the syntax of the generic *string* to generate other versions, seeking to adapt the peculiarities of parameterization of the different databases used. Then, we performed a search for studies in the databases. We obtained the number of primary studies returned by the digital library as follows: (i) IEEE Xplore: 513; (ii) Scopus: 465; (iii) ACM: 1240; (iv) Springer Link: 683; (v) ScienceDirect: 826.

With the initial set of 3727 primary studies identified, we defined five (5) selection cycles. In these iterations, duplicate studies were excluded (with 3513 remainings), selection of studies based on title and abstract (with 34 remainings), selection based on full text (with 18 remainings), and selection based on quality assessment (with 10 remainings). Each iteration had the objective of eliminating studies that were outside the scope of the research or considered not relevant. In the last iteration, 18 studies had their quality analyzed. We established that only studies with scores above five (5) would be accepted.

Thus, after applying the QCs, eight (8) studies were excluded. The final set of ten (10) studies approved by the SLM proceeded to the data extraction stage. With academic works, the lowest score was 5.3, while the highest reached ten (10). Table 2 summarizes the results obtained in the quality assessment.

Table 2: Quality Assessment Results.

Primary Studies	**Quality Criteria**							
Reference	**CQ1**	**CQ2**	**CQ3**	**CQ4**	**CQ5**	**CQ6**	**CQ7**	**Total**
Ayadi et al. (2016)	P	P	T	P	N	T	P	5.7
Celikovic et al. (2014)	T	T	N	T	P	P	T	7.0
Dimitrieski et al. (2015)	T	T	T	T	T	T	T	10.0
Hammer and Mc Leod (1981)	T	T	N	T	N	P	P	5.5
Jagannathan et al. (1988)	T	T	N	T	P	N	N	5.5
Kersten et al. (2011)	P	T	T	P	N	N	P	5.3
Litwin et al. (1989)	P	T	N	T	N	P	T	5.4
Mazairac and Beetz (2013)	T	T	P	T	N	N	P	5.9
Shipman (1981)	T	T	N	T	P	N	P	6.0
Tian et al. (2006)	T	P	N	T	P	P	T	6.4

Caption: T = *Total*, P = *Partial*, N = *Negative*

The execution of the search protocol in the gray literature returned a total of 132 tools. After summarizing, duplicates were excluded (with 67 remainings). During this process, we did not evaluate twelve (12) tools due to some unavailability type, *e.g.* the impossibility of installation or incompatibility with the environment used. In the end, it was possible to run usage tests on 55 tools, and consequently, extract relevant data for the study. During the use of the tools, we created simple database models, in which we tried to observe the support for modeling levels and the notations and languages used.

5 MLM RESULTS

Regarding the state-of-the-art (**RQ1**) in the development of DSLs applied to database modeling, a study was identified that presents the Multi-paradigm Information System Modeling Tool (MIST) (Dimitrieski et al., 2015) , which uses a bidirectional DSL called EERDSL, for conceptual modeling using the Extended Entity-Relationship (EER) approach. Bidirectional modeling and transformation is a model-driven engineering paradigm concept. In this context, we are referring to a DSL capable of transforming models to code, and from code to models. As for the refinement methodologies, techniques, or proposals (**RQ1.1**) based on data models, only the two studies mentioned applying concepts for refinement, supported by database normalization to help developers use their solutions.

With regard to the technologies used to support DSL development (**RQ1.2**), we registered the following: Xtext, Xtend, Sirius and Eugenia (Celikovic et al., 2014; Dimitrieski et al., 2015), StarUML (Ayadi et al., 2016), IfcDoc Tool and ViewEdit Tool (Mazairac and Beetz, 2013), MonetDB (Kersten et al., 2011), Java, JFlex, and JCup (Tian et al., 2006). However, older studies were generally DSL specifications, not showing any form of implementation or tool used (Jagannathan et al., 1988; Litwin et al., 1989; Shipman, 1981).

The database representations adopted (**RQ1.3**) have `Tables` and `Functions` in all primary studies analyzed. There are also explicit references to the definition of `Stored Procedures`, `Triggers`, and `Views` in other studies. Table 3 summarizes the data retrieved from each of the primary studies.

Table 3: Represented Database Objects.

Primary Studies		**DB Objects**				
Reference	**DSL**	**TB**	**SP**	**F**	**T**	**V**
Ayadi et al. (2016)	Ayadi's Notation	✓				
Celikovic et al. (2014)	EERDSL v.1	✓		✓	✓	
Dimitrieski et al. (2015)	ERRDSL v.2	✓	✓	✓	✓	✓
Hammer and Mc Leod (1981)	SDM	✓		✓		
Jagannathan et al. (1988)	SDM	✓		✓		
Kersten et al. (2011)	SciSQL	✓		✓		
Litwin et al. (1989)	MSQL	✓	✓	✓	✓	✓
Mazairac and Beetz (2013)	BIMQL	✓	✓	✓		
Shipman (1981)	DAPLEX	✓		✓		
Tian et al. (2006)	NeuroQL	✓		✓		

Caption: TB = *Tables*, SP = *Stored Procedures*, F = *Funtions*, T = *Triggers*, V = *Views*

On the methods used to evaluate DSL (**RQ2**), there is only one preliminary study that presents the validation of the proposal (Dimitrieski et al., 2015) using 16 participants, two (2) experts in IHC, three

(3) experts in systems modeling and 11 students (6 masters in the database area and five (5) doctoral students with experience in modeling). In general, the other studies indicate the lack of an evaluation of their propositions as possible future work.

Among the positive and negative aspects observed (**RQ2.1**), the positive aspects of being easy to understand and with intuitive modeling and platform independence (Mazairac and Beetz, 2013; Tian et al., 2006) stand out. The negatives were the lack of automatic SQL code generation (Ayadi et al., 2016; Dimitrieski et al., 2015) and the lack of actual implementation of DSLs, with only specifications among the works (Ayadi et al., 2016; Hammer and Mc Leod, 1981; Jagannathan et al., 1988; Kersten et al., 2011; Tian et al., 2006).

In general, the main challenges identified by the studies (**RQ2.2**) are the evaluations of the approaches, as well as the evolution and/or simplification of the proposals. Table 4 presents the DSLs concerning their type. However, it is worth noting that the studies that mark the column `bidirectional` (Celikovic et al., 2014; Dimitrieski et al., 2015) are different versions of the same DSL implementation, while (Hammer and Mc Leod, 1981; Jagannathan et al., 1988) are a specification and implementation of DSL based on this specification, respectively.

Table 4: Categorization of DSLs Proposals.

Primary Studies		DSL Type		
Reference	**DSL**	**T**	**G**	**B**
Ayadi et al. (2016)	Ayadi's Notation		✓	
Celikovic et al. (2014)	EERDSL v.1			✓
Dimitrieski et al. (2015)	EERDSL v.2			✓
Hammer and Mc Leod (1981)	SDM	✓		
Jagannathan et al. (1988)	SDM	✓		
Kersten et al. (2011)	SciQL	✓		
Litwin et al. (1989)	MSQL	✓		
Mazairac and Beetz (2013)	BIMQL	✓		
Shipman (1981)	DAPLEX	✓		
Tian et al. (2006)	NeuroQL	✓		

Caption: T = *Textual*, G = *Graphical*, B = *Bidirectional*.

As for the state of practice (**RQ3**) of the tools used in the modeling of DBs, we mapped 55 tools. Thus, there was the classification as to its type, being 29 exclusive modeling DBs (*Data Modeling*), 13 of modeling with connection to integrated DBs and execution of queries (*Full IDE*), 10 with support for diagramming of different types of models (*Diagramming*) and three (3) tools designed for large companies (*Enterprise Modeling*).

Regarding the notations used in the tools (**RQ3.1**), we identified 18 notations, where the Crow's Foot notation highlights 35 occurrences and the IDEF1X notation with 23 records. Table 5 listed these and other data. And finally, concerning the models supported by the tools (**RQ3.2**) it was found that, individually, 26 tools support conceptual modeling, 48 tools support logical modeling and 37 support physical modeling. Figure 2 presents the set representing the intersections of the model support and the tools is presented in the Venn diagram. The numbers inside the diagram represent the tool IDs, also displayed in Table 5.

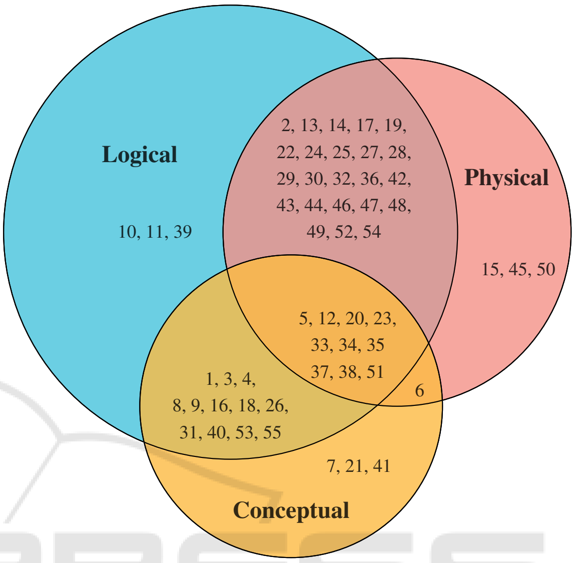

Figure 2: Venn diagram of the models supported in the tools.

6 THREATS TO VALIDITY

We identified threats to the study result in the MLM performed and then categorized it into the following types: construct validity, internal validity, external validity, and conclusion validity, according to the guidelines of Cook and Campbell (1979); Wohlin et al. (2012).

Construct Validity: Addresses the possibility that the RQs or search terms that structure the search string are inappropriate or incomplete. To mitigate these threats, we consulted DSL and researchers of data modeling. In addition, we perform a pilot survey to assess the consistency of our search string. Another threat is the quality of published material that has been collected in gray literature.

Internal Validity: Some possible threats are the implementation of incorrect search methods, which can lead to the exclusion of relevant studies, an application of poor data extraction strategy, the occurrence of biases in the selection or content of primary studies. In an attempt to mitigate these risks, we defined a protocol based on well-established reference models

Table 5: Database Modeling Tools.

		Type				Models			Supported Notations								Plataform		License	
ID	**Ferramenta**	**DM**	**FIDE**	**DG**	**EM**	**C**	**L**	**F**	**CF**	**IDEF1x**	**CN**	**MN**	**UML**	**BN**	**AN**	**ON**	**D**	**W**	**C**	**FC**
1	AnalyseSI	✓				✓	✓			✓		✓					✓			✓
2	Aqua Data Studio ER Modeler		✓				✓	✓	✓	✓							✓		✓	
3	Astah			✓		✓	✓		✓	✓							✓		✓	
4	brModelo			✓		✓	✓			✓	✓					✓			✓	
5	Creately			✓		✓	✓		✓									✓	✓	
6	Database Deployment Manager		✓			✓	✓	✓			✓						✓			✓
7	Database Workbench		✓			✓		✓	✓								✓		✓	
8	DB Designer	✓				✓										✓	✓		✓	
9	DB-Main	✓				✓	✓						✓				✓			✓
10	DBDesigner 4	✓				✓	✓		✓	✓		✓					✓			✓
11	DBDesigner.net	✓					✓									✓		✓	✓	
12	dbdiagram.io	✓					✓									✓		✓		✓
13	dbDiffo	✓				✓	✓	✓	✓						✓			✓		✓
14	dbForge Studio for MySQL		✓				✓	✓	✓	✓							✓		✓	
15	DBSchema		✓				✓	✓	✓	✓				✓			✓		✓	
16	DBVisualizer		✓					✓								✓	✓		✓	
17	DbWrench	✓				✓	✓		✓								✓		✓	
18	DeZign for Databases	✓					✓	✓	✓	✓							✓		✓	
19	Dia			✓		✓	✓		✓	✓	✓		✓	✓	✓		✓			✓
20	dModelAid	✓					✓	✓										✓	✓	
21	Enterprise Architect				✓	✓	✓	✓	✓	✓							✓		✓	
22	ER-Assistant	✓				✓			✓								✓			✓
23	ER/Builder	✓					✓	✓								✓	✓			✓
24	ER/Studio Data Architect	✓				✓	✓	✓	✓	✓							✓		✓	
25	ERD Concepts		✓				✓	✓	✓	✓							✓		✓	
26	ERDesigner NG	✓					✓	✓	✓								✓			✓
27	ERDPlus	✓				✓	✓		✓									✓		✓
28	Erwin Data Modeler	✓					✓	✓	✓	✓							✓		✓	
29	GenMyModel RDS			✓			✓	✓	✓									✓	✓	
30	InfoSphere Data Architect				✓		✓	✓	✓	✓							✓		✓	
31	Jeddict			✓			✓	✓	✓								✓			✓
32	ModelRight	✓				✓	✓			✓							✓		✓	
33	MySQL Workbench		✓				✓	✓	✓	✓			✓			✓	✓			✓
34	N Data Modeler	✓				✓	✓	✓	✓	✓							✓		✓	
35	Navicat Data Modeler	✓				✓	✓	✓	✓	✓			✓				✓		✓	
36	Open ModelSphere	✓				✓	✓	✓	✓		✓	✓					✓			✓
37	Oracle SQL Developer Data Modeler		✓				✓	✓	✓	✓							✓			✓
38	pgModeler	✓				✓	✓	✓								✓	✓			✓
39	PowerDesigner				✓	✓	✓	✓	✓	✓	✓	✓		✓			✓		✓	
40	QuickDBD	✓					✓									✓		✓	✓	
41	RISE			✓		✓	✓		✓								✓			✓
42	Software Ideas Modeler			✓		✓			✓	✓	✓						✓		✓	
43	SQL Database Modeler	✓					✓	✓		✓								✓	✓	
44	SQL Maestro		✓				✓	✓		✓							✓		✓	
45	SQL Power Architect	✓					✓	✓	✓								✓		✓	
46	SQL Server Management Studio		✓					✓								✓	✓			✓
47	SQLDBM	✓					✓	✓		✓								✓		✓
48	SQLyog		✓				✓	✓								✓	✓		✓	
49	Toad Data Modeler	✓					✓	✓	✓	✓							✓		✓	
50	Valentina Studio		✓				✓	✓	✓								✓		✓	
51	Vertabelo	✓						✓	✓									✓	✓	
52	Visual Paradigm			✓		✓	✓	✓	✓								✓		✓	
53	Win A&D			✓			✓	✓	✓								✓		✓	
54	WWW SQL Designer	✓				✓	✓									✓		✓		✓
55	xCase	✓					✓	✓	✓								✓		✓	

Caption: DM (*Data Modeling*), FIDE (*Full IDE*), DG (*Diagramming*), EM (*Enterprise Modeling*) | C (Conceptual), L (Logical) F (Physical) | CF (*Crow's Foot*), CN (*Chen's Notation*), MN (*Merise Notation*), BN (*Barker's Notation*), AN (*Arrow Notation*), ON (*Other Notation*) | D (*Desktop*), W (Web), C (Comercial), FC (*Free of Charge*) in the literature.

External Validity: External threats often address whether findings from one study can be generalized to another domain. One reason for this threat would be the occurrence of primary studies selection containing incomplete information. However, it is likely that, as this is an area of intersection between data modeling and DSLs, the results cannot be generalized to other research topics, thus reducing this risk naturally.

Conclusion Validity: One possible threat is bias in data extraction, which leads to completion errors. To alleviate this problem, we performed a careful reading and tests of the use of the tools, and there was the synthesis of data in electronic spreadsheets for a better analysis.

7 RESEARCH PERSPECTIVES

This section presents the main research perspectives on still open questions that researchers can explore in the future.

Lack of Previous Research on the Topic: We easily found in the literature different secondary studies of domains in data modeling or DSLs. Notwithstanding, before the execution of this MLM, we carried out an ad hoc search for other research initiatives similar to ours, but no results with exactly the same objectives were obtained.

Absence or Limitation in the Evaluation of Study Proposals: There is a consensus that software technologies need to be investigated and empirically evaluated and that primary studies should generally report how their proposal is validated. However, the lack of evaluation descriptions in the studies seen in this MLM was frequent during the analysis studies. It demonstrates a weakness in the methodology implemented to carry out the studies that highlight the great concern with development. It is also possible to understand that activities inherent to Software Engineering that could add value to the quality of the study proposal are also not discussed, including domain analysis for the creation of DSLs or their maintenance.

Automatic SQL Code Generation in Academic Studies: The lack of emphasis on approaches to transform data models is notable as opposed to tools applied in the industry, although at least two studies deal with automatic SQL code generation for databases. Although most of the identified DSLs have good specifications, we did not find initiatives to model databases using textual DSLs. This fact may indicate the odds this potential field to become a relevant research area.

8 FINAL REMARKS

Every year, several contributions to modeling (ER) are published. Database modeling is a fundamental area in Software Engineering, and DSLs that support this activity are did not trivially find in the literature. Alternatives research for database modeling is necessary to follow the evolution and trends of various Database Management Systems (DBMSs). This paper aims to provide an overview of the DSLs used by ER modeling and the tools used in the industry through an MLM. The principal motivation for this research is that we believe that the usability of DSLs, with their high power of domain expertise, is an element that can sharp help in modeling activities in DBMSs, especially at a conceptual level.

This study covered 3513 distinct studies with the intention of investigating primary studies that proposed DSLs for modeling DBs. Likewise, we raised a set of 67 different tools that support database modeling, seeking to map their notations and supported models. We detailed the MLM protocol, its conduction and subsequent, analysis of the results obtained. In the end, we selected ten (10) primary studies and 55 tools to be analyzed quantitatively and qualitatively. As a result, only DSLs currently used to support ER modeling and tools used in database design were rated. Among the results, the study by Dimitrieski et al. (2015) stands out, which presents a bidirectional modeling tool that applies its DSL based on the EER approach.

Finally, our research provides shreds of evidence that each year a significant of papers presenting different notations types are published, as seen in the number of publications analyzed from the result returned from the base string. It is somewhat surprising, given that the ER notations used today by industry and academia, such as those by Peter Chen and Barker, are not recent proposals. Therefore, we concluded that ER modeling remains a wide field of research with a few gaps, *e.g.* the loss of tools that support the design in many lifecycles through automated design techniques.

REFERENCES

ANSI, A. N. S. I. (1975). Interim Report: ANSI/X3/SPARC Study Group on DBMSs 75-02-08. *ACM SIG on Management of Data*.

Ayadi, M. G., Bouslimi, R., and Akaichi, J. (2016). A framework for medical and health care databases and data warehouses conceptual modeling support. *Network Modeling Analysis in Health Informatics and Bioinformatics*.

Celikovic, M., Dimitrieski, V., Aleksic, S., Ristic, S., and Lukovic, I. (2014). A DSL for EER Data Model Specification. In *23rd Int. Conf. on Information Systems Development*, pages 290–297.

Chen, P. P.-S. (1976). The Entity-relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1):9–36.

Cook, T. and Campbell, D. (1979). *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. Houghton Mifflin.

Cougo, P. (2013). *Modelagem conceitual e projeto de banco de dados*. Elsevier, Rio de Janeiro, RJ, Brasil.

Dimitrieski, V., Čeliković, M., Aleksić, S., Ristić, S., Alargt, A., and Luković, I. (2015). Concepts and Evaluation of the Extended Entity-relationship Approach to Database Design in a Multi-paradigm Information System Modeling Tool. *Computer Languages, Systems & Structures*, 44(Part C):299–318.

Dybå, T. and Dingsøyr, T. (2008). Strength of evidence in systematic reviews in software engineering. In *2nd ACM-IEEE Int. Symp. on Empirical Software Engineering and Measurement*, pages 178–187. ACM.

Fowler, M. (2010). *Domain Specific Languages*. Addison-Wesley, 1st edition.

Garousi, V. and Felderer, M. and Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information & Software Technology*, 106:101–121.

Gregersen, H. and Jensen, C. S. (1999). Temporal entity-relationship models-a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(3):464–497.

Hammer, M. and Mc Leod, D. (1981). Database Description with SDM: A Semantic Database Model. *ACM Transactions on Database Systems*, 6(3):351–386.

Jagannathan, D., Fritchman, B. L., Guck, R. L., Thompson, J. P., and Tolbert, D. M. (1988). SIM: A Database System Based on the Semantic Data Model. *ACM SIGMOD Record*, 17(3):46–55.

Kersten, M., Zhang, Y., Ivanova, M., and Nes, N. (2011). SciQL, a Query Language for Science Applications. In *Work. on Array Databases*, pages 1–12, New York, NY, USA. ACM.

Litwin, W., Abdellatif, A., Zeroual, A., Nicolas, B., and Vigier, P. (1989). MSQL: A Multidatabase Language. *Information Sciences*, 49:59–101.

Lorenz, M., Hesse, G., and Rudolph, J.-P. (2016). Object-relational Mapping Revised - A Guideline Review and Consolidation. In *11th International Joint Conference on Software Technologies*, pages 157–168.

Ma, Z. and Yan, L. (2010). A Literature Overview of Fuzzy Conceptual Data Modeling. *Information Science and Engineering*, 26:427–441.

Mazairac, W. and Beetz, J. (2013). BIMQL - an open query language for building information models. *Advanced Engineering Informatics*, 27(4):444–456.

Mernik, M., Heering, J., and Sloane, A. M. (2005). When and How to Develop Domain-specific Languages. *ACM Computer Survey*, 37(4):316–344.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. In *12th Int. Conf. on Evaluation and Assessment in Software Engineering*, pages 68–77. BCS Learning & Dev. Ltd.

Shipman, D. W. (1981). The Functional Data Model and the Data Languages DAPLEX. *ACM Transactions Database Systems*, 6(1):140–173.

Siau, K. (2012). *Innovations in Database Design, Web Applications, and Information Systems Management*. Information Science Reference.

Tian, H., Sunderraman, R., Calin-Jageman, R., Yang, H., Zhu, Y., and Katz, P. (2006). NeuroQL: A Domain-specific Query Language for Neuroscience Data. In *Int. Conf. on Current Trends in Database Technology*, pages 613–624, Berlin, Heidelberg. Springer.

Torres, A., Galante, R., Pimenta, M. S., and Martins, A. J. B. (2017). Twenty years of object-relational mapping: A survey on patterns, solutions, and their implications on application design. *Information and Software Technology*.

Van Deursen, A., Klint, P., and Visser, J. (2000). Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, 35(6):26–36.

Vara, J., Bollati, V., Jiménez, A., and Marcos, E. (2014). Dealing with Traceability in the MDD of Model Transformations. *Trans. on Soft. Eng.*, 40(6):555–583.

Voelter, M. (2009). Best Practices for DSLs and Model-Driven Development. *Journal of Object Technology*, 8(6):79–102.

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., and Wessln, A. (2012). *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated.