Protocol for a Systematic Literature Review on Design Decisions for UML-based DSMLs

Sobernig, Stefan; Hoisl, Bernhard; Strembeck, Mark

DOI:
10.57938/0e8da10b-9dda-43f0-9a53-3a9ee073b421

Published: *01/10/2014*

Document Version:
Publisher's PDF, also known as Version of record

Protocol for a Systematic Literature Review on Design Decisions for UML-based DSMLs*

Stefan Sobernig[1], Bernhard Hoisl[1,2], and Mark Strembeck[1,2]

[1] Institute for Information Systems and New Media,
Vienna University of Economics and Business (WU Vienna)
[2] Secure Business Austria Research (SBA Research)
{firstname.lastname}@wu.ac.at

Document History

Version 1	Initial document, top-level structure adapted from Kitchenham et al. [65]. The procedural steps as proposed by Kitchenham [64] are applied.	Stefan Sobernig, Oct 2012
Version 2	Initial data extraction scheme and research questions; comments on research questions, search strategy and process, inclusion/exclusion criteria, quality assessment.	Bernhard Hoisl, Oct 2012
Version 3	Protocol revision, the procedure has been extended to include steps described by guidelines on the quasi-gold standard by Zhang et al. [129] and backward snowballing (see, e.g., [60]).	Stefan Sobernig, Jan 2013
Version 4	Revised inclusion/exclusion criteria.	Bernhard Hoisl, Jan 2013
Version 5	Revised inclusion/exclusion criteria.	Bernhard Hoisl, Feb 2013
Version 6	Protocol revision, adding dedicated sections about intermediate results and deviations from the protocol, for each step.	Stefan Sobernig, Jul 2013
Version 7	Adding Section 8 on applying a *frequent item-set analysis* [16] on the extracted decision-option sets.	Stefan Sobernig, Jul 2014
Version 8	Revising document, proof-reading.	Mark Strembeck, Aug 2014
Version 9	Revising document, cleaning up, proof-reading and fixing some typos.	Stefan Sobernig, Oct 2014
Version 10	Adding background section on prior research stages leading to this SLR.	Stefan Sobernig, Jan 2015
Version 11	Revising document, cleaning up, proof-reading.	Bernhard Hoisl, Feb 2015

On this Document

This document is the protocol on planning and reporting the conduct of a systematic literature review (SLR) to identify and to extract design decisions from existing and documented domain-specific modeling languages (DSMLs) implemented using the Unified Modeling Language (UML [78]). It is provided as supplemental material for two other documents. On the one hand, there is a manuscript submitted for publication which presents and discusses the results of this SLR. On the other hand, there is the decision-record catalog [54] which was extended and validated using this SLR study. Therefore, the protocol is supposed to be read in conjunction with those two documents, to trace all details of performing and eventually reproducing the steps of the SLR. Without the context of the two companion documents (manuscript, catalog), it may be difficult to comprehend the protocol as a standalone artifact.

Format and Notation. In this protocol, two different currents of text interleave. On the one hand, the procedural steps planned to perform the SLR are documented in the main text stream (sections, paragraphs). On the other hand, deviations from the planned procedure as observed when performing the SLR, as well as the intermediate results (selection decisions, data processing) obtained from individual steps are documented in greyed text blocks (including auxiliary data tables and figures). These reports on deviations, as well as on intermediate and raw results follow the corresponding main-text sections directly.

In Sections 2 and 4, word groups are decorated (e.g., underline, strikethrough) to indicate whether they were considered for formulating the search string used in the automated search of the SLR or not. These text decorations do *not* mark change information between different revisions of this document.

*This work has partly been funded by the Austrian Research Promotion Agency (FFG) of the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) through the Competence Centers for Excellent Technologies (COMET K1) initiative and the FIT-IT program.

0.1 Background: Early, Preparatory Research Stages

Design Reviews. As a pre-study, we initially conducted ten design reviews of UML-based DSMLs which we had developed ourselves. These DSMLs had emerged mostly from a long-running research effort to construct a modeling and runtime environment for business processes and process-related security properties (such as access control, confidentiality, or integrity). Based on a formal, UML-independent metamodel for process-related security properties, we defined a domain-specific UML extension called *BusinessActivities* for UML activity diagrams (see, e.g., [113, 98, 97, 96, 57, 55]). Moreover, we implemented a library and runtime engine that can manage Business Activity runtime models and enforce the different policies and constraints in a software system.[1] Based on the ten DSML designs, which were also documented in corresponding scientific publications, we drafted decision records for two decision points (D1, D2) comprising eleven decision options at that time. These first results (i.e., the decision records and option coding for the ten DSMLs) are reported in full detail in Hoisl et al. [52]. In a next step, we drafted three more decision records to cover three additional decision points in DSML design (D3, D4, D6) for the ten DSMLs. This extension of the original decision points was reported in an initial version of our standalone catalog [53].

Furthermore, we described eleven decision associations. A decision association exists if a decision option chosen at one decision point may influence other decision options at the same or at a subsequent decision point (see also Section 1). For example, a choice can favor, pre-determine, or exclude subsequent options. For the ten DSML projects, we described eleven decision associations within a single decision point or between two decision points [53]. Each association was represented by an option set containing two related decision options. One of the associations stated was $\{2.2, 4.6\}$, indicating that applying UML profiles (O2.2) implies diagram symbol reuse (O4.6), to give one example.

Backward Snowballing. For our pre-study, we then performed a backward-snowballing search (see [60]) based on the reference lists collected from the twelve (originally: ten; see above) scientific publications which documented our ten DSMLs at this point. This way, we selected six additional, third-party DSML designs for review. These additional UML-based DSMLs served as the basis for a revision of our decision-record catalog. The additional six DSML designs also helped us to test the applicability of our decision-record format and to evaluate the resulting decision descriptions beyond our own project experience.

For the six different points of decision making in DSML development [115], we gathered a total of 31 different design options. The decision options sum up to a total of 158 design choices detected in the 16 DSMLs. Furthermore, we identified 27 drivers for the different options of the corresponding decision points [53].

Drafting the Review Protocol and Pilot. After the pre-study, our work on decision records was still limited in important ways. First, the catalog only reflected a small number (16) of different DSML designs. Second, after the design reviews and backward snowballing on our own DSMLs, there was the clear risk of a bias towards the authors' experience and certain security-related technical domains of UML-based DSMLs. The ultimate objective, however, was to reduce any authors' bias and to extend the reach to DSML designs in diverse application domains.

Assuming scientific publications to be the most frequent and the most accessible primary sources of DSML design documentation, we chose the instrument of an SLR. Therefore, we strived for identifying existing review protocols on SLR research designs which could be reproduced or partly reused in a derived design. At the time, we could only spot two review-driven designs: a systematic mapping study on DSLs in the broadest sense by Nascimento et al. [75] and an earlier SLR on UML profiles by Pardillo [83]. However, while the former was not specific to UML-based DSMLs, the latter only reflected on a single UML extension technique (profiles). Hence, they did not qualify for adoption or reuse in designing our SLR.

Thus, we started setting up a corresponding SLR protocol. The initial review design is documented in full detail in revision 3 of [106]. This first SLR design closely followed the review procedure and guidelines that were proposed by Kitchenham et al. [64]. Our protocol draft was used for conducting a pilot review in a Bachelor thesis project that was supervised by the protocol authors [36]. The pilot involved an automated search based on a search string that was derived from metadata of the publications documenting the 16 DSMLs we had considered up to that point. The pilot was limited to a small number of handpicked search engines (e.g. without considering their coverage) and it applied ad hoc restrictions on the search hits. The pilot especially helped us to revise this review protocol.

[1] Available at `http://wi.wu.ac.at/home/mark/BusinessActivities/library.html`; last accessed: Feb 2, 2015.

1 Motivation

A growing number of publications document the principles and the design process of domain-specific languages (DSLs; see, e.g., [71, 110, 115, 128]). A domain-specific modeling language (DSML) is (usually) a DSL with a graphical concrete syntax. DSMLs often focus on the (platform independent) problem concerns in the target domain rather than issues of implementing the domain (see, e.g., [5]). The DSML itself is built on top of an abstract syntax (i.e., the core language model) and typically developed using metamodeling techniques (rather than grammars, for instance). Often DSML developers use formal specification techniques to express the structural and behavioral semantics [58]. A DSML is then typically integrated into a model-driven development (MDD) tool chain (e.g., based on Eclipse).

In the following, we are particularly interested in DSMLs which are based on the Unified Modeling Language (UML) and are developed via UML extension mechanisms—native ones and beyond. Examples include pruning/reduction [102]), metamodel slicing [15], package referencing and merging [22, 28], as well as UML profiles [38, 84]. Due to the wide variety of different extension options, researchers started to document and reflect on best practices of UML-based DSML development (see, e.g., [5, 48, 67, 68, 83, 100, 101, 103, 111, 112, 121]).

In this context, different research approaches and research methods have been applied to systematically collect, organize, and review current (best and worst) practices, such as case studies [111], controlled experiments [112], critical-analytical studies based on a reference theory [103], and, more recently, SLRs [83]. However, so far these contributions focused on single elements of a DSML design in isolation (e.g., the concrete syntax for its cognitive effectiveness or patterns of structuring the abstract syntax).

Prior to this SLR study, we have already collected and documented development experiences for UML-based DSMLs in terms of a catalog of recurring design decisions [52, 53]. The corresponding *decision records* describe the decision *context* (e.g., a development phase or certain technology choices), state a repeatedly reported design *problem* regarding a DSML design element, and document design *options* to solve the problem. These decision records can the be referenced from design-rationale documents (e.g., decision templates or decision diagrams) documenting the decision making process for a particular DSML. A particular decision (and the corresponding design-rationale document) then refers to the respective design options (see Figure 1). All options adopted for a DSML are referred to as the *decision-option set* of this DSML.

However, there is currently no empirical evidence on the actual acceptance and adoption of different decision options (as identified by our catalog) in research-driven or industry-driven DSML development projects. In addition, the inter-dependencies between the design decisions (e.g., between designing the abstract and the concrete syntax) have not been investigated and documented empirically. In particular, inter-dependencies as indicated by recurring decision-option sets have not been addressed yet.

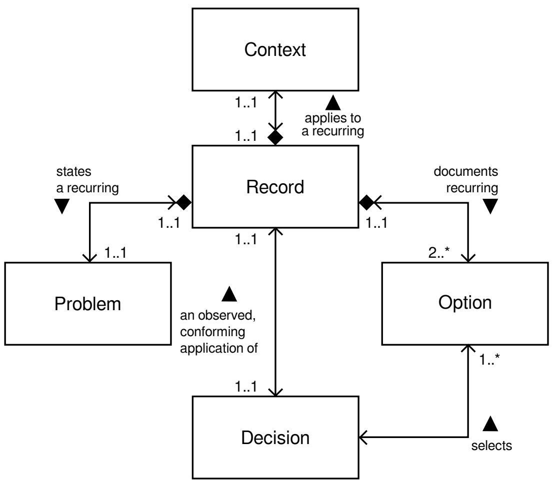


Figure 1: Conceptual overview of decision records, observed decisions, and decision option-sets.

Given the number of DSML artifacts (abstract syntax, language constraints, concrete syntax, behavior specification, and platform integration) and the iterative nature of DSML development [115], we propose a systematic review of scientific publications documenting the design processes of developing UML-based DSMLs. The aim of this systematic review is to collect a data set on actual design decisions and decision-option sets to document them and to evaluate our catalog of decision records. This way, we provide practical means in terms of decision records and characteristic solutions which help DSML engineers to reuse rationale preserved from prior design-decision making to document the rationale behind their own decision making.

2 Research Questions

We are motivated to establish whether decision-making processes for designing domain-specific modeling[l] (DSM)[l] languages can be adequatly documented in a structured and reusable manner via the *decision records* in our decision catalog (see Figure 1). Therefore, our research questions are:

RQ 1 Is the DSML decision catalog complete with respect to the DSML design documentation? Do the decision records [52, 53] provide sufficient coverage of decision options[d], decision drivers[d], and decision consequences[d] for the Meta Object Facility[t] (MOF[t]) [79] and the Unified Modeling Language[t] (UML[t])? In addition, is the format used to represent the DSML design-decision records [53, 52] sufficient?

RQ 2 Based on the decision records from our decision catalog: What are the recurring combinations of design decisions[d] and decision options[d] that can be mined from scientific papers and DSML project documentation on design-decision making[d] ?

3 Overview

To answer the above research questions, we perform a systematic literature review (SLR). I particular, our systematic review includes five major steps (see also Figure 2):

0. *Preparation*: In [52], we documented a draft version of our decision-record catalog for selected phases of DSML development. This version of the catalog was based on a narrow and subjectively chosen set of DSML projects (documented via corresponding publications). In this context, we also drafted a first protocol for a systematic review based on the guidelines from [65]. This draft was used for running a pilot review in a Bachelor thesis project that was supervised by the protocol authors [36]. The pilot study was limited to a small number of search engines (e.g., without considering their coverage) and it applied ad hoc restrictions on the search hits. The pilot study helped revise the protocol for the actual review run (see also Figure 2). In addition, we tried to identify comparable secondary studies (e.g., literature surveys, systematic reviews) on the same subject or on closely related subjects. While there is secondary research on designing DSMLs in the context of the UML (see, e.g., [5, 48, 67, 68, 83, 100, 101, 103, 111, 112, 121]), we did only find two methodically comparable studies: a systematic mapping study on DSLs in the broadest sense by Nascimento et al. [75] and a SLR on UML profiles [83]. However, while the former is not specific to UML-based DSMLs, the latter only reflects on a single UML extension technique. As a consequence, they did not qualify as adequate review designs for our SLR.

1. *Establishing a quasi-gold standard (QGS) corpus*: We will first build up a QGS corpus of reference publications, collected in a systematic process of manual search. A QGS can be used to evaluate the automated search conducted in an SLR (see [129]). The main objectives when using a QGS are to quantify the retrieval quasi-sensitivity (see below) of the automated search (per search engine, overall) and to provide means for refining the search terms in a stepwise manner. To create the QGS corpus, we will select top publication venues from our seed publications (see [52]), our pilot review (see [36]), and the third-party mapping study we identified (see [75][2]). By manually screening the corresponding proceedings and archives, we will then identify the publications for the QGS. Further details on the planned QGS procedure are reported in Section 4.

2. *Performing the automated search*: We will then apply the search strategy as elaborated on in Section 4. This involves defining (and continuously refining) the set of search terms, translating them into search-engine-specific representations, running the individual search operations, and collecting the results from the search engines in a processable manner. Based on the quasi-sensitivity computation allowed by the previously constructed QGS corpus, the automated search will be ended as soon as the quasi-sensitivity threshold is reached. Following Zhang et al. [129], the quasi-sensitivity is defined as the ratio of relevant papers identified by the QGS retrieved through the automated search to the total number of relevant papers (i.e., the size of the QGS corpus). The threshold to be reached is a quasi-sensitivity level between 70% and 80% [129].

3. *Performing the first literature review*: In our SLR, reviewing the search hits involves both filtering them according to predefined selection criteria (see Section 6) and extracting design-decision data from the included search hits. The literature review is therefore tightly coupled with the step of data extraction (see Section 7).

[2] We received the data set of publications collected during this mapping study in private communication from the authors of [75]

4. *Performing backward snowballing to identify additional DSML projects*: To further extend the coverage of our SLR, we will perform an additional manual review of the bibliographies (i.e. the reference sections) extracted from the publications of the automated search. The goal of this backward snowballing [120] is to reduce the risk of missing relevant DSML development projects (see Section 5 for details).

5. *Performing a second literature review*: In a last step, we will review the snowballing hits, again by applying the respective selection criteria and by extracting the decisions according to our data-extraction guidelines (see Section 7).

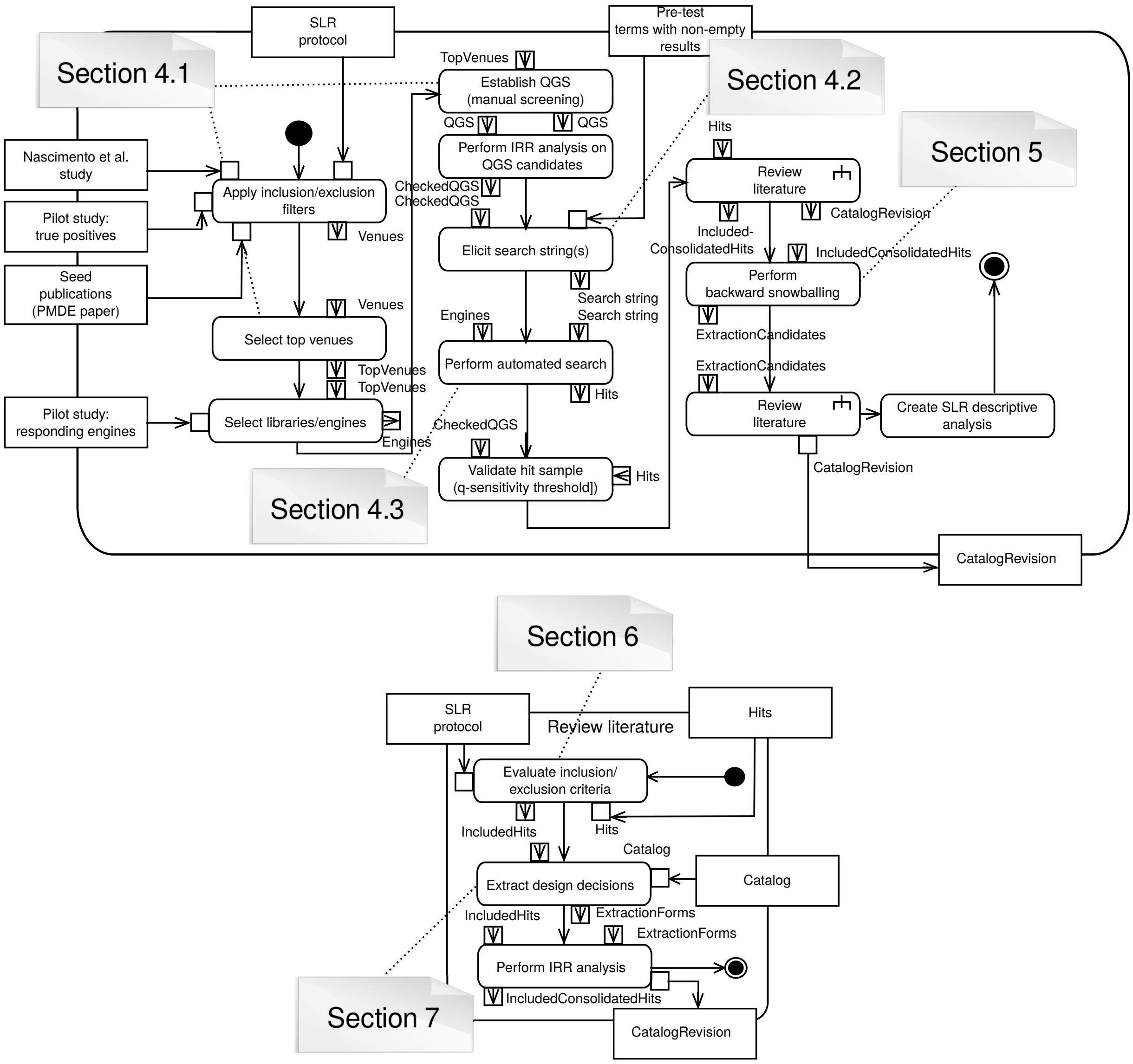


Figure 2: Procedural overview of the systematic review and the corresponding protocol sections providing the details on the procedural steps.

At all stages of the review procedure, we will make sure that we follow established guidelines for designing, conducting, and reporting SLRs (in particular, [60, 65, 120, 129]). In addition, we use different, accepted statistical techniques of inter-rater reliability (IRR) analysis (e.g., the Kupper-Hafner Index [66]) to report on the level of agreement and disagreement, respectively, between the experts involved in reviewing publications for inclusion/exclusion and for extracting design decisions.

4 Search Strategy

In revision 3 of this protocol, the original search strategy was superseded by a QGS-based one. The original search strategy (revision 2) consisted of an subjective construction of a search string and an ad hoc selection of search engines. We report the details of the original search strategy in Section 4.0

to the extent the results of the original automated search entered the revised SLR design (e.g., to elicit the search string). The authoritative, QGS-based search strategy is documented starting with Section 4.1. All details of the pilot study are documented in Filtz [36].

Valid search terms considered in our SLR are both atomic and composite words, separated by whitespace or dash characters.

4.0 Pilot Search

We defined three groups of search terms: terminology denoting a relevant (meta-)modeling technology (technology^{t}), terminology specific to (meta-)modeling language engineering (language engineering^{l}), and terminology indicating the documentation of design rationale in general (documentation^{d}). To form the actual, complex search string, we brought these three subsets into a disjunctive-composite form (see below). We derived the actual search terms from the following sources:

1. *Research questions*: Derive major terms from the research questions. The terms derived from the five research questions are highlighted in Section 2.

2. *Primary studies*: Extract terms from the title, abstract, and keywords (if any) from the 18 seed publications, available at that time (Oct. 2012), on the 16 DSML projects that we considered initially (see also Section 4.1 and Table 2).

[114] Extended Activity Models^{t}, ~~Interdependent Concern Behavior~~

[113] ~~Process Modeling~~, ~~Role-based Access Control~~, ~~Role Engineering~~, ~~Systems Modeling~~, UML^{t}, Extended UML Activity Models^{t}, ~~Process-related RBAC Models~~

[93] ~~Process-Related Duties~~, Extended UML Activity Diagrams^{t}, Extended UML Interaction Diagrams^{t}

[94] ~~Access Control~~, ~~Business Processes~~, ~~Delegation~~, ~~RBAC~~, ~~Consistent Delegation~~, ~~Process-Aware Information Systems~~

[92] ~~Access Control~~, ~~Delegation~~, ~~RBAC~~, UML^{t}, Modeling Support^{l}, ~~Roles~~, ~~Tasks~~, ~~Duties~~, ~~Process-Related RBAC Context~~

[50] Modeling Support^{l}, ~~Confidentiality~~, ~~Integrity~~, ~~Object Flows~~, Activity Models, ~~Security Properties~~, ~~Process Modeling~~, UML

[91] ~~Consistency Checks~~, ~~Duties~~, Extended UML2 Activity Models^{t}, ~~Binding of duty~~, OCL^{t} (**Object Constraint Language**^{t}), ~~RBAC~~, ~~Security~~, ~~Separation of Duty~~, UML^{t}

[49] ~~Integrity~~, ~~Confidentiality~~, Annotations^{l}, ~~Service Interfaces~~, ~~Service-Oriented Architecture~~, ~~Security Engineering~~, UML^{t}, ~~Web Services~~, SoaML^{t} (**Service-oriented architecture Modeling Language**^{t}), Model-Driven Development^{l}

[55] ~~Secure Object Flows~~, ~~Process-driven SOAs~~, Integrated Model-driven Approach^{l}, ~~Process modeling~~, ~~Secure object flows~~, ~~Security engineering~~, ~~Service-oriented architecture~~, Model-driven Development^{l}, UML^{l}, SoaML^{t}, ~~Web services~~

[95] ~~Context-Aware RBAC Models~~, ~~Business Processes~~, ~~Ubiquitous Computing Environments~~, ~~Access Control~~, ~~Business Process Modeling~~, ~~Context Constraints~~, UML^{t}

[51] UML Extension^{l}, Model-driven Specification^{l}, ~~Audit Rules~~, Model-driven Development^{l}, UML^{t}

[127] ~~Composition~~, ~~Dynamic Programming Environments~~, ~~Model Transformations~~

[8] ~~Model Driven Security~~, UML Models^{t}, ~~Access Control Infrastructures~~, Software Engineering^{l}, Requirements/Specifications Languages^{l}, Requirements/Specifications Methodologies^{l}, Requirements/Specifications Tools^{l}, Design Tools^{l}, Design Techniques^{l}, Computer-aided Software Engineering^{l}, ~~Object-oriented Design Methods~~, ~~Management of Computing and Information Systems~~, ~~Security~~, ~~Protection~~, Design^{l}, Languages^{l}, ~~Security~~, ~~Role-Based Access Control~~, Model Driven Architecture^{l}, Unified Modeling Language^{t}, Object Constraint Language^{t}, Metamodeling^{l}, ~~Security Engineering~~

[62] ~~Secure Systems Development~~, UML^{t}

[43] ~~Security Engineering~~, ~~Service-Oriented Architectures~~

[37] MDA^{l} (**model-driven architecture**^{l}), ~~Access Control Specifications~~, MOF^{t}, UML Profiles^{t}, MDD^{l}, UML^{t}, ~~CORBA~~, ~~J2EE~~, ~~Security~~

[88] ~~Business process~~, ~~Security requirement~~, UML 2.0[t], Activity Diagrams[t], ~~Secure Business Process Model Specification~~, Activity Diagram Profile [t]

[35] ~~Data Warehouses~~, Multidimensional Modeling[l], ~~Access Control~~, ~~Audit~~, UML[t], ~~Audit Model~~

3. *Secondary studies*: Extract terms from the title, abstract, and keywords (if any) from four secondary studies on DSML development and DSML design processes. This group of terms aims primarily at identifying search terms specific to explicit design-rationale documentation for DSMLs.

[80] Modeling Language Design[l], Modeling languages[l], Design principles[d], UML[t], Unification[l]

[87] ~~Lightweight Approach~~, Domain-Specific Modeling Languages Design [l], Model-Driven Engineering[l], UML[t], profiles[t], Domain-Specific Modeling Languages[l]

[100] ~~Systematic Approach~~, Domain-Specific Language Design, UML, ...

[20] Customizing[l] (**customization**[l]), UML[t]

[115] Systematic Development[l], Domain-Specific Languages[l], ~~applied software engineering~~, Model-Driven Software Development[l] (**MDSD**[l]), Language Engineering[l]

After merging the three subsets, data cleansing was performed—this esp. includes:

- Removing duplicates
- Reducing plural to singular forms
- Appending the extended writing for acronyms and vice versa
- Adding alternative noun forms for gerunds
- Turning uppercase into lowercase writing

Terms added due to data cleansing are written in bold font face in the term list above. This way, we obtained the search string in Listing 1.

Listing 1: The initial search string used for the pilot search (see [36]).

```
(meta object facility or MOF or unified modeling language or UML or
object constraint language or OCL or profile or UML profile or UML 2.0
or UML2 or UML 2 or extended activity diagram or extended activity
model or extended UML activity diagram or extended UML interaction
diagram or activity diagram profile)
and
(domain-specific modeling language or DSML or model-driven development
or MDD or model-driven engineering or MDE or modeling language design
or modeling language or unification or domain-specific modeling
language design or customizing or customization or model-driven
software development or MDSD or language engineering or modeling
support or activity model or SoaML or service-oriented architecture
modeling language or integrated model-driven approach or extension or
requirement language or specification language or requirement
methodology or specification methodology or requirement tool or
specification tool or design tool or model-driven architecture or MDA
or metamodeling)
and
(design decision or design-decision making or decision option or
decision driver or decision consequence or design documentation or
design principle or systematic development or software engineering or
design technique or computer-aided software engineering)
```

The resulting, complex search string comprised three clauses (technology, language, documentation) grouping 58 atomic and composite search terms. When expanded from its disjunctive-composite form, the search string yielded 5,456 unique search triples. Each triple consists of three terms, one from each clause.

4.1 Quasi-Gold Standard Corpus (QGS)

For our SLR, we adopt the procedural guideline for QGS corpus construction by Zhang et al. [129]. This involves the following steps:

1. *Identify publication venues*: In a first step, we establish a pool of potential publication venues (journals and conference series) which are then screened for publications relevant to our research questions. In particular, we compiled the venue pool from three different sources: 1) the mapping study by Nascimento et al. [75], 2) the venues of our own DSML publications, and 3) the venues identified during our pilot review.

   Mapping-study pool. Nascimento et al. [75] kindly provided their study data to us, in private communication. For their study, they collected and reviewed 2,688 publications from 669 presence venues (i.e. conferences and workshops) and 180 archival venues (i.e. journals). The pool of potential venues is created in a procedure of three filtering steps (see also Table 1):

(Step 1) From the initial set, we consider those publications (and the respective venues) which were finally included into the mapping study by Nascimento et al. [75]. This amounts to 435 presence and 131 archival venues for, in total, 1,440 papers.

(Step 2) We further restrict the manual screening to those venues that are specific to DSMLs. Nascimento et al. [75] tagged these publications accordingly, allowing us to filter for the corresponding tag "DSML" in their data set. This gives us a subset of 91 presence and 23 archival venues, accounting for 163 DSML-specific publications—according to the working definitions and the coding applied by Nascimento et al. [75].

(Step 3) Nascimento et al. [75] categorized the publications in their data set into six types of research reports: validation research, evaluation research, solution proposal, philosophical papers, opinion papers, and experience papers (see Table 3 in [85]). For our SLR, we extract those publications (and their respective venues) which are likely to contain documented DSML design rationale: *solution proposal* or *experience paper*. This is because solution proposals (which document a novel DSML-based technique or an important extension to an existing DSML-based approach) and experience research reports (which reflect critically on DSML implementations in practice, based on the personal experience of the DSML engineers) are the two categories most likely to contain papers documenting design rationale. This gives us a final subset of 80 presence venues and 21 archival venues (contributing 137 publications) for the subsequent inclusion and screening steps. They are listed by number and their number of entries in the original data set in the two first subsections of Table 4 and 3, respectively. We, therefore, consider 11.9% of the venues found in the original data set by Nascimento et al. [75], accounting for 5.1% of the total publications, for our QGS corpus construction.

> When performing the identification substep, we found that two of the 21 journals selected from the mapping-study pool are monographs and were therefore mislabeled journal volumes [75]: Graph Transformations and Model-Driven Engineering and SENSORIA (see Table 4). We excluded the respective sources from the subsequent data collection steps.

Table 1: Summary of filtering steps performed over the data set by Nascimento et al. [75]; presence venue: conference, symposium, workshop; archival venue: journal, monograph.

Number of ...	Papers	Presence venues	Archival venues
Complete data set	2,688	669	180
Step 1: Included subset	1,440	435	131
Step 2: DSML-specific subset	163	91	23
Step 3: DR-specific subset	137	80	21

Seed publications. Based on ten DSML projects that have been conducted in our research group over a period of five years (2008–2012), we established a first version of our decision catalog ([52]; see P1–P10 in Table 2). In addition, we identified six third-party DSML approaches (see P11–P16 in Table 2) via backward snowballing. As a result, we consider a total set of 19 corresponding publications for identifying QGS venues. From these 19 publications, 13 publications document the authors' DSML projects and six publications cover the six related projects (see Table 2). Seven out of the 19 publications are journal articles and, therefore, we add six additional journals (which have not been found in the mapping-study corpus [75], see above) to the pool of QGS candidate journals (see Table 5).

Table 2: Overview of seed publications.

	Objectives	Domain
	Authors' DSML projects (P1–10)	
P1	An approach to model interdependent concern behavior using extended UML activity models [114].	Separation of concerns
P2	An integrated approach for modeling processes and process-related RBAC models (roles, hierarchies, statically and dynamically mutual exclusive tasks etc.) [113].	Business processes, role-based access control (RBAC)
P3	A UML extension for an integrated modeling of business processes and process-related duties; particularly the modeling of duties and associated tasks in business process models [93, 94].	Business processes, process-related duties
P4	An approach to provide modeling support for the delegation of roles, tasks, and duties in the context of process-related RBAC models [94, 92].	Business processes, delegation of roles, tasks, and duties
P5	A UML extension to model confidentiality and integrity of object flows in activity models [50].	Data confidentiality and integrity
P6	UML modeling support for the notion of mutual exclusion and binding constraints for duties in process-related RBAC models [91].	RBAC (consistency checks for duties)
P7	Incorporation of data integrity and confidentiality into the MDD of process-driven SOAs [49, 55].	Integrity and confidentiality for service invocations
P8	Integration of context constraints with process-related RBAC models and thereby supporting context-dependent task execution [95].	Business processes, RBAC, context constraints
P9	A generic UML extension for the definition of audit requirements and specification of audit rules at the modeling-level [51].	Audit rules
P10	An approach based on model transformations between the valid structural and behavioral runtime states that a system can have [127].	Model transformation
	Related, third-party DSML projects (P11–16)	
P11	A combination of UML modeling languages with security modeling languages for formalizing access control requirements [8].	RBAC
P12	A profile-based UML extension for secure systems development which can be employed to evaluate UML specifications for vulnerabilities using a formal semantics [62].	Multiple security properties (e.g., integrity, secrecy, authenticity etc.)
P13	An approach for the modeling and implementation of security-critical applications and inter-organizational workflows using model-driven security [43].	SOA security
P14	A MDD approach for the development of access control policies for distributed systems [37].	View-based access control
P15	An extension to UML activity diagrams which allows security requirements to be specified in business processes [88].	Secure business process modeling
P16	A UML extension to specify access control and audit properties in the multi-dimensional modeling of data warehouses [35].	Access control, authorization, and audit rules

Pilot publications. From out pilot corpus, we further consider three additional presence venues and six archival venues. These venues were uniquely found in our pilot and therefore those venues have not yet been included (see above).

To summarize: In order to establish the QGS corpus, we review 86 presence venues (conferences, symposia, workshops; see Table 3) and 33 journals (see Table 4) identified for 159 publications in three different publication corpora.

Table 3: Overview of candidate conference venues for QGS construction.

	Name of venue	Entries #	Issues #	Listed with		Comment
				MS/AS (rank #)	Xie	
	Nascimento et al. [75]					
*	MoDELS	7	16	55	x	
	ECBS	5	19			
	VL/HCC	5	28		x	
	EDOC	4	16		x	
	CAiSE	3	24		x	
	SLE	3				
*	ASE	2	27	15	x	
	SAC	2	27	17		
	ACMSE	2	50			
*	GPCE	2	11	42	x	no screening hits
	ICEIS	2	14			
	CEC	2	9		x	

	Name of venue	Entries #	Issues #	Listed with		Comment
				MS/AS (rank #)	Xie	
	ICALT	2	12			
*	COMPSAC	2	36	25	x	
	EDOCW	2				
*	ECMFA, ECMDA-FA	2	8	95	x	
	ICSE	2	34	1	x	
*	EMSOFT	1	12		x	
	DEBS	1	32			
	AAMAS	1	19			
	(COP)	1				workshop
	SenSys	1	10			
	FML	1				workshop
	ECSA	1		164		
	DSM	1				workshop
	ATIO	1	12			
*	SAFECOMP	1	31	75	x	
	CCCM	1				
	ESAW	1				workshop
	OTM	1	9			
	SYNASC	1	13			
	ISC	1	10			
	SOCC	1	9			
	DSVIS	1			x	
	CSEDU	1				
	WaGe	1				workshop; collocated with GPC
	AOSE	1			x	workshop
	SCCC	2	29	141		
	CEA	1				
	EFTA	1	17			
	IAT	1	11			
*	REFSQ	1	18	88	x	no screening hits
	ICINCO	1	9			
	DATE	1	17			
	PERCOM	1	10			
	DAC	1	49			
	AMOST	1				workshop; collocated with ICSTW
	ITSIM	1	8			
	APSCC	1				
	ICIT	1	18			
	ISORC	1	15			
	SOSE	1				
	INDIN	1	11			
	ITNG	1	11			
	WISES	1				workshop
	ASRU	1				
	WSC	1	23			
	ICSEA	1		184		
	FDL	1	15	127		
	HICSS	1	45			
	ICCSE	1	8			
	FTDCS	1				workshop
*	OOPSLA/ SPLASH	1	26	4	x	
	ICSOC	1	10		x	
	ESWC	1	8			
	ICOODB	1				
	ISEC	1				
	ER	1	31		x	
*	SERP	1	8	92	x	
	SoMeT	1	11	245		
	SIGMOD	1	37			
	DETC	1	9			
	EGOVIS	1				
	BMFA	1				workshop; published in ICPS
	EICS	1				
	RTSS	1	32			
	ECDL	1	16			
	CRIWG	1				workshop
	AGTIVE	1				
	ICWE	1	12		x	
10	80	113	53	16	20	

	Name of venue	Entries #	Issues #	Listed with		Comment
				MS/AS (rank #)	Xie	
			Seed publications			
	MUSIC	1	3			
	SC	1	11			
	BIS	2	15			
0	3	4	3	0	0	
			Pilot publications			
	CIT	1	12			
	DaWaK	1	14			
	SysCon	1				
0	3	3	2	0	0	
10	86	120	28 (≤ 7)	16	20	11 workshops

Table 4: Pool of candidate journals for QGS; **ERA** [7]: Excellence in Research for Australia Journal Ranking 2012 (subset of 304 journals categorized into "Computer Software (803)" and/or "Information Systems (806)"); **MS/AS** [72]: Microsoft Academic Search (subset of 53 journals in category "Computer Science"/"Software Engineering"; Feb 4, 2013); **SJR** [99]: SCImago Journal Rank (subset of 184 journals for subject area "Computer Science", subject category "Software", 2012).

	Journal title	Entries #	Listed with		
			ERA	MS/AS (rank #)	SJR (rank #)
			Nascimento et al. [75]		
∗	TSE	3	x	1	7
	IJSEKE	2	x	29	
	ComSIS	2			
	TOMACS	1			88
	ISSE	1		44	96
	JSUSE	1			
	Computer	1			
	TTBE	1			
∗	IETSoftw	1	x	7	92
	CJA	1			
	(Graph Transformations and Model-Driven Engineering)				
	PRC	1			
	(SENSORIA: Rigorous Software Engineering for Service-Oriented Systems)				
	Simulation	1	x		103
	TII	1	x		
	TEC	1	x		
	TC	1	x		
∗	AES	1	x	13	65
	JSA	1	x		
	VLC	1	x		
	IJCIS	1	x		
3	19 (21)	25	11	5	6
			Seed publications		
∗	JOT	1	x	28	115
	IST	1	x		
∗	SoSyM	1	x	21	30
∗	TOSEM	1	x	5	13
	ENTCS	1	x		
	DSS	2	x		
3	6	7	6	3	3
			Pilot publications		
∗	JSS	1	x	6	40
	IJWIS	2	x		
	DKE	1	x		
	JUCS	1	x		
	IJOR	1	x		
	ISF	1	x		
1	6	7	6	1	1
7	31 (33)	39	23	9	10

2. *Select publication venues* Two researchers review the pool of publication venues to decide whether to screen them for publications to be included into the QGS corpus. The corresponding selection criteria are discussed in Section 6.1. Note that for establishing the QGS corpus, criteria evaluation is tightened as compared to later steps (e.g., engine-based search; see Section 6.4.1). In summary, the following major criteria are evaluated.

   Archival venues (Journals)

   - *Community relevance*: A journal must be acknowledged as relevant by a broad, public scientific audience (rather than by the reviewers alone); see C_2 in Section 6.1.

     23 of the 31 journals are listed with ERA journal list [7].

   - *Software-engineering focus*: A journal must have a clear and acknowledged focus on software-engineering (SE) topics. This is because we are interested in sources on DSML design rationale taking a SE perspective, rather than a domain-specific angle (e.g., a DSML's application and evaluation in a particular application domain); see C_3 in Section 6.1.

     Seven of the 23 ERA-listed journals [7] are contained in both SE venue lists. That is, 16 journals which are not listed in both sources were dropped at this stage.

   - *Content maturity*: It must be a peer-reviewed journal; see C_4 in Section 6.

     All seven remaining venues have a peer-review scheme in place.

   Results: After having applied the two criteria, **seven out of 31 journals** were finally considered for the manual screening of QGS publications. The seven selected journals are marked by "*" in Table 4.

   Presence venues (Conferences, symposia, workshops)

   - *Community relevance*: The venue must have a regular publication history with respect to our review period; see C_2 in Section 6.1.

     28 out of the remaining 75 venues have not been held regularly during our review period (and before).

   - *Software-engineering focus*: For the same reasons given for journals (see above), we only consider venues having an accepted SE focus (in the most inclusive sense); see C_3 in Section 6.1.

     37 out of the remaining 47 venues are not listed by the two SE conferences list.

   - *Content maturity*: Workshops are excluded; see C_4 in Section 6.1.

     Eleven out of the 86 venues have a workshop format.

   Results: After having applied the three criteria, **ten out of 86 venues** were finally considered for the manual screening of QGS publications. The ten selected venues are marked by "*" in Table 3.

3. *Manually search selected venues for gold-standard publications*: In a first round, two authors (Hoisl, Sobernig) screen all papers published in the previously selected publication venues in the review period (2005–2012). The selection criteria C_5–C_{11} are reported in full detail in Section 6.1 and, specifically, in Section 6.4.1. Each author takes on a randomly chosen and possibly equally sized subset of journals and conferences for screening. Each author then revisits the paper candidates proposed by the other author in a second round. In case of disagreement on a paper's inclusion or exclusion, the two authors jointly review the respective papers and find a common judgement (e.g., based on the full-text of the paper) in a third round. The level of agreement between the two authors about the candidate list of QGS papers is documented using the Kappa inter-rater statistic.

Table 5: Overview of venues manually searched for quasi gold-standard (QGS) publications, the number of publications considered for inclusion, and the publications actually incorporated into the QGS corpus; effective date of manual screening: Jan 25, 2013.

Venue (# Num./Iss.)	Publisher	Engine	Papers		
			#fnd	#incl.	References
Journals					
TSE (31.1–39.1; 65)	IEEE	IEEEXplore	5	2	[29, 124]
IETSoftw (1.1–7.1; 36)	IEEE	IEEEXplore	5	3	[73, 74, 104]
AES (36.1–42.12; 43–55; 97)	Elsevier	Scopus	2	0	
JOT (4.1–12.1; 56)	AITO	Scopus	13	7	[19, 24, 33, 61, 108, 116, 117]
JSS (74.1–86.1; 100)	Elsevier	Scopus	16	6	[25, 45, 86, 89, 118, 119]
TOSEM (14.1–21.4; 32)	ACM	ACMDL	2	1	[105]
SoSyM (4.1–11.4; 32)	Springer	SpringerLink	9	5	[1, 14, 39, 63, 132]
|Ven|: 7	|Pub|: 5	|Eng|: 4	Σ 52	Σ 24	
Conference proceedings					
MoDELS (8)	Springer	SpringerLink	8	3	[27, 32, 131]
COMPSAC (8)	IEEE	IEEEXplore	4	2	[125, 130]
ECMFA, ECMDA-FA (8)	Springer	SpringerLink	9	3	[10, 30, 59]
SERP (8)	CSREA Press	n/a	1	0	
OOPSLA (8)	ACM	ACMDL	1	1	[126]
SAFECOMP (8)	Springer	SpringerLink	6	4	[12, 13, 46, 56]
ICSE (8)	ACM	ACMDL	2	0	
REFSQ (8)	Springer[3]	SpringerLink	0	0	
GPCE (8)	ACM	ACMDL	0	0	
ASE (8)	IEEE	IEEEXplore	0	0	
10	|Pub|: 4	|Eng|: 4	Σ 31	Σ 13	
17	|Pub|: 6	|Eng|: 4	Σ 83	Σ 37	

Table 5 gives an overview of the scientific venues, journal volumes and conference proceedings, screened for publications qualifying to be included into the quasi-gold standard corpus:

- In total, we reviewed 418 volume numbers of the seven journals selected in the previous step.
- 52 journal articles were initially considered QGS candidates by the two reviewing authors. These included articles from each of the seven journals.
- 24 out of 52 journal articles were finally included into the QGS corpus, representing six out of seven journals.
- We mined 80 proceeding issues of ten conferences.
- In total, 31 proceedings articles were considered QGS candidates taken from seven conferences. For three conferences (ASE, REFSQ, GPCE), the screening substep did not yield any search results.
- 13 out of the 31 proceedings articles were adopted as QGS publications, representing five out of ten venues.
- The workload of publication screening was distributed equally. Out of the 17 screened venues, nine were checked by Hoisl (four journals, five conferences), eight by Sobernig (three journals, five conferences).
- The QGS corpus, finally, consisted of 37 publications (i.e., 24 journal and 13 proceedings articles).
- The decision on including or excluding each of the 83 (52+31) publication items was initially performed by two authors of this SLR as raters separately, yielding 166 ratings in total.
- For the 52 journal articles, the two raters showed perfect agreement on ≈79% of the publication items (percent agreement, $p_a = 0.789$). Chance-corrected agreement is at medium level $\kappa = 0.578$ (see Table 6a).

[3]Only covers years 2007 through 2012; 2005 and 2006 were self-published by a German university press.

- For the 31 conference articles, the two raters agreed in 71% of the cases (percent agreement, $p_a = 0.71$), with $\kappa = 0.395$ (see Table 6b).
- In total, for all 83 articles, the two raters agreed in 75.9% of the cases (percent agreement, $p_a = 0.759$), with $\kappa = 0.513$ (see Table 6c).

Table 6: Distribution of publications by rater; each article was rated according to a two-level scale: y(es) for inclusion, n(o) for exclusion; p_a: percent agreement; $\widehat{\kappa_c}$: Cohen's Kappa statistic; $Var(\widehat{\kappa_c})$: leave-one-out Jackknife estimate of variance of Cohen's Kappa statistic.

(a) 52 journal articles; $p_a = 0.789$; $\widehat{\kappa_c} = 0.578$; $Var(\widehat{\kappa_c}) \approx 0.0131$

		Rater B		
		n	y	Total
	n	20	7	27
Rater A	y	4	21	25
	Total	24	28	52

(b) 31 conference articles; $p_a = 0.71$; $\widehat{\kappa_c} = 0.395$; $Var(\widehat{\kappa_c}) \approx 0.0276$

		Rater B		
		n	y	Total
	n	7	2	9
Rater A	y	7	15	22
	Total	14	17	31

(c) 83 articles; $p_a = 0.759$; $\widehat{\kappa_c} = 0.513$; $Var(\widehat{\kappa_c}) \approx 0.0092$

		Rater B		
		n	y	Total
	n	27	9	36
Rater A	y	11	36	47
	Total	38	45	83

4. *Identify search engines*: Depending on the venues selected for QGS publication screening, we identify the candidate search engines for the automated search as follows:
   (a) For each publisher, an authoritative search engine is considered.

   The six publishers of the publication venues considered for screening were (ordered by the number venues) Springer (5 publication venues), ACM (4), IEEE (4), Elsevier (2), AITO (1), and CSREA Press (1); see also Table 5.

   For these six publishers, the following four authoritative engines were selected (in the same order as above): SpringerLink (Springer), ACMDL (ACM), IEEEXplore (IEEE), and Scopus (Elsevier, AITO). To the best of our knowledge, there is no dedicated search engine available which covers the CSREA proceedings.

   Figure 3a reports the per-venue overlap between these four search engines.

   (b) We only consider search engines from which at least one publication entered the QGS corpus. This is because quantifying search validity (precision/recall) is only representative for search engines likely to yield a QGS corpus publication.

   When considering the venues and engines of the finally adopted QGS corpus publications (see above), there remain five unique publishers: Springer (4), IEEE (3), ACM (3), Elsevier (1), and AITO (1).

   For these five publishers, the four authoritative engines initially identified remain unchanged: SpringerLink (Springer), IEEEXplore (IEEE), ACMDL (ACM), and Scopus (Elsevier, AITO).

   Figure 3b reports the remaining per-venue overlap between these four search engines, after having removed venues which have not entered the QGS. This way, there was no longer an overlap between ACMDL and IEEEXplore. Besides, the overall level of per-venue overlap was reduced (from ten to seven overlapping venues).

   (c) Engines reported as accepted and as widely adopted in primary SLR studies take precedence [18].

   (d) Engines which are publicly accessible for SE researchers (e.g., without institutional subscription) is given preference, to facilitate repeating the automated search in replication studies.

Note that we refer to public access to the search facilities, and not e.g. to the full-text collection, if any.

> Both criteria would have clearly favoured adopting ACMDL and IEEEXplore. In addition, the ACMDL comes with an extended data base[a] which potentially covers the required non-IEEE and the required non-ACM venues. Therefore, we further assessed the possibility of limiting our automated search to these two engines:
>
> - The extended ACMDL would overlap with IEEEXplore regarding two venues (TSE, COMPSAC) from the QGS corpus. The COMPSAC overlap would have been specific to a subset of COMPSAC proceedings. ACMDL only incorporates twelve COMPSAC proceedings out of 36 COMPSAC proceedings (as of 2012, according to DBLP). To reduce overlap, we limited the search in the ACMDL to the ACM-only (base) full-text collection.
> - Three out of four Springer venues (MoDELS, ECMFA, SAFECOMP) are not covered by neither the extended nor the base ACMDL. Therefore, SpringerLink could not be removed from the engine list.
> - The Elsevier and AITO venues (JOT, JSS) are also not covered by the base ACMDL, requiring us to incorporate the Scopus engine as fourth one.
> - Scopus, however, has a large, although often partial, overlap with the ACMDL and IEEEXplore. Scopus and the extended ACMDL share nine QGS venues. Four out of those nine venues are only partially covered by Scopus (e.g., varying year ranges). For the base ACMDL, the overlap amounts to two venues: TOSEM and OOPSLA/SPLASH representing twelve QGS publications (see also Figure 3b). For these two, the overlap covers all venue years. Between IEEEXplore and Scopus, there is an overlap of another three venues: TSE, IETSoftw, and COMPSAC representing four QGS publications (see also Figure 3b). Two out of the three overlaps turned out to be partial. With SpringerLink, Scopus shares one venue (SoSyM; seven QGS publications).
>
> The overlap between Scopus, on the one hand, and the ACMDL, IEEEXplore, and SpringerLink, on the other hand, risks introducing duplicate hits in the automated search, which must be controlled explicitly.
>
> [a] The base collection is named *ACM Publications and Affiliated Organizations*, the extended one *ACM Guide to Computing Literature*.

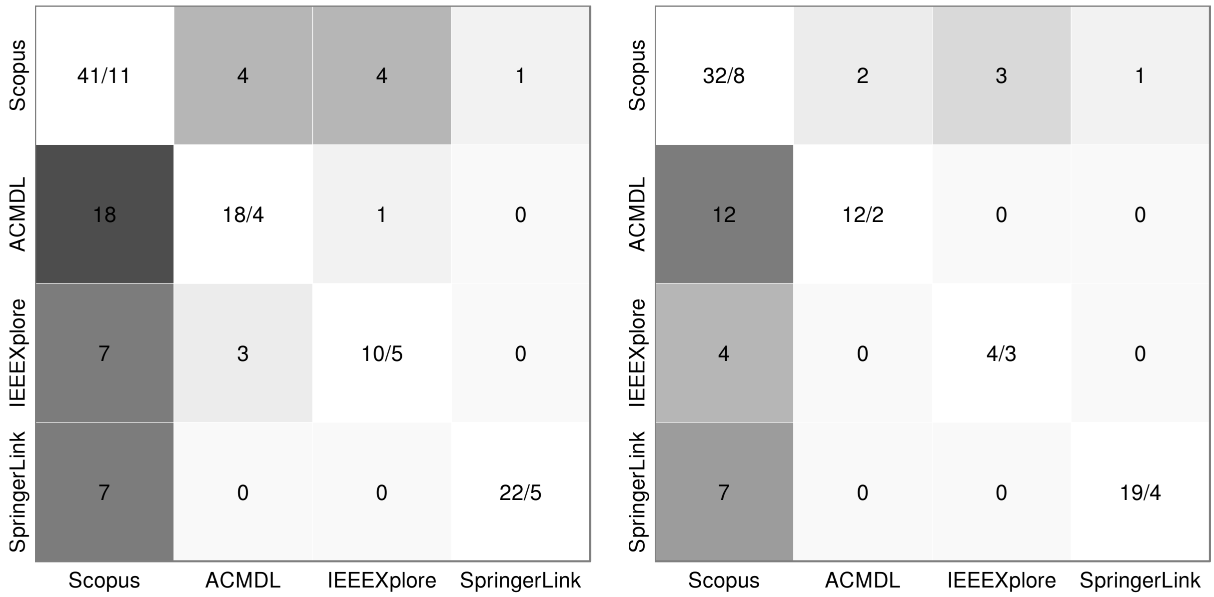
(e) When multiple engines per venue are available, the one providing the most complete time coverage in terms of publication years (i.e., 2005–2012) takes precedence.

> At this decision point, no alternative engines for a given venue providing different time coverages remained.
>
> Venues only partly (SERP, REFSQ) covered by any of the available search engines had already been excluded in an earlier decision step (see above).

(f) The objective is to adopt a minimum number of search engines covering a maximum number of venues in a minimally overlapping manner (see [129]).

> For three engines—ACMDL (in its base resources collection), IEEEXplore, and SpringerLink—there was no overlap in terms of QGS venues. The need to consult Scopus to include two critical QGS venues (JSS, JOT), however, introduces substantial overlap. Reducing this overlap by limiting our automated search to Scopus only was not an option because Scopus lacks complete coverage of certain ACM and IEEE venues (IETSoftw, COMPSAC), as well as all Springer venues. To mitigate the risks resulting from this unavoidable overlap, we thoroughly inspected the results of the automated search for duplicates.

> Based on the decisions above, the following search engines were selected for the automated-search step:
>
> - ACM Digital Library (ACMDL)
> - IEEEXplore
> - Scopus



(a) Based on all found articles (QGS candidates).

(b) Based on included articles (QGS publications).

Figure 3: Matrix plots showing the per-venue (upper segment) and per-publication (lower segment) overlap between search engines. The diagonal depicts the non-unique publications/venues covered by each search engine.

- SpringerLink

4.2 Defining and Refining Search Terms

Starting with revision 3, the new strategy to construct the search string superseded the original strategy reported in Section 4.0. The new strategy involved using the systematically selected QGS corpus when constructing the final search string. The main advantage of this QGS-based strategy is avoiding an authors' bias in formulating the search string, which could otherwise result in search hits which are close to the terminology employed by the authors and found in application domains the authors are primarily looking at. In addition, the pilot search allowed us to refine certain details of the revised search strategy. These deviations from the procedure summarized in Section 4.0 are highlighted below.

We derive the initial search terms in multiple steps:

1. We extract terms from the publication items in the previously established QGS corpus. We mine the titles, abstracts, and publication-specific keywords (authors' keywords, publisher classification items) for critical terms. The publications in the QGS corpus represent primary sources, i.e. they document individual and multiple DSML designs in the context of their respective application domains. We, therefore, adopt a primarily subjective strategy to define the search terms (see [129]).

2. We then derive additional or complementary terms from the research questions, as stated in Section 2.

The revised procedure differed from the original in one important manner (see Section 4.0): We had learned from the pilot [36] that querying for terms that indicate the explicit description of DSML design rationale (e.g., "decision consequence") is not likely to result in search hits. This reflects the more general observation in related literature that design rationale and its documentation are often not made explicit; and even if design rationale is documented, it is not reported using an agreed upon standard vocabulary. At the same time, the complexity of the expanded search string was considerable (i.e., 5,456 tuples including the third clause vs. 496 excluding the third clause). Therefore, we dropped the third clause and with it the secondary studies as sources to extract search terms for our SLR.

Terms obtained from step 1: The initial set of search terms contains the following technical terms found in the titles, abstracts, and keywords of the publications forming the QGS corpus (see

Section 4.1):

[124] metamodel[l], **meta-model**[l], **meta model**[l]

[29] design pattern[d], stereotype[l], ~~constraint~~, profile[l], Unified Modeling Language[t], UML[t], ~~model-driven architecture~~, ~~MDD~~

[74] Unified Modeling Language[t], UML[t], Meta Object Facility[t], MOF[t], domain-specific language[l], DSL[l], profile[l].

[73] ~~model-driven software development~~, ~~MDSD~~, ~~language engineering~~, ~~model-driven development~~, ~~MDD~~, ~~platform-independent language~~, Unified Modeling Language[t], UML[t], metamodel[l], profile[l]

[104] ~~model-driven engineering~~, ~~MDE~~, domain-specifc language[l], DSL[l], code generation[l]

[116] Unified Modeling Language[t], UML[t], ~~UML 2.0~~, ~~light-weight extension~~, meta-model[l], **meta model**[l], **metamodel**[l]

[33] Unified Modeling Language[t], UML[t], meta-level[l], **meta level**[l], **metalevel** profile[l], meta-model[l], **meta model**[l], **metamodel**[l], ~~XML metadata interchange~~, ~~XMI~~

[108] meta-model[l], Unified Modeling Language[t], UML[t], abstract syntax[l], Object Constraint Language[t], OCL[t]

[24] metamodel[l], **meta model**[l], **meta-model**[l]

[61] Meta Object Facility[t], **Meta Object Facility**[t], ~~MOF 2~~, metamodel[t], **meta model**[l], **meta-model**[l], ~~modeling abstraction~~, ~~model view~~

[117] Unified Modeling Language[t], UML[t], visual language[l], object constraint language[t], OCL[t], profile[l]

[19] Unified Modeling Language[t], UML[t], concrete syntax[l], metamodel[l], **meta model**[l], **meta-model**[l], profile[l], ~~model versioning~~

[86] ~~MDD~~, ~~model-driven development~~, extension[l], Unified Modeling Language[t], UML[t], ~~UML 2.0~~

[45] ~~model-driven development~~, domain-specific modeling[l], UML[t] profile[l], ~~platform independent~~

[119] ~~model-driven approach~~, UML[t], ~~model driven development~~, profile[l], Unified Modeling Language[t], profile[l].

[25] ~~model-driven development approach~~, ~~MDD approach~~, ~~MDD philosophy~~, ~~Model Driven Development~~, ~~MDD~~

[118] MOF[t], **Meta Object Facility**[t], metamodel[l], **meta model**[l] , **meta-model**[l], OCL[l], **Object Constraint Language**[t]

[89] Meta Object Facility[t], **MOF**[t], Object Constraint Language[t], **OCL**[t]

[105] profile[l], UML[t], **Unified Modeling Language**[t], specialization[l], refinement[l]

[1] UML[t], **Unified Modeling Language**[t], profile[l]

[63] UML[t], **Unified Modeling Language**[t], profile[l]

[39] ~~model-driven development~~, UML[t], **Unified Modeling Language**[t], profile[l], model-to-model transformation[l], **M2M transformation**[l], model-to-text transformation[l], **M2T transformation**[l], ~~Model-driven engineering~~

[14] profile[l], Unified Modeling Language[t], UML[t]

[132] Unified Modeling Language[t], UML[t], profile[l], meta-model[l], **meta model**[t], **metamodel**[t], conceptual model[l]

---

[32] ~~Unified Modeling Language 2~~, ~~UML 2~~, profile[l], ~~model driven architecture~~, ~~MDA~~

[131] ~~Unified Modeling Language 2~~, ~~UML 2~~, profile[l], ~~software modeling language~~

[27] metamodel[l], **meta-model**[l], **meta model**[l],~~model-driven architecture~~, ~~MDA~~, ~~platform-specific~~, ~~platform-independent~~, domain model[l], Meta Object Facility[t], MOF[t], ~~model-driven development~~, ~~MDD~~, model transformation[l]

[125] ~~domain-specific graphical modeling~~, meta-model[l], metamodel[l], **meta model**[l], domain-specifc modeling [l], DSM [l], Unified Modeling Language[t], UML [l], ~~UML2.0~~, ~~model-driven architecture~~, ~~MDA~~

[130] Unified Modeling Language[t], UML[t]

[30] ~~Object Management Group~~, ~~OMG~~, Unified Modeling Language[t], UML[t], meta-model[l], **meta-model**[l], **meta model**[l]

[59] : Systems Modeling Language[t], SysML[t], Unified Modeling Language[t], UML[t], profile[l]

[10] Systems Modeling Language[t], SysML[t], domain-specifc language[l], DSL[l], Unified Modeling Language[t], UML[t], profile[l]

[126] ~~UML 2~~, extension[l], **Unified Modeling Language**[t], UML[t], **Object Constraint Language**[t], OCL[t]

[13] ~~model-driven approach~~, ~~model construction~~, extension[l], profile[l], model-to-model transformation[l], **M2M transformation**[l], model generation[l], model transformation[l], UML[t], **Unified Modeling Language**[t]

[46] UML[t], **Unified Modeling Language**[t], profile[l], tool[d]

[56] UML[t], **Unified Modeling Language**[t], profile [l], OCL[t], **Object Constraint Language**[l], ~~constraint~~, ~~model-based development~~

[12] UML[t], profile[l], ~~UML 2.0~~, Unified Modeling Language[d], ~~domain specific modeling language~~, tool[d], ~~generation~~

Terms from step 2: The research questions yield the following search terms (as identified by *underlining* the atomic and non-atomic words in the research questions 1–5): domain-specific modeling[l], DSM[l], design decision[d], design-decision making[d], decision option[d], decision driver[d], Meta Object Facility[t], MOF[t], Unified Modeling Language [t], UML[t]

After merging the two subsets, data cleansing is performed:

1. Removing duplicates
2. Reducing plural to singular forms
3. Appending the extended writing for acronyms and vice versa
4. Adding established synonyms or spellings (e.g. dashed or whitespace-separated spellings of composite nouns), as found in secondary studies on DSML development
5. Adding alternative noun forms for gerunds
6. Turning uppercase into lowercase writing
7. Adding alternative BE/AE spellings of a word

Compared to the original search procedure (see Section 4.0), we added the two cleansing steps 4 and 7 because the pilot search has shown that these are critical to obtain relevant hits.

For steps 3 and 4, we consult the previously identified secondary studies on DSL and DSML development (e.g., [20, 75, 80, 87, 100, 115]).

Note that in contrast to the original procedure (see Section 4.0), we do not derive search terms from secondary studies directly. We rather use them for the above cleansing operations. This is due to the decision to drop the third search clause on design documentation (see above). Besides, by the time of the authoritative second search, a fifth secondary study was added for consideration (i.e., [75]).

We group the above terms into a disjunctive-composite form as documented in Section 4.0. Terms denoting a relevant modeling technology (technology^t) form one clause, a relevant notion of modeling language engineering (language engineering^l) a second clause. Terms within a clause are linked by XOR connectives (i.e., exclusive-or). The two clauses are joined by AND connectives.

Listing 2: The QGS search string used for the automated search.

```
(OCL or UML or object constraint language or meta object facility
or MOF or unified modeling language or SysML or systems modeling
language)
and
(customization or customisation or metamodel or meta-model or meta
model or stereotype or profile or domain-specific language or DSL
or code generation or meta-level or metalevel or meta level or
extension or abstract syntax or visual language or concrete syntax
or domain-specific modeling or DSM or specialization or
specialisation or refinement or model-to-model transformation or
model-to-text transformation or M2T transformation or M2M
transformation or conceptual model or domain model or model
transformation or model generation)
```

The resulting, complex search string comprised two clauses (technology, language) grouping 38 atomic and composite search terms. When expanded from its disjunctive-composite form, the search string yielded 240 unique search pairs. Each pair consists of two terms, one out of each clause.

We continuously evaluate the performance of the engine-based search by computing the quasi-sensitivity of the conducted search over all selected engines (i.e. their respective result sets). Following [129], the quasi-sensitivity is defined as the ratio of relevant papers identified by the QGS retrieved through the automated search to the total number of relevant papers (i.e., the size of the QGS corpus). The objective is to reach a quasi-sensitivity level between 70% and 80% (see [129]).

The automated search and its repeated, stepwise sensitivity validation via the QGS corpus allowed for unanticipated changes to the first version of the search (see Listing 2) to maximize the QGS coverage of the combined search hits over the four search engines. To expand the search to effectively include a maximum number of QGS publications, we modified the search string in Listing 2 as follows:

- We added three composite search terms to the technology clause ("sysml profile", "uml profile", and "uml2 profile") introducing a one-word overlap with the language-engineering clause ("profile").
- We added term variants containing major version numbers in the technology clause: "uml2", "uml2.0", "uml 2.0", "mof2", "mof 2.0", and "mof2.0".
- We added two term variants ("model to model transformation", "model to text transformation") to the language clause.

Listing 3: The revised search string used for the automated search.

```
(ocl or uml or object constraint language or meta object facility or
mof or unified modeling language or sysml or systems modeling language
or sysml profile or uml profile or uml2 profile or uml2 or uml2.0 or
uml 2.0 or mof2 or mof 2.0 or mof2.0)
and
(customisation or customization or metamodel or meta-model or meta
model or stereotype or profile or domain-specific language or DSL or
code generation or meta-level or metalevel or meta level or extension
or abstract syntax or visual language or concrete syntax or
domain-specific modeling or DSM or specialization or specialisation or
refinement or model-to-model transformation or model to model
transformation or model-to-text transformation or model to text
transformation or M2T transformation or M2M transformation or
conceptual model or domain model or model transformation or model
generation)
```

The resulting, complex search string comprised two clauses (technology, language) grouping 49 atomic and composite search terms. When expanded from its disjunctive-composite form, the search string yielded 544 unique search pairs. Each pair consists of two terms, one from each clause.

Using the above search string (and engine-specific extensions), we maximized the quasi-sensitivity in our search to an acceptable level of ≈75.7% [129]; that is, 28 out of the 37 QGS publications are contained by the set of 4,695 total search hits. See also Sections 4.3 and 4.6.

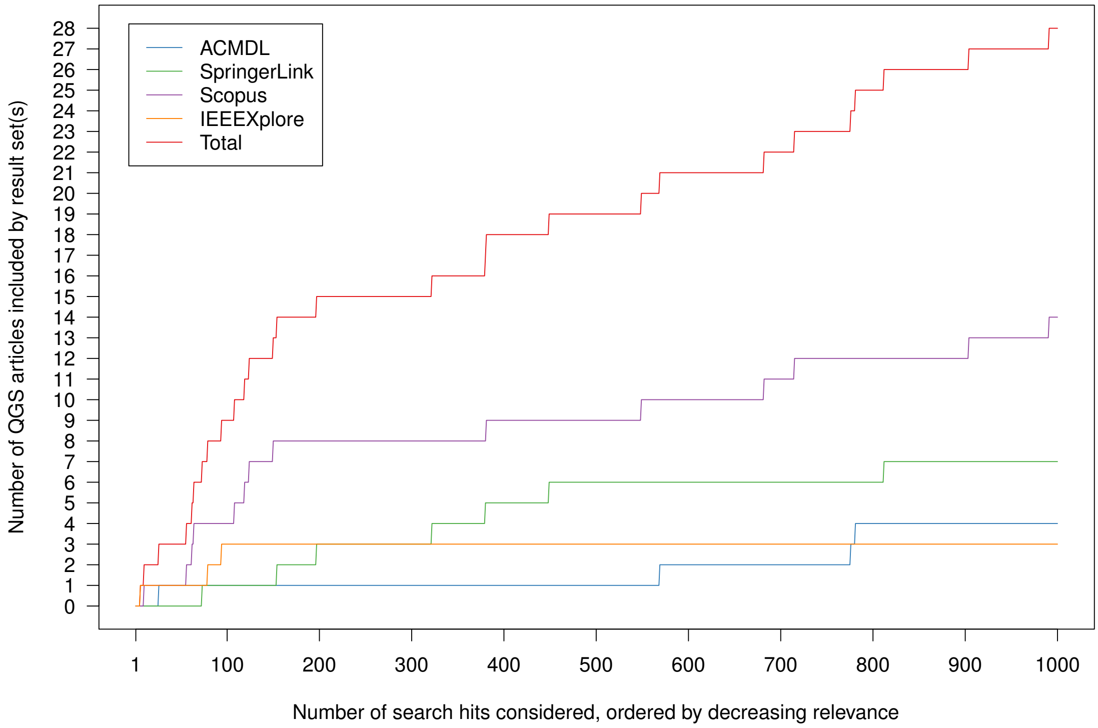


Figure 4: QGS articles found in search results per search engine/method.

Figure 4 illustrates the cumulative numbers of QGS publications found within each of the four relevance-ordered result sets yielded by the main search. On the x-axis, the hit ranks in each result set are shown. On the y-axis, the cumulative numbers of QGS publications found up to a certain hit rank are printed. In total, 15 of the 28 QGS publications (≈54%) were found when considering search hits up to the 200th one in each result set, 21 out of 28 (75%) were ranked up the 600th. In the lowest hit range up to rank 200, a QGS publication was therefore found every ≈13th publication, in average. In upper ranges, the hit probability decreased to every ≈62th publication and below, above rank 200. We capped each result set at the rank of the last QGS publication found in the respective result set: ACMDL at position 781, IEEEXplore at 94, Scopus at 991, and SpringerLink

Table 7: QGS articles found in different range steps of result hits (by increments of 275 hits per step), following the relevance ordering of the result sets (for the main search hits) and the order of manual review for the snowballing candidates.

Result range	ACMDL	SpringerLink	Scopus	IEEEXplore	Snowballing	Total	Recall
1–275	1	3	8	3	0	15 (52%)	.405
276–550	1	6	10	3	0	20 (69%)	.541
551–825	4	7	12	3	0	26 (90%)	.703
826–1,100	4	7	14	3	0	28 (97%)	.757
1,101–1,375	4	7	14	3	0	28 (97%)	.757
1,376–1,650	4	7	14	3	0	28 (97%)	.757
1,651–1,925	4	7	14	3	0	28 (97%)	.757
1,926–2,200	4	7	14	3	1	29 (100%)	.784
2,201–2,337	4	7	14	3	1	29 (100%)	.784

at 812. This way, with a total number of 2,678 reviewed search hits, we incorporated all retrieved QGS publications (28) while limiting an otherwise excessive review workload.

Table 7 lists the cumulative numbers of QGS publications found within each of the four relevance-ordered result sets yielded by the main search and adds the results from the citation-based search (snowballing). Via the snowballing search, we found one more QGS publication, arriving at a total number of 29.

4.3 Engine-based Search

Prior to revision 3 of this protocol, and the adoption of a QGS-based approach in more recent revisions, we planned to use a different set of search engines (e.g., Google Scholar, ScienceDirect) and the way of documenting the engine-based search. Please see [36] for details on the search as planned prior to revision 3.

4.4 Assessing Engine Capabilities

The search procedure using the search engines that were selected in the previous step (see Section 4.1) will be prepared as follows. For each search engine, we will first establish its capabilities:

(1) Search site: `URL`

(2) Support for complex queries (i.e., nested, boolean query expressions and metadata fields)? `yes`/`no`

(3) Indexes full text of publications (rather than providing access to full text)? `yes`/`no`

(4) Indexes abstract of publications (i.e., queries are evaluated also against abstract texts and the search can be restricted to abstracts)? `yes`/`no`

(5) Indexes title of publications (i.e., queries are evaluated also against title texts and the search can be restricted to titles)? `yes`/`no`

(6) Indexes author-provided keywords of publications (i.e., queries are evaluated also against keywords and the search can be restricted to keywords)? `yes`/`no`

(7) Contains publications in languages other than English (i.e., the full-text language is not English)? `yes`/`no`

(8) Allows for restricting the search to ranges of publication years? `yes`/`no`

(9) Allows for restricting the search to venue types (e.g., conferences, journals)? `yes`/`no`

(10) Are there per-query restrictions (e.g., capped result sets, limited number of expression components etc.)? `yes`/`no`

(11) Allows for controlling the ordering of the result set (e.g., sort by relevance, sort by publication date)? `yes`/`no`

(12) Allows for controlling the output format of result sets? `yes`/`no`

(13) Allows for exporting the references lists of individual search hits/ publications? `yes`/`no`

Table 8: Overview of capabilities and restrictions of each search engine.

	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
ACMDL	yes	yes	yes	yes	yes	no	yes	yes	yes (paging, results not exportable)	yes	partly yes (search hits as markup, single bibliographical records in various formats)	no
IEEEXplore	yes	yes	yes	yes	yes	no	yes	yes	yes (15 search terms, export of 2,000 results max.)	yes	yes (csv, markup)	no
Scopus	yes	no	yes	yes	yes	yes	yes	yes	yes (download of 2,000 results max. incl. keywords/abstracts)	yes	yes (e.g., csv, Mendeley, RefWorks, BibTeX)	no
SpringerLink	yes	yes	no	yes	no	yes	yes	yes	yes (export of 1,000 results max.)	yes	yes (csv, markup)	no

We consulted the following search sites (1) of the four selected engines:

- ACMDL *Advanced Search*:
  `http://dl.acm.org/advsearch.cfm`
- IEEEXplore *Command Search*:
  `http://ieeexplore.ieee.org/search/advsearch.jsp?expression-builder`
- Scopus *Advanced Search*:
  `https://www.scopus.com/search/form.url?display=advanced`
  (Assuming an existing, institutionalized access to Scopus.)
- SpringerLink *Advanced Search*:
  `http://link.springer.com/advanced-search`

The capabilities and restrictions are summarized in Table 8. The key findings were:

- Ad (2): All four search engines allowed for expressing and for processing complex, nested search expressions using basic boolean operators. Therefore, we could run single, manual searches for IEEEXplore, Scopus, SpringerLink. ACMDL was an exception, due to restrictions of the result presentation. However, even for ACMDL, there was no need for transforming the generic search string (e.g., into a series of atomic, pairwise sub-queries).
- Ad (3): Three of the four engines operate on full-text collections (ACMDL, IEEEXplore, and SpringerLink), Scopus is a pure bibliographical search engine. To this end, the evaluation of the search strings was expected to be performed on different grounds. Therefore, and as the objective was to have the engine-based search evaluate primarily against title, abstract, and keywords, we adapted the implementations of the generic search string specific to the full-text engines accordingly, if supported; see Items (4)–(6).
- Ad (4)–(6): Having the objective to assess titles, abstracts, and keywords, and as a result of the different nature of three out of four search engines (see Item 3), we aimed at turning the generic search string into engine-specific strings which are evaluated against these three metadata fields of each stored publication. All four engines provide access to the publication titles. Abtracts metadata is accessible for all except for SpringerLink. Similarly, author keywords are available in three engines (Scopus, IEEEXplore, and ACMDL); SpringerLink does not allow to elicit the search implementation in this respect. Note that accessibility (in our definition) implies having control over this metadata. SpringerLink may certainly include abstract and author keywords in search evaluation, but this behavior cannot be controlled by the reviewer.

  The four engines differed in the way a search that has been restricted to the metadata fields needed to be expressed: ACMDL offers metadata-restrictions using a web form or metadata prefixes in the boolean query expression. Due to the need of automating the ACM search (see Item 12), we adopted the prefix annotation of the search-string implementation. IEEEXplore *Command Search* offers a global switch ("Metadata only") or prefixes to prepended to the search terms (e.g., `Abstract:`*`term`*). We opted for the switch-based restriction, therefore avoiding introducing redundant search terms due to the documented query restrictions (see Item 10). In Scopus, the defaults in the advanced search mode imply a restriction to title, abstract, and keywords (note that keywords include also publisher-provided keywords). SpringerLink did not allow for limiting the search to all of the three metadata entries. Thus, for SpringerLink we could not control the results in this respect.

- Ad (7): While the ACMDL and IEEEXplore maintain only a corpus of publications having an English full-text body, Scopus and SpringerLink may also return non-English publications (e.g., Scopus just requires an additional English abstract). Therefore, we added language constraints to the query implementations of the latter two. Scopus provides a dedicated metadata constraint (i.e., a field code), SpringerLink provides a corresponding filter control.

- Ad (8): All four search engines allowed us to query the years of publication (2005–2012); the query implementations were extended accordingly. Except for Scopus, we also could define start and end years. Scopus required us to add an exclusive-or clause of eight metadata terms (one per year). Running the search early in 2013 (end of January), we actually set the end years to 2013 to cover more recent additions to the searched publication collections.

- Ad (9): The objective is to obtain search hits which correspond to the relevant venue types, that is archival venues (journals) and presence venues (conferences); see also Section 6. ACMDL allows for limiting the search to journal, proceedings (including workshops) and ACM transactions through the web UI or metadata prefixes (`PublishedAs`). We used the latter due to the automation need (see Item 12). IEEEXplore allows for filtering the initial search hits for "Conference Publications" and for "Journals & Magazines" via the web UI. Scopus, again, provides metadata annotations (field code `DOCTYPE`) to restrict a search to those venue types (i.e., `ar` for journal articles, `cp` for conference proceedings). SpringerLink allows for filtering different publication types (e.g., book chapters, articles), however, these Springer-specific categories did not match with our categorization (e.g., many conference articles are listed as chapters in LNCS proceedings, while others are not). Therefore, we did not apply this filter.

- Ad (10): The four search engines presented very unique challenges resulting from various per-query restrictions, some constraining the query itself, others constraining the result set which was extractable.

  None of the four engines imposed any *actual* restrictions on the query length (e.g., clauses, terms) or on the nesting depth which would apply to our generic search string. While IEEEXplore *Command Search* announces a maximum of 15 search terms, our generic search string (which counted more than 15 search terms being connected by boolean operators) was accepted and processed.

  Regarding result-set extraction, we encountered important limitations, however. ACMDL does not impose a general capping, but the presentation of the result sets as a paged collection of mark-up documents required custom application support to retrieve the complete result set. As a result, we could not run a single, manual query for ACMDL. We rather had to iterate the paging ranges running individual, batched HTTP requests. This was because of restrictions on the result presentation (ACMDL) and/or restrictions on the size of the query expressions (ACMDL, SpringerLink).

  Scopus and SpringerLink allowed only 2,000 and 1,000 search hits, respectively, to be stashed away in a processable representation including the metadata (in particular abstracts, keywords) necessary for the selection decisions. Likewise, IEEEXplore has a download cap of 2,000 search hits.

- Ad (11): All four engines provided control over ordering of the results, depending on various criteria (e.g., by publication year, by author, by relevance, by recency). When running the searches, we used relevance-based orderings applied via the web UIs or by parametrizing the HTTP requests accordingly. We considered relevance-based ordering important given the ruling caps imposed on exporting result sets from some search engines (see Item 10).

- Ad (12): To some extent, all four engines provided at least two alternative representations of the search hits: a markup based (e.g., HTML, XHTML) or a non-markup, yet structured textual representation, such as, comma-separated value lists (csv). For IEEEXplore, Scopus, and SpringerLink we opted for csv, mainly because it was the only non-markup option available (IEEEXplore, Scopus) and because this representation can easily be converted in a spreadsheet (for the review and selection phase). The ACMDL was more challenging. First, there was no built-in markup-based export facility available to persist the result set. Second, the result set was presented as paged markup. Therefore, for ACMDL only, we automated the search to some extent: Initially, we performed a direct HTTP request to obtain the paged markup to extract the ACM-specific publication identifier from the markup. Then, for each identifier, we ran another HTTP request to obtain the bibliographical record (i.e., as a BibTeX entry) corresponding to every identifier.

- Ad (13): None of the four search engines and the respective digital libraries or portals allowed for exporting the references lists of individual search hits and publications (e.g., into a processable format, along with the main publication record). For the step of backward snowballing (see Section 5), we therefore manually extracted reference lists from the markup records of the individual papers (which were found during the main search) into a spreadsheet document for the subsequent manual review.

4.5 Performing Engine Searches

Depending on a search engine's capabilities, the generic search string (see Section 4) will be converted into an engine-specific variant. For instance, if complex queries are not supported, the generic search string will be expanded into simple, flat variants (e.g., term tuples assuming an implicit conjunction between them). If time-range restrictions and constraints over metadata fields (e.g., publication language) can be expressed, the generic search string will be extended to represent as many search constraints as possible. The resulting, engine-specific search strings will be documented as follows:

- Exact search string(s);

  The exact search strings for the four search engines are listed in the Appendix; see Section B.

- Date of running the search(es);

  - ACMDL: Feb 1, 2013 (both, retrieving search and extracting ACM identifiers as well as retrieving bibliographical record for each identifier);
  - IEEEXplore: Feb 1, 2013;
  - Scopus: Feb 1, 2013;
  - SpringerLink: Feb 1, 2013;

- Number of results;

  In total, we obtained a result set of 5,778 publications from the four search engines (see also Table 12):

  - ACMDL: 933 (uncapped exports, equals total hits);
  - IEEEXplore: 1,845 (capping at 2,000 did not apply; total hits: 1,845);
  - Scopus: 2,000 (capped results; total hits: 16,579);
  - SpringerLink: 1,000 (capped exports; total hits: 8,526);

- Modifications or extensions to the basic search string, required by a search-engine's capabilities (see above) or resulting from the stepwise sensitivity-based refinement (e.g., added terms to have a QGS publication enter the result set; see also Section 4.2);

  We modified the engine-specific implementations as follows:

  - ACMDL
    * Adding HTTP query parameter `since_year=2005`
    * Adding HTTP query parameter `before_year=2013`
    * Adding a clause of three search terms prefixed with `PublishedAs` to restrict publication types (see Item 9): `(PublishedAs:journal OR PublishedAs:proceeding OR PublishedAs:transaction)`
    * Adding a metadata-only clause to constrain the search to the targeted technology context (UML, MOF, SysML; see Items 4–6): `((Keywords:UML OR Keywords:"unified modeling language" OR Keywords:MOF OR Keywords:"meta object facility" OR Keywords:SysML OR Keywords:"systems modeling language") OR (Abstract:UML OR Abstract:"unified modeling language" OR Abstract:MOF OR Abstract:"meta object facility" OR Abstract:SysML OR Abstract:"systems modeling language"))`

- IEEEXplore:
  - (Command search page) Checking `Metadata only`;
  - (Result page) Setting `Publication Year`: 2005–2013;
  - (Result page) Setting `Content Type`: `Conference Publications`, `Journals & Magazines`;
- Scopus
  - Adding a clause of nine metadata terms `LIMIT-TO(PUBYEAR, yyyy)` to limit the search to the review years: `(LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR LIMIT-TO(PUBYEAR, 2009) OR LIMIT-TO(PUBYEAR, 2008) OR LIMIT-TO(PUBYEAR, 2007) OR LIMIT-TO(PUBYEAR, 2006) OR LIMIT-TO(PUBYEAR, 2005));`
  - Adding a clause of two metadata terms `LIMIT-TO(DOCTYPE,value)`: `(LIMIT-TO(DOCTYPE, "cp") OR LIMIT-TO(DOCTYPE, "ar"));`
  - Adding a clause of one metadata term `LIMIT-TO(LANGUAGE, value)` to restrict result to English full-text publications (see Item 7): `LIMIT-TO(LANGUAGE, "English");`
- SpringerLink
  - (Result page) Setting `Language` to `English` (see Item 7);
  - (Result page) Setting `Discipline` to `Computer Science`;
  - (Result page) Setting `Subdiscipline` to `SWE`;
  - (Result page) Setting `Date published` to 2005–2013;

- Specific conditions for each search (e.g., custom automation support);

  The search in ACMDL was scripted, based on low-level HTTP requests, to tackle the paged presentation of the result set and to retrieve the bibliographical records. For the remaining three engines, we instrumented the browser and web UI of the respective engines.

4.6 Post-processing Result Sets

Automated metadata processing. Based on our experience from our pilot study, we run the following cleansing operations on the bibliographical metadata:

- Selecting a metadata subset of each result set for further review, in particular for the selection decisions as defined in Section 6. We expect to find the following metadata entries to be extractable automatically: Unique identifier (DOI), Authors, Title, Publication Year, Page Range, URL (see also Section 7).

  The correspondences we used for extracting key metadata fields from the four resulting data sets as required by Section 7, are documented in Table 9. Metadata missing from some result sets (namely: Abstract, Keywords) was accessed by the reviewers by consulting the external resources (as identifiable by the DOI or the URL), rather than replicating these metadata.

Table 9: Correspondences between key metadata fields in the result sets, as retrieved by the four search engines; n/a: not available in result set.

	ACMDL	IEEEXplore	Scopus	SpringerLink
DOI	DOI	DOI	DOI	Item.DOI
Authors	Author	Authors	Authors	Authors
Title	Title	Document.Title	Title	Publication.Title
Publication Year	Year	Publication.Year	Year	Publication.Year
Page Range	Pages	Start.Page, End.Page	Page.start, Page.end	n/a
URL	URL	PDF.Link	Link	URL
Publisher	Publisher	Publisher	Publisher	n/a (Springer only)
Abstract	n/a	Abstract	Abstract	n/a
Keywords	n/a	Author.Keywords	Author.Keywords	n/a
Engine identifier	Identifier	PDF.Link (`arnumber` query parameter)	Link (eid query parameter)	URL

- Adding missing Document Object Identifiers (DOIs), e.g. by extracting them from other metadata fields such as URLs. For those that cannot be established without manual intervention by the authors, a temporary auto-generated identifier will be added.

Automated duplicate marking. We mark duplicate publications (duplicates, hereafter) in several incremental steps. Note that a duplicate is the repeated occurrence of exactly the same publication within one (intra-source duplicate) or between two or more result sets (inter-source duplicate).

First, we will assess the DOI coverage (i.e. the number of search hits having a DOI relative to the total number of hits) in each result set. Based on the DOI coverage, we will then decide how to proceed in detecting intra-source and inter-source duplicates.

From 5,778 search hits, 5,072 or 87.8% provided a DOI. 706 hits or 12.2% lacked a DOI. SpringerLink provided full DOI coverage, Scopus the lowest DOI coverage with only 29.5% (456/1,544) of its hits having a DOI (see Table 12).

In case of missing DOIs, detecting intra-source duplicates will be tackled using engine-provided identifiers.

For this purpose, we used four metadata fields documented in Table 9.

As for missing DOIs for detecting inter-source duplicates, we will resort to a matching strategy using the publication titles plus manual reviews:

1. By matching publication titles literally and exactly.
2. By computing the pairwise Jaccard similarity [76] of publication titles:
   - We first tokenize and process all publication titles into a unique, canonical form: stopwords removal, punctuation removal, stemming. For this purpose, we will use the standard transformations available in the R package `tm` [34].
   - Publications of titles having a Jaccard similarity of exactly 1 will be manually reviewed.

Table 10: Overview of engine/engine overlap in terms of duplicated DOIs. Each cell of the lower segment represents the absolute number of DOIs contained at least once in a pair of two result sets. The upper cells contain the ratios of duplicated DOIs between two result sets and the total number of non-unique hits between in the two result sets. The diagonale represents the summed number of DOIs, each appearing at least in one or more result sets, contained by one result set.

	ACMDL	IEEEXplore	Scopus	SpringerLink
ACMDL	**226**	1.5%	6.3%	2.7%
IEEEXplore	42	**413**	10%	0%
Scopus	184	385	**959**	14.8%
SpringerLink	53	0	443	**457**

Table 11: Overview of engine/engine overlap in terms of duplicated publication titles (exact and validated Jaccard matches, no DOIs available). Each cell of the lower segment represents the absolute number of non-unique titles contained at least once in a pair of two result sets. The upper cells contain the ratios of non-unique titles in two result sets and the total number of non-unique hits in the two result sets. The diagonale represents the total number of non-unique titles, each appearing at least in one or more result sets, contained by one result set.

	ACMDL	IEEEXplore	Scopus	SpringerLink
ACMDL	**16**	0.2%	0.4%	0%
IEEEXplore	6	**15**	0.3%	0%
Scopus	11	10	**20**	0%
SpringerLink	0	0	0	**0**

The findings of the duplication-detection step were the following (see Table 12):

- *DOI coverage*: By propagating DOIs between duplicated hits in different data sets, we improved the overall DOI coverage from 87.8% to 90.6% (or from 5,072 to 5,232 search hits out of 5,778).
- *Intra-source duplicates*
  - ACMDL, IEEEXplore and SpringerLink did not contain duplicates identifiable in this step, regardless of the detection strategy used.
  - For Scopus, we found ten intra-source duplicates in total; seven using the DOIs, three

additional based on title matching.

- *Inter-source duplicates*
  - In total, we found 1,001 unique DOIs with at least one occurrence in at least two result sets: 948 re-occurred in two result sets, 53 occurred in three result sets.
  - Nearly half of the DOIs retrieved from Scopus and SpringerLink were not unique to these two sets: Scopus contained 959 non-unique DOIs (present in at least one other result set) or 48% (959/2,000); SpringerLink 47.5% (457/1,000; see Tables 10 and 12).
  - Considering pairs of result sets: The maximum DOI overlap of 14.8% was found between Scopus and SpringerLink, followed by 6.3% between Scopus and the ACMDL (see Tables 10).
  - From the remaining hits which do not contain a DOI, we found 25 non-unique titles over all four data sets.
  - Scopus contained the maximum of 20 publications having no DOI and a non-unique title.
  - Scopus contributed to the maximum pairwise overlap observed, e.g., between Scopus and ACMDL as well as between Scopus and IEEEXplore (see Tables 11).
  - In addition to exact title matching, title matching based on Jaccard similarity yielded another eight duplicate candidates. Manual review confirmed that six from these are actual duplicates.

Table 12: Results of the different duplicate-marking steps; intra-source: within the result of one engine; inter-source: between the result sets of the four engines.

	ACMDL	IEEEXplore	Scopus	SpringerLink	Σ
# search hits	933	1,845	2,000	1,000	5,778
# duplicated hits	80	132	747	124	1,083
# cleaned hits	853	1,713	1,253	876	4,695
before DOI cleansing					
# hits without DOI	180	70	456	0	706
# hits with DOI	753	1,775	1,544	1,000	5,072
after DOI cleansing					
# hits without DOI	149	66	331	0	546
# hits with DOI	784	1,779	1,669	1,000	5,232
intra-source duplicates					
# based on DOI	0	0	7	0	7
# based on title match	0	0	3	0	3
# based on engine identifier	0	0	0	0	0
inter-source duplicates					
# based on DOI	226	413	959	457	n/a
# based on title match	16	15	20	0	n/a

The removal procedure for duplicates was performed in the following way:

- The objective is to preserve the engine-based relevance ordering. That is, maintain those (duplicated) occurrences which rank higher in the relevance-ordered result sets.
- In turn, removing only the relatively lower-ranked (duplicated) occurrences makes non-duplicated search hits become positioned higher in the ordered, cleansed result sets.

- *Overall duplicate removal*: Based on these findings, we removed 1,083 duplicated hits or 18.7% of the total search hits over all four result sets. The authoritative combined data set for the actual selection step therefore contained 4,695 publications.
- In relative terms, we found, verified, and removed the largest number of duplicates from Scopus: 774 duplicates or 38.7% (774/2,000). Scopus is followed by SpringerLink with 12.4% of removed duplicates (124/1,000), ACMDL with 9.6% (80/933), and IEEEXplore with 7.2% (132/1,845).
- This finding matched our initial intuition concerning the expected overlap between Scopus and the other three engines, stated earlier (see Section 4.1).

- Limitations of the duplicate detection procedure:
  - *Metadata quality*: We encountered a number of metadata defects which limit the effectiveness of either duplicate detection strategy (DOI, title matching, engine identifier):
    * Mistyped or inconsistent DOIs: We found at least one DOI which was mistyped in two sources (containing a dash rather than an underscore). Due to this sort of inconsistency, the DOI-based detection strategy might have missed actual duplicates. In one case, we found two different DOIs pointing to one and the same publication.
    * Title matching was certainly affected by character-set encoding issues, turning special characters into a different encoding or encoding representation. We encountered five different cases for different dash and wildcard characters. The Jaccard similarity check helped us to find these cases though.
    * The engine-specific identifiers did not help detect duplicates not already properly sorted out be the search engine. For example, the intra-source duplicates in Scopus could not be detected in this way.
  - *Title ambiguity*: Title matching (whether in an exact or similarity-based manner) is only a weak indicator because of the possibility of title sharing between follow-up publications of the same authors or similar corner cases. Manual review was essential, but caused a considerable effort.
- As a result, we manually identified another 33 duplicates in the collection of 4,695 hits during the selection procedure.

Sensitivity analysis. The quasi-sensitivity is defined as the ratio of relevant papers identified by the QGS retrieved through the automated search to the number of total number of relevant papers (i.e. the size of the QGS corpus; [129]). The objective is to reach a quasi-sensitivity level between 70% and 80% [129].

Based on the unprocessed and the cleansed result sets, we will compute the following statistics related to the analysis of quasi-sensitivity:

- Quasi-sensitivity ratio per engine/result set: The number of QGS publications expected to be covered by a search engine (given its publisher; see Table 5) relative to the number of these QGS publications actually retrieved from this engine.

  The raw data is summarized in Table 13. columns 1 (expected) and 2 (intra-source):

  - The result set retrieved from ACMDL contained all of the two expected QGS publications (quasi-sensitivity ratio of 1).
  - The result set retrieved from IEEEXplore contained three out of seven expected QGS publications (quasi-sensitivity ratio of ≈0.43).
  - The result set retrieved from Scopus contained seven out of 13 expected QGS publications (quasi-sensitivity ratio of ≈0.54).
  - The result set retrieved from SpringerLink contained seven out of 15 expected QGS publications (quasi-sensitivity ratio of ≈0.47).

Table 13: Overview of per-engine quasi-sensitivity for the four search engines and the respective result sets.

	expected	intra-source	extra-source	inter-source
ACMDL	2	2	1	2
IEEEXplore	7	3	4	5
Scopus	13	7	0	7
SpringerLink	15	7	13	14

- Quasi-sensitivity ratio across all engines/result sets: The number of QGS publications retrieved from all selected search engines relative to the total number of QGS publications (see Section 4.1).

  - The total, unique search hits retrieved from all four search engines (4,695) contained 28 (i.e., sum of column 4, inter-source, of Table 13), out of 37 QGS publications. This yields a quasi-sensitivity ratio of ≈0.76.

- More than 70% of the expected QGS publications linked to ACMDL, IEEEXplore, and SpringerLink could be found either in the respective engine itself or any of the other engines (inter-source).
- The lowest inter-source sensitivity was obtained for Scopus: Only seven out of 13 expected QGS publications (≈0.54) could be found in Scopus itself and the three other engines.

- Overlap: Number of QGS publications co-occurring in the different result sets.

The raw data are depicted in Table 14:

- There was no overlap in terms of QGS publications between IEEEXplore and ACMDL, on the one hand, as well as between SpringerLink and IEEEXplore.
- The maximum QGS overlap was observed between SpringerLink and Scopus with nearly half (42.9%) of the total expected QGS publications (28; see Table 13 co-occurring between the two.
- The result set of Scopus was involved in most co-occurrences (25); see Table 14.

Table 14: Overview of co-occurring QGS publications between the four search engines. Each cell of the lower segment represents the absolute number of QGS publications co-occurring once in a pair of two result sets. The upper cells contain the ratios of QGS publications co-occurring in two result sets and the total number of expected QGS publications over the two result sets. The diagonale represents the summed number of co-occurring QGS publications, each appearing once in the given result set and the other result sets.

	ACMDL	IEEEXplore	Scopus	SpringerLink
ACMDL	6	0%	33.3%	17.6%
IEEEXplore	0	4	15%	0%
Scopus	5	3	25	42.9%
SpringerLink	3	0	12	13

5 Snowballing Strategy

To ensure that we obtain a paper corpus which represents a preferably complete selection of relevant literature on UML-based DSML designs, we will run a citation-based search. In particular, we adopted the steps described by [60, 120] to establish the following snowballing guideline:

1. *Start set*: Initially, a set of papers to extract the references must be identified by the reviewers. This set must contain only papers which also satisfy the selection criteria (see Section 6) to which candidate papers retrieved by snowballing are subject during evaluation. In addition, any authors' bias should be minimized when establishing the start set. Therefore, this start set will be formed by the finally included papers from the main search step (see Section 4.3).

2. *Iterations*: We are interested in prior work on DSML designs which is considered relevant and which is therefore referenced by authors of the start set (*backward snowballing*). The initial iteration will be run based on the start set (see above). Subsequent iterations will be performed on papers selected during the prior iteration. This will be continued till no further paper is included. The following sub-steps are performed repeatedly, for each iteration.

3. *References extraction*: To extract the references from the start-set publications, we will consider the capabilities, if any, for retrieving references of papers from the respective digital libraries that we used to run the engine-based search (see Section 4.4). In any case, and for the follow-up iterations based on third-party sources, we will manually extract the references into a spreadsheet document. The minimum entries per reference record are (see also Section 7):

   - Publication title
   - Publication venue
   - Authors list

4. *Postprocessing*: For each iteration, the candidate papers extracted initially will be processed to mark duplicates, using the procedure (if applicable) as outlined in Section 4.6.

The postprocessing sub-step could not be realized because of the lacking and heterogeneous nature of the reference records. We resorted to manual duplicate detection.

5. *Selection*: The postprocessed references will be evaluated against the selection criteria defined in Section 6. The selection tasks will be split between the authors, so that each author reviews an equally sized subset of candidate papers. A second reviewer will check a 20% random sample drawn from the subset reviewed by another reviewer, by making the selection decision independently from the first reviewer (see also Section 6.4.3).

6. *Quality Assessment*: The candidate papers for inclusion will be further assessed according to the procedure defined in Section 6.3.

7. *Data extraction*: The included papers will be subjected to the decision-data extraction procedure defined in Section 7. These papers form the start set for the subsequent iteration (see above).

6 Selection Criteria

6.1 Inclusion and Exclusion

For evaluating a publication for inclusion or exclusion, we distinguish between venue-specific and publication-specific criteria. The former are to be evaluated first (as a venue-based decision may apply to several publications under evaluation), the latter are evaluated afterwards and only iff the venue-specific criteria are met by the publication. Within each set of criteria, the order of evaluation is based on experiences drawn from the pilot study. Criteria are considered earlier if the evaluation has the potential of being reused for multiple publications (e.g., venue-specific ones) and/or the criteria can be checked in an assisted and guided manner (e.g., by verifying an item against a check list). Criteria requiring an in-depth analysis of publication content are positioned at the end of the evaluation orders.

The venue-specific criteria are (in their order of evaluation):

C_1 *Time coverage*: The publication venue must cover a time period between 2005–2012, that is, the venue must have a regular record of publishing in this time span. The start year follows from the release of UML 2.0 specification in July 2005. Work published until Jan 30, 2013 is included in the systematic review. For presence venues, evaluating this criterion is inherently linked to evaluating C_2 (see below).

C_2 *Community relevance*: We want to include publication venues considered relevant by sources beyond the venue's native community and the authors' judgment alone.

- For our SLR, a *journal* must be listed with the *Excellence in Research for Australia Journal List 2012* (ERA [7]) as one of the 304 journals categorized into *Computer Systems* (803) and/or *Information Systems* (806). We adopt this journal list because it results from a public and international consultation of the various scientific communities, it does not impose prescriptive journal ranks, it has already been used for systematic reviews, and it has been acknowledged by international research bodies (WKWI[4]). Using the 2012 edition of the ERA journal list also guarantees that journals over our entire review period are potentially covered (2005–2012).
- As for a *conference*, there must be a continuous publishing history. A presence venue (conference, symposium) must have been held at a regular basis during the review period. By regular, we mean repeatedly according to the conference format (yearly, biyearly etc.). For yearly venues, for example, this is equal to eight issues corresponding to the SLR period of 2005–2012. However, if a venue has changed the format within the review period (e.g., from an annual to a biannual scheme) this mark can differ. In doubt about a changing venue's schedule (e.g., because the scheme change has not been announced explicitly or it cannot be induced from reviewing a bibliographical data base), the venue is accepted under this criterion. A regular publication history is considered an indicator of acceptance and support of a venue by its scientific community and by other stakeholders (e.g., publishers, bibliographical data bases such as DBLP, and industry sponsors). Note that for presence venues, applying this criterion equals evaluating C_1.

C_3 *Software-engineering focus*: We include publications having a SE audience because we are interested in DSML design decisions primarily from a SE perspective (e.g., separation of concerns), methodologies (e.g., MDD), as well as SE-specific methods/tools (e.g., model transformations). Therefore, we consider only dedicated SE venues (rather than conferences that aim at a specific application domain, such as health-care for example):

[4] `http://wi.vhbonline.org/index.php?id=104`; last accessed: Feb 2, 2015.

- A *journal* must have a clear and acknowledged focus on SE topics. A journal's SE profile is judged according to the authors' expertise (two of which are senior SE researchers) and by verifying whether the journal is contained in the 1) 53 journals in category *Computer Science*, subcategory *Software Engineering* of Microsoft Academic Search (MS/AS [72]; as of Feb 4, 2013) and/or in the 2) 184 journals in the subject area *Computer Science*, subject category *Software*, of the 2012 *SCImago Journal Rank* list (SJR [99]).

  Both journal lists are publicly available and allow for discriminating between SE venues and other venues independently from each other (i.e., based on different bibliographical data sets and on classifications developed independently).

  We do *not* consider, even if available, any journal ranks in our inclusion/exclusion decision (e.g. the SJR indicator). The only criterion is the condition of being listed with the sources above.
- A *conference* is considered as having an accepted SE focus (in the most inclusive sense) when it is contained in the list of 169 SE conferences maintained by Tao Xie ([123]; as of Feb 4, 2013) and/or in the Microsoft Academic Search list of 284 SE venues (as of Feb 4, 2013). This way, we base this selection decision on two independent sources.

C_4 *Mature content*: A *peer-review procedure* must be in place for a venue as a first prerequisite to identify mature and rigorous research content (i.e., research results including research evaluation). As for presence venues we exclude *workshops*. A presence venue is qualified as a workshop if the proceedings title identifies the venue as such and/or the main, supporting venue (e.g. a conference or symposium) lists the venue as a workshop event (e.g. in its CFP or at its website, if available). As for archival venues, formats other than periodicals (e.g. a monograph such as a festschrift, an article collection, or a research report) are excluded.

The publication-specific criteria are (in order of their evaluation):

C_5 *Full-text accessibility*: The publication's full-text must be accessible to the authors, either via the publisher's digital libraries (assuming suitable subscriptions of the authors and their institutions), the author's Web page or third-party online locations (e.g., Citeseer). Full-text availability is critical for evaluating the publication-specific criteria C_6 through C_{11}.

> From all 4,695 unique search hits (see Section 4.6), 47 publications turned out not to be accessible to the authors and were therefore excluded.

C_6 *Publication language*: The publication full-text must be written in English or, alternatively, there must be an English version available/accessible. It is not sufficient to have (additional) English title, abstract, keywords and other metadata only.

C_7 *Primary study*: We include only a *primary study*, i.e. papers falling into the two following categories according to [122]: 1) Proposal of DSML solution and/or 2) DSML personal experience paper. In particular, a publication will be excluded:

- if it is not related to a DSL or DSML based on the UML (according to title, keywords, and abstract). In case of disagreement on the selection decision, the full content of the paper must be considered;
- if it represents a *secondary study* on DSMLs (e.g., an SLR on DSLs or DSMLs).

C_8 *Publication type*: A candidate publication will be excluded:

- if it was not published as a regular research paper and special-issue research paper in a journal venue. Article types such as editorial papers and columns as well as book chapters and technical reports are excluded;
- if it was not published as a full research paper in a presence venue. Therefore, short and position papers, as well as notes on posters, talks, tutorials, and tool demos are excluded.

C_9 *UML version match*: The publication and the DSML must be based on the UML version 2.0 (implying MOF 2.0) or newer. This is because, in our catalog, we assume the availability of UML 2.x and MOF 2.x extension features. Note that this criterion is also meant to eliminate publications which have been published post 2005 but use earlier UML/MOF revisions (see also C_1).

C_{10} *Originality*: The candidate publication (provided that the publication is accepted as primary study under C_7) must be the original published work documenting a DSML design. Any follow-up, extending, or promoted publications will be excluded (e.g., proceedings papers extended to yield a journal article) based on the publication year (or other date stamps indicating a precedence order between two publications). Nevertheless, subsequent publications, while not being included in the paper corpus initially and reported as part of the SLR, will be consulted for clarifying details on a DSML design (e.g. to clarify deviating judgements between the co-reviewers when codifying the design decisions). Extensions of prior UML-based DSMLs (e.g. by adding a new UML profile or by extending a previously reported metamodel extension) will be recorded as a separate DSML design. In case of doubt, a candidate publication is accepted under this criterion.

C_{11} *Minimum coverage*: The publication documents design details (e.g., abstract syntax, concrete syntax, constraints, behavioral specification) of a DSML or a set of related DSMLs based on the MOF/UML. The match is assessed by reviewing the publication's full-text. Minimum coverage must be the description of the language model definition (D1) and formalization decisions (D2).

6.2 Selection States

During the selection process as documented in Section 6.1, a given publication under review keeps a number of intermediate selection states to finally reach a final decision state (i.e., `excluded` or `included`). The statechart in Figure 5 documents the three main evaluation steps towards a selection decision and the resulting, valid state transitions:

1. If any of the venue-specific criteria (C_1–C_4) evaluates to false, a publication will be `excluded` because it does Not Match the venue-specific Criteria (`NMC`). Likewise, if the publication-specific criteria on the full-text accessibility (C_5) or on the publication language (C_6) do not hold, the publication will also be marked `NMC` and will therefore be `excluded`.
2. If not already `excluded` and if the publication-specific criterion on a topical match (C_7) evaluates to false, a publication is `excluded` because it is considered Off Topic (`OT`).
3. If not already `excluded` and if any of the remaining publication-specific criteria (C_8–C_{11}) evaluates to false, a publication is `excluded` because it is considered Not Matching these publication-specific Criteria (i.e., again `NMC`).

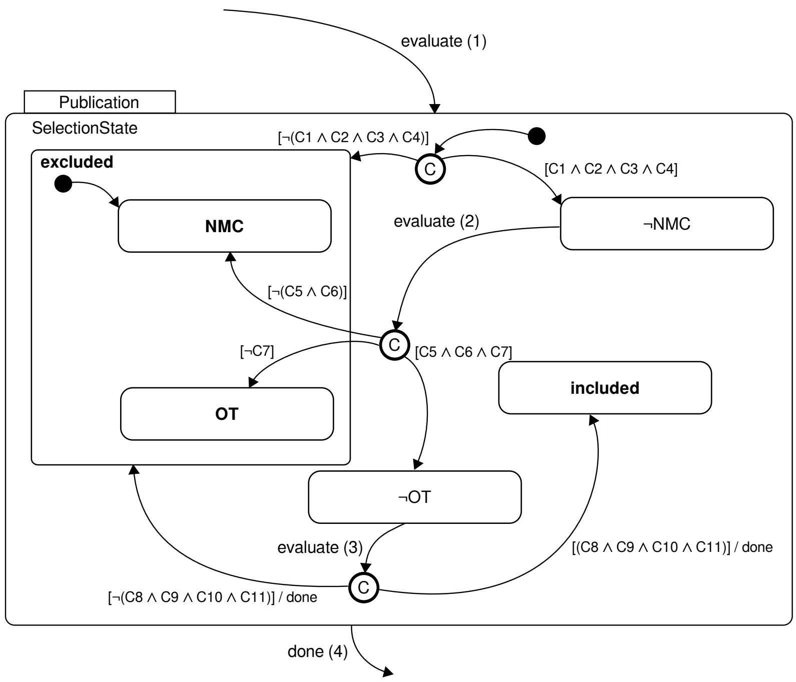


Figure 5: Selection states (i.e., `NMC`, `OT`, `excluded`, `included`) of a publication under review, following from the evaluation order between venue-specific and publication-specific criteria.

For documentation and analysis purposes, the final selection states (`included`, `excluded.NMC`, `excluded.OT`) are recorded (see also Section 7).

Table 15 aggregates the number of papers excluded during the selection procedure, grouped by exclusion criteria (see also Figure 6). In total, 5,731 reviews were conducted by three different reviewers. All articles considered (5,023) were reviewed by at least one reviewer. Excluded articles were classified according to three criteria: NMC (article does not match inclusion criteria), OT (article is off topic; i.e. it does not deal with UML-based DSML developments), and ERR (article contains serious errors). The first review excluded 4,859 articles (81% NMC, 18% OT, <1% ERR; see Table 16). We selected 190 articles (21%) of all ERR- and OT-classified articles (903) as well as all included articles (100) from the first reviewer for a second classification by a different reviewer. Additionally, there were a few articles the first reviewer was not sure about whether they should be included or excluded (31). These, as well as all duplicates (33) were also double-checked. Thus, the second reviewers classified 354 articles of which 240 articles where excluded (20% NMC, 74% OT, 6% ERR; see Table 16). The final decision was made by comparing both reviews for all 354 articles and, if differences had occurred, by agreeing on a final decision by both reviewers. The final decision excluded 237 articles (18% NMC, 76% OT, 6% ERR; see Table 16), leaving 117 articles in the paper corpus (84 unique articles and 33 duplicates).

Table 15: Overview of excluded/included papers during selection (OT, NMC) and quality assessment (erroneous studies, duplicates).

Hits/papers	ACMDL	IEEEXplore	Scopus	SpringerLink	Main Search	Snowballing	QGS	Total
Total	933	1,845	2,000	1,000	5,778	2,337	37	8,152
Unique	853	1,713	1,253	876	4,695	2,337	37	7,069
Considered	781	94	991	812	2,678	2,337	8	5,023
`excluded.OT`	186	11	266	241	704	170	2	876
`excluded.NMC`	571	73	681	543	1,868	2,128	1	3,997
`ERR`	8	3	9	5	25	6	2	33
Duplicated	1	1	3	3	8	25	0	33
Included	15	6	32	20	73	8	3	84

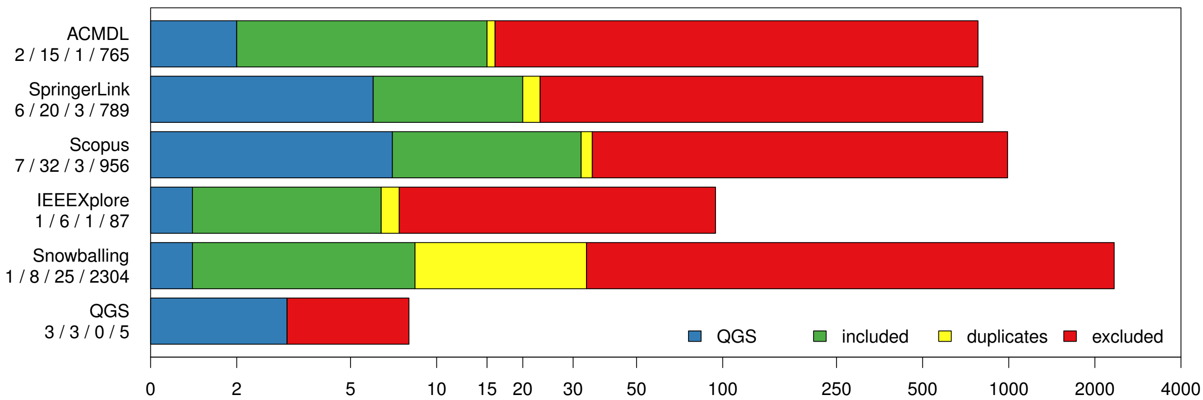


Figure 6: Classification of articles found per search engine/method. Please read the figures on the x-axis as follows: number of QGS articles in included articles / number of included articles / number of duplicated articles / number of excluded articles.

Table 16: Classification of excluded articles (reviewer 1 / reviewer 2 / final decision).

Classif.	ACMDL	SpringerLink	Scopus	IEEEXplore	Snowballung	QGS	Total
NMC	559 / 15 / 12	530 / 14 / 14	671 / 13 / 10	70 / 1 / 3	2,125 / 4 / 3	1 / 0 / 0	3,956 / 47 / 42
OT	190 / 31 / 32	240 / 52 / 52	256 / 57 / 58	11 / 4 / 3	174 / 32 / 33	2 / 2 / 2	873 / 178 / 180
ERR	6 / 5 / 5	4 / 3 / 3	9 / 5 / 5	3 / 0 / 0	6 / 0 / 0	2 / 2 / 2	30 / 15 / 15
Total	755 / 51 / 49	774 / 69 / 69	936 / 75 / 73	84 / 5 / 6	2,305 / 36 / 36	5 / 4 / 4	4,859 / 240 / 237

6.3 Quality Assessment

To further assess the quality of the papers that we considered as inclusion candidates (see Section 6.1), we evaluated the papers and their content regarding two aspects: 1) duplicate publications and 2) the correctness of the DSML design documentation:

1. *Duplicates*: The duplicate-detection procedure as part of our data cleansing step risks missing duplicates. For example because an identical publication could be listed with two separate DOIs

(e.g., due to parallel publishing channels for proceedings articles). Similarly, title matching might have missed duplicates because of different practices of reporting main and subtitles of publications. Therefore, during manual quality assessment, we assess the manually inspected publications for duplicates (e.g., by maintaining a reference list of DSML project names, author names etc.).

> From the 106 candidate publications from the main search, eight previously undetected duplicates were found (see Table 15). Backward snowballing yielded another 25 duplicates; resulting in 33 duplicates in total.

2. *Correctness*: The papers finally included in our selection (see Section 6.1) are subjected to an assessment of the correctness of the documented DSML design, especially the formalized design options. To the extent the DSML documentation permits such an assessment, the reviewers will evaluate whether the modeling and specification artifacts violate, for example, syntactic and/or semantic rules set by the (meta-)modeling language (CMOF, UML Infrastructure, UML Superstructure). In contrast to related work (see, e.g., [40]), we explicitly factor out interpreting intended domain semantics and application scenarios of DSMLs (e.g. whether the terminology used for model elements matches common domain vocabulary or whether models are unambiguously defined in terms of linguistic concepts). Included papers, which exhibit formalization and/or critical documentation defects, will be marked `ERR`oneous (`ERR`). In the following, such publications are excluded from the extraction of encoded DSML design decisions. In case of doubts and/or an unresolvable disagreement between reviewers (i.e., the two independent data extractors; see Section 7.2), the paper is excluded from decision extraction. However, formalization and documentation defects will be explicitly documented in the resulting research report. Focus will be set on the definition and formalization of the DSML language model because this particular design decision is mandatory in language model driven DSML development.

   Based on the state of relevant literature (see, e.g., [4, 6, 21, 23, 31, 40, 48, 83]) and on our experience, including the pilot study, we will assess the included publications for (but not limited to) the following types of formalization and design-documentation defects (depending on the corresponding extension technique):

   - Defects in language model formalization (see, e.g., [40]), such as, not standard compliant metamodel definitions or ambiguous specifications for metamodel elements.
   - Defects in applying UML profiles (see, e.g., [83]) and/or defects in usage of the stereotype mechanism (see, e.g., [48]), such as, missing or underspecified profile definitions or syntactical and semantic defects in stereotype specifications. Defects in the usage of profiles/stereotypes are not only facilitated by their ambiguous definition in the UML specification, but also because the profile/stereotype specification has changed in UML 2.x compared to previous versions UML 1.x (see, e.g., [6, 48]).
   - Defects because of tool-specific definitions and limitations, such as, using implementation-specific, not standard compliant syntax and semantics or insufficient support of MOF-compliant metamodeling layers (see, e.g., [4, 31]).
   - Defects related to the specification of language model constraints, such as, logical errors, ambiguous definitions, or syntactically incorrect formal constraints (see, e.g., [21, 23]).

> From the 106 candidate publications from the main search, 25 were marked `ERR` (see Table 15).
>
> During backward snowballing, we identified six additional erroneous publications; adding up to 31.
>
> By evaluating the QGS articles neither found via the main search nor via backward snowballing, we identified two additional erroneous publications; summing up to 33 in total.

6.4 Applications of Selection Criteria

During the different review phases, i.e. establishing the QGS corpus (see Section 4.1), running engine-based search (see Section 4.1), and performing the final backward-snowballing search (see Section 5), the selection criteria take effect differently.

6.4.1 QGS-specific Selection Process

We applied the selection criteria for the identification of suitable publication venues when establishing the QGS corpus (see Section 4.1). However, for this purpose, the criteria and the procedure are slightly modified, primarily to increase the inclusion barrier for prospective QGS publications:

- *Venue identification*: In this step, the venue-specific criteria C_1–C_4 are evaluated. For a publication venue to be acceptable under C_3, the venue must be listed with *both* sources (MS/AS [72] as well as SJR [99] and Xie [123], respectively). To the extent venue identification is the base for the search-engine selection, these criteria affect the engine selection.
- *Paper screening*: The papers are screened based on title, abstract, and keywords. Due to the extensiveness of the this manual screening step (see Section 4.1), the publication-specific criteria C_5–C_{10} are *not* evaluated when deciding whether a paper is incorporated into the QGS corpus or not.
- Quality assessment (see Section 6.3) and decision-data extraction (see Section 7) are not performed immediately.

6.4.2 Engine-specific Selection Process

The second application of the selection criteria differs in two main aspects from the first one (see Section 6.4.1): On the one hand, the publication-specific criteria are evaluated. On the other hand, venue selection is less restrictive than for QGS construction.

- *Venue identification*: In this step, the venue-specific criteria C_1–C_4 are evaluated. For a venue to be acceptable under C_3, the venue must be listed with at *least one* of the sources (MS/AS [72] as well as SJR [99] and Xie [123], respectively).
- *Paper screening*: For this step, the publication-specific criteria C_5–C_{11} are evaluated.
- Quality assessment (see Section 6.3) and decision-data extraction (see Section 7) are performed immediately.

6.4.3 Snowballing-specific Selection Process

In this third application, we applied the exact same evaluation steps as for the engine-specific selection; see Section 6.4.2.

6.5 Selection Validity

The IRR statistics used will be Cohen's Kappa $\widehat{\kappa_C}$ for the selection decisions. Hits rated by just one author, originating from collecting inter-ratings only from a subset of co-reviewed search hits, will be considered when computing the Kappa coefficient (i.e., to establish the chance agreement). However, every search hit is reviewed by at least one author (see above and also Section 7.2).

Table 17 summarizes the collected ratings of the 2,678 search hits into the two categories `included` ("y") and `excluded` ("n"):

- 2,400 hits were only reviewed by one author.
- 278 hits were reviewed by two authors.
- Percent agreement (p_a) amounted to ≈ 0.885.
- Cohen's Kappa $\widehat{\kappa_c}$ including missing data amounted to ≈ 0.875.

Table 17: 2,678 rating items; $p_a \approx 0.885$; $\widehat{\kappa_c} \approx 0.875$; $Var(\widehat{\kappa_c}) \approx 0.0004$.

		Rater B			
		n	y	NA	Total
Rater A	n	183	2	2,400	2,585
	y	30	63	0	93
	NA	0	0	0	0
	Total	213	65	2,400	2,678

Tables 18–20 summarize the collected ratings for each of the three snowballing iterations into the two categories `included` ("y") and `excluded` ("n"). As the reference items of the iterations were rated separately from each other, we do also report the reliability analyses separately:

- The numbers of references only reviewed by one author were 2,054 (1st iteration), 194 (2nd iteration), and 20 (3rd iteration).
- The numbers of references reviewed by two authors were 62 (1st iteration), six (2nd iteration), one (3rd iteration).
- Percent agreement (p_a) amounted to ≈ 0.968 (1st iteration), ≈ 0.833 (2n iteration), and 1 (3rd iteration).
- Cohen's Kappa $\widehat{\kappa_c}$ including missing data amounted to ≈ 0.967 (1st iteration), ≈ 0.829 (2nd iteration), and 1 (3rd iteration).

Table 18: 1st iteration of snowballing procedure (2,116 rating items); $p_a \approx 0.968$; $\widehat{\kappa_c} \approx 0.967$.

		Rater B			
		n	y	NA	Total
Rater A	n	56	1	2,054	2,111
	y	1	4	0	5
	NA	0	0	0	0
	Total	57	5	2,054	2,116

Table 19: 2nd iteration of snowballing procedure (200 rating items); $p_a \approx 0.833$; $\widehat{\kappa_c} \approx 0.829$.

		Rater B			
		n	y	NA	Total
Rater A	n	5	1	194	200
	y	0	0	0	0
	NA	0	0	0	0
	Total	5	1	194	200

Table 20: 3rd iteration of snowballing procedure (21 rating items); $p_a = 1$; $\widehat{\kappa_c} = 1$.

		Rater B			
		n	y	NA	Total
Rater A	n	1	0	20	21
	y	0	0	0	0
	NA	0	0	0	0
	Total	1	0	20	21

7 Data Extraction

The data extraction procedure described below applies to the sub-steps of establishing the QGS corpus (see Section 4.1), to the main engine-based search (see Section 4.1), and to the final backward-snowballing search (see Section 5).

7.1 Data Records

The data records are stored and maintained as Google Drive spreadsheets. For each search engine, there will be one sheet named after the respective search engine. The column "organization" in each sheet follows from the record structure given below. The data extracted by the two different data extractors will be maintained in the same sheet. However, by controlling column visibility, the two data extractors will be blinded for the data extracted by its alter. Although the intention is to arrive at a full (and if needed, negotiated) agreement on each publication, we compute IRR statistics to reflect on the state of agreement before negotiation and the complexity of negotiating an agreement. The IRR statistics used will be Cohen's Kappa for the inclusion/exclusion decisions and the Kupper-Hafner Index [66] on ratings for the design-decision data.

In our pilot review, and before revision 3 of this document, the data items recorded per publication slightly differed from the ones in the authoritative review (see below). The original data-extraction procedure, as of revision 2 of this protocol, is described in full detail in [36]. The main differences arise from additional search engines (and their characteristics; e.g., Google Scholar, MS Academic Search) used for the pilot and different analyses planned for the pilot. In our pilot, we did not include the DOI as a record item which complicated duplicate cleansing. Therefore, we changed our extraction procedure accordingly. Based on the pilot search, we also aimed at identifying search terms which did not produce any hits to further revise our search string. Therefore, we also collected the search terms responsible for each search hit. The remainder of the record entries per publication were re-adopted for the main review, despite some relabeling (e.g., annotator vs. extractor).

For each paper in the corpus, a basic data record is created. The data for this record is extracted from the paper content and its accompanying bibliographical metadata, as provided by the search engine returning the respective paper as search hit. Depending on the data-extraction capabilities of the search engine, some data entries are created manually or adopted in a semi-automated manner.

- *Data extractor #1*: Name of the reviewer (Hoisl, Sobernig, Strembeck) a) running the authoritative search yielding the publication as search hit, b) making a first selection decision according to the criteria in Section 6, and c) extracting the first corresponding data record about the selected search hit (publication).
- *Data extractor #2*: Name of the reviewer (Hoisl, Sobernig, Strembeck) a) making a second, independent selection decision according to the criteria in Section 6 and b) extracting the second corresponding data record about the selected search hit (publication). The person must be different from the person acting as first extractor.
- *Unique identifier*: If available, we retrieve the *Document Object Identifier* (DOI) to disambiguate the publication item within each search-engine result set and in the total result set. If missing, we generate an identifier for the disambiguation purpose in our review.
- *Authors*: The list of publication authors as retrieved from the respective search engine.
- *Title*: The publication title as retrieved by the respective search engine.
- *Publication year*: The year of publication as retrieved by the respective search engine.
- *Range of page numbers*: The range between start and final page as retrieved by the respective search engine.
- *URL*: The locator as retrieved by the search engine. DOI-resolver locators, if available, are preferred over engine-specific or publisher-specific ones.
- *Venue*: The publication venue of the article. Often synonymous with the title of the containing publication, if applicable (e.g., a proceedings or journal title).
- *Publisher*: The publisher which is responsible for the corresponding publication (e.g., a proceedings or journal publisher).
- *Abstract*: The publication's English abstract, if available.
- *Keywords*: The keywords provided by the authors of the publication, if available.
- *Comments*: Allows the two extractors to leave auxiliary notes. For *excluded* papers, the two extractors signal the reason for exclusion (see also Sections 6.2 and 6.3):
  - N(ot) M(atching) C(riteria), `NMC`;
  - O(ff) T(opic), `OT`;
  - ERR(oneous).

Note that, as we employed a single, complex search string for each of the four selected search engines, there was no need to separately store the corresponding search string producing the publication as search hit [36].

Apart from the entries *data extractor #1*, *data extractor #2*, and *search engine*, all fields per item could be extracted automatically from the result sets returned by the search engines. While deviating in their labeling, all required data was available from the search engines and could be aligned with little effort. Extracting DOIs, however, led to missing data: Not for all items, a DOI was returned. Rather than iterating over the entire search hits to identify missing DOIs, we manually added them if needed for included papers (e.g., to complete their bibliographical record and for identifying duplicates).

For the papers resulting from the step of engine-based search, two additional fields were recorded:

- *Search engine*: The name of the search engine yielding the publication as search hit. The possible engine values are set by the engine selection outlined in Section 4.1.
- *Search datetime*: A datetime stamp indicating when the search yielding the publication as hit was executed.

Note that, as all four search engines allowed for complex search queries, the *Search datetime* for all publication items of a given, engine-specific result set are the same.

For the papers included into the review result set (according to Section 6), we extract further data on DSML design decisions:

- *DSML name*: If identifiable from the paper, the name given to the DSML by the publication's authors (e.g., project or working title, acronyms, UML package names). If not identifiable, we assigned a name based on cues by the authors (paper title, names of central model elements etc.).
- *Application or target domain(s) of DSML*; as identified by the publication's authors or as identified by the reviewers through studying the papers, through reading the running and motivating examples given, and/or the application cases reported.
- *Affected UML diagram types*; by studying the definitions of the corresponding UML profile or UML metamodel extensions (e.g., as indicated by the specific UML Superstructure packages providing the extended metaclasses). We consider the 14 diagram types as enumerated by Annex A in [78], in particular Figure A.5:
  - Activity diagram;
  - Class diagram;
  - Communication diagram;
  - Composite structure diagram;
  - Deployment diagram;
  - Interaction overview diagram;
  - Object diagram;
  - Package diagram;
  - Profile diagram;
  - State machine diagram;
  - Sequence diagram;
  - Timing diagram;
  - Use case diagram.
- *D1*: One or more decision options recovered for the given DSML, taken at the decision point *language-model definition* according to the two data extractors. See the catalog for available options [54].
- *D2*: One or more decision options recovered for the given DSML, taken at the decision point *language-model formalization* according to the two data extractors. See the catalog for available options [54].
- *D3*: One or more decision options recovered for the given DSML, taken at the decision point *language-model constraints* according to the two data extractors. See the catalog for available options [54].
- *D4*: One or more decision options recovered for the given DSML, taken at the decision point *concrete-syntax definition* according to the two data extractors. See the catalog for available options [54].
- *D5*: One or more decision options recovered for the given DSML, taken at the decision point *behavior specification* according to the two data extractors. See the catalog for available options [54].

- *D6*: One or more decision options recovered for the given DSML, taken at the decision point *platform integration* according to the two data extractors. See the catalog for available options [54].

The above D1–D6 entries are recorded threefold for each publication. Once for each data extractor, respectively, and a third time to represent the negotiated agreement of the two reviewers in case of an initial disagreement.

Data extraction was performed on the 84 finally included papers (see also Section 6.3). During data extraction (especially of DSML names, author lists, diagram types, and application domains) we identified non-original and complementary papers which document one and the same DSML (e.g., different design fragments, different application scenarios, at different research stages). This is a deviation from the original protocol, as we expected to trap non-original publications during the selection procedure (see C_{10} in Section 6.1). In total, we identified eight publications out of 84 included ones covering four unique DSMLs (see Table 21). As a result, data extraction was performed on 80 rather than on 84 items. Complementary publications per DSML (i.e., two each) were both considered for extracting the design-decision data (e.g., by forming the union of decision-option sets, of the affected diagram types etc.).

Table 21: Overview of the rationale for considering eight publications documenting four DSML designs.

Publications	DSML	Comment
Gilmore et al. [39] Mayer et al. [70]	UML4SOA	[39] covers UML4SOA in an overview to emphasize on an extension part for non-functional properties (NF), while [70] gives more details. However, UML4SOA contained the NFP part right from the beginning, as documented in deliverables of the underlying SENSORIA project.
Bernardi et al. [13] Bernardi et al. [14]	MARTE-DAM	[13] elaborates on the details of the redundancy and maintenance parts of MARTE-DAM, while [14] covers the rest; see `https://bitbucket.org/mberenguer/marte-dam/wiki/Home` for an integrated overview.
Hatebur and Heisel [46] Hatebur and Heisel [47]	UML4PF	[46] is about the tools supporting the approach based on problem frames more generally presented in [47].
Panesar-Walawege et al. [81] Panesar-Walawege et al. [82]	IEC61508	[82] presents an application case of the IEC61508 implementation to the petroleum industry, [81] captures the approach more generally.

Decision Phases. All of the 80 DSMLs covered the phases of language-model definition (D1) and language-model formalization (D2), respectively. This was also a minimum requirement on a DSML design to become included which was evaluated during the different selection stages (see, e.g., Section 6.1). For the remaining four decision phases (D3–D6), we recorded whether any decision was taken at all for a given DSML. For D3 (language-model constraints) and D4 (concrete syntax), important shares of the DSMLs involved one or several documented decisions. 32 of 80 DSMLs did not document language-model constraints (D3; e.g., OCL expressions) beyond the constraints expressed directly via formalized language models. Only seven DSMLs did not involve any decision on the concrete syntax (D4) of the DSML. For the remaining decision phases of behavioral specification (D5) and platform integration (D6), the consulted sources did not report any decision for the majority of the 80 DSMLs; i.e., for 77 and 54 out of 80, respectively.

Table 22 summarizes the collected option codes per decision phase for 80 DSML designs. The option codes are the final ones, in doubt negotiated between the rater pairs.

Decision Options. In total, we extracted decisions mapping to 623 option codes across all the 80 DSML projects. 170 were no-option codes signalling the actual lack of documented decisions for certain decision phases (see also above). Table 22 shows the number of applied design options of included DSML projects.

For the decision phase of language-model definition (D1), we marked 111 option codes over all 80 DSMLs. Each of the four available option codes was identified at least once.

- For 52 DSMLs, we found just a single applied option. 25 DSMLs applied two and three DSML designs applied three out of four possible options. Not a single design used all four options.

[5] *Projects per option*: The percentage of projects which applied the design option.
[6] *Option per decision*: The percentage of an option chosen at a design decision point.

Table 22: Applied design options of included DSML projects. Note: For D2, a special-purpose option code "(2.4)" was introduced having eleven occurrences which are excluded from the descriptive analysis here; see also Section 6.2.

Option	#	p/o[5]	o/d[6]	Total
O1.1	80	100%	72%	
O1.2	5	6%	5%	
O1.3	3	4%	3%	
O1.4	23	29%	21%	111
O2.1	4	5%	5%	
O2.2	62	78%	73%	
O2.3	17	21%	20%	
O2.4	2	2%	2%	85
O3.1	31	39%	32%	
O3.2	0	0%	0%	
O3.3	0	0%	0%	
O3.4	35	44%	36%	
O3.5	32	40%	33%	98
O4.1	62	78%	39%	
O4.2	14	18%	9%	
O4.3	3	4%	2%	
O4.4	1	1%	1%	
O4.5	1	1%	1%	
O4.6	69	86%	44%	
O4.7	7	9%	4%	157
O5.1	1	1%	1%	
O5.2	1	1%	1%	
O5.3	2	2%	2%	
O5.4	0	0%	0%	
O5.5	77	96%	95%	81
O6.1	4	5%	4%	
O6.2	16	20%	18%	
O6.3	7	9%	8%	
O6.4	1	1%	1%	
O6.5	9	11%	10%	
O6.6	54	68%	59%	91
Total	623			623

- All 80 DSML design reports described the language-model definition using a textual, natural-language representation (O1.1). In 52 DSMLs, this style of defining the language model is the only adopted option at all. These correspond directly to the 52 DSMLs with a single decision option only, see above.
- 23 reported DSMLs provide a formal diagrammatic definition (O1.4) of their language model prior to actually formalizing (implementing) the language model on top of UML. In all 23 cases, this decision option is accompanied by a textual, natural-language representation (O1.1) as a complementary way of defining the language model.

As for implementing the language model using UML (D2), we noted 85 option codes in total. Each available option code was marked at least for one DSML design.

- For 75 out of the 80 DSMLs, we observed only one formalization applied in this decision phase. Five projects adopt two options out of the four available ones.
- More than 3/4 of the DSML projects (≈78%) involved the creation of one or a set of complementing UML profiles (O2.2) to realize the language model.
- ≈21% of the DSMLs (17/80) extend the UML metamodel in a non-intrusive manner, i.e., without modifying the UML metamodel (O2.3).
- Only three of the 17 DSMLs based on a UML metamodel extensions (O2.3) pair with UML profiles (O2.2), leaving 14 DSMLs as pure UML metamodel extensions.
- Only two DSMLs explicitly modify the conditions of the UML metamodel (O2.4) along with extending it (e.g., adding metaclasses; O2.3).

As for expressing language-model constraints beyond the intrinsic constraints contained by formal language models (D3), we extracted 98 option codes, out of which 32 were no-option codes (O3.5). In the remainder of 66 option codes, only two out of the four available options were found applied: constraint-language expressions (O3.1) and textual annotations (O3.4). Two options were not found applied in any of the 80 reviewed DSML designs at all: code annotations (O3.2) and translational constraining (O3.3).

- The two observed option codes (i.e., O3.1, O3.4) take approximately equal shares in the total occurrences (O3.1: 31, O3.4: 35).
- In 30 DSMLs, only one option was adopted. Both, O3.1 and O3.4 are adopted equally often (13 and 17 times, respectively).
- In the remaining 18 DSMLs, both O3.1 and O3.4 are used in a complementary manner.

Codifying the design decisions on the concrete syntaxes (D4) of the 80 DSMLs yielded the largest number of recorded option codes (157). Only seven DSMLs did not document any decision on a concrete syntax or did not foresee a concrete syntax at all (e.g., because the primary intention of the DSML is serving as an intermediate representation in a M2M transformation scenario). Each of the available six options (O4.1–O4.5) has been found applied at least once.

- Comparatively many DSML projects (49) took two complementing decision options on concrete-syntax style. In 17 projects, only one decision option was applied, in 14 DSMLs there were three options out of the six available ones.
- The two most frequently found decision options are model annotations (O4.1: 62 DSMLs) and the unmodified reuse of UML diagram symbols (O4.6: 69 DSMLs).
- The most frequently applied solitary concrete-syntax option is symbol reuse (O4.6: 6 DSMLs).
- 48 of the 49 two-options DSML designs have O4.1 and O4.6 as the option pair.
- A dedicated UML diagram-syntax extension (O4.2; e.g., by introducing new symbols through resampling existing ones) is only adopted in 14 DSMLs.
- Intermediate syntax styles such as mixed syntaxes (O4.3), frontend syntaxes (O4.4), and alternative syntaxes (O4.5) are found in at least one but not more than three DSMLs.

A comparatively large number of DSML documentations remain silent or lack a behavioral specification (D5). This is reflected by the lowest count of 81 option codes of all decisions phases, including 77 no-option codes (O5.5) recorded for this decision phase. Only three DSMLs modify and/or extend the underlying UML behaviors. The three DSMLs document these refinements of behavioral-semantics by adopting a UML M1 model representation (O5.1), a formal textual specification (O5.2), or in an informal textual way (O5.3). Only one of these three DSMLs applies two options in a complimentary manner (O5.1 and O5.3), the remaining two picked a single option only (O5.2, O5.3). Constraining model execution (O5.4) as one option identified by our catalog out of the four available (O5.1–O5.4) was not found applied in any DSML.

Platform integration (D6) was documented in terms of 91 option codes. At a level comparably low with respect to D5, decisions on platform-integration techniques supported by a DSML (D6) were not documented by ≈68% of the projects (O6.6: 54 DSMLs). In the remainder of 26 DSMLs, each of the available platform-integration techniques (O6.1–O6.5) was applied at least once.

- 19 DSMLs applied a single platform-integration option, four DSMLs were found to realize a combination of three options, and three DSMLs combined two options each.
- The single, most frequently applied option are DSMLs supporting generation templates (O6.2), with 20% (16/80) of the DSMLs. Generation templates (O6.2) are also the most frequently employed solitary decision option with twelve out of 19 DSMLs.
- Generation templates are followed by M2M transformations (O6.5: 9 DSMLs) and API-based generators for platform-specific models (O6.3: 7). M2M transformations (O6.5) are found mostly (i.e., in 6 out of 9 DSMLs) in combination with at least one other option.
- Intermediate model representations (O6.1: 4 DSMLs) and model execution (O6.4: 1) are the comparatively least frequently found options.

7.2 Procedure

For each result set (per search engine), two authors (Hoisl, Sobernig, Strembeck) will be nominated as the two data extractors per search hit (per publication), in a way that preserves mutual exclusion. Once the target size of the per-researcher subsets has been established (based on the cutoff computation from the QGS-based sensitivity analysis; see Section 4.1), the actual subset for each researcher will be computed. The objective is to balance the workload between the three researchers (in both roles) as far as possible.

Both data extractors assess the hits (publications) in each result set independently from each other. Each extractor verifies the automatically extracted data record for each publication (see Section 7.1). Second, each data extractor applies the inclusion and exclusion criteria (see Section 7.1). If included, each data extractor continues by extracting the design-decision data from the included publication in a third step (see Section 7.1).

Each second data checker applies the same three-step sequence. However, the number of publications rated and reviewed by each of the data extractors per result set can differ from each other. All candidate papers for the QGS publication corpus are reviewed by two independent extractors (see Section 4.1). For the engine-based and snowballing searches, however, the publication-set to be verified by the second extractor is a subset of the extractor's papers to balance between the checking effort and the need for accuracy (which is particularly critical for corner cases such as OT and ERR ratings; see above). The second extractor's subset will be established as follows:

1. By considering all publications included by the first data extractor (i.e., to check for false positives).
2. By omitting all publications excluded by the first data extractor because they do not match the objectively verifiable inclusion criteria (NMC; see above).
3. The remainder are publications excluded by the first data extractor because of being OT or ERR (see above). The checker's subset will contain a randomized 20% sample drawn from this remainder to check for false negatives.[7]

Table 23 documents the final role/engine assignments to each of the three reviewers. The workload in terms of unique publications rated as both extractor and checker distributes as follows: 1. Hoisl: 3,469; 2. Sobernig: 1,980; 3. Strembeck: 94. A balanced workload could not be achieved due to individual time constraints in the respective review period, access limitations to publisher databases (e.g., personalized VPN-based access to Scopus) while—at the same time—maintaining mutual exclusion between checker and extractor roles.

Table 23: Overview of role/engine assignments for the three researchers. Within braces, the number of investigated and rated publications per role/engine are depicted.

	Extractor #1	Extractor #2
	QGS	
QGS corpus	Hoisl (46), Sobernig (37)	Sobernig (46), Hoisl (37)
DSML selection[8]	Sobernig (8)	Hoisl (7)
	Main search	
ACMDL	Hoisl (781)	Sobernig (65)
IEEEXplore	Strembeck (94)	Hoisl (13)
Scopus	Hoisl (991)	Sobernig (108)
SpringerLink	Sobernig (812)	Hoisl (92)
	Snowballing	
Pool 1	Hoisl (1,481), Sobernig (635)	Sobernig (48), Hoisl (14)
Pool 2	Sobernig (200)	Hoisl (6)
Pool 3	Sobernig (21)	Hoisl (1)

If there is a disagreement in the extracted data between the two independently operating extractors, the two extractors will re-read the paper and reach a final decision in a joint session.

7.3 Reliability

The extraction procedure will result in independent ratings of two reviewers (see Section 7.2) on both the selection state (included, excluded) of a publication and on the decision options detected by reflecting

[7] Picking a sample size of 20% was a heuristic based on our pilot observation that ≈17% of the reviewed publications were included.

[8] Only those QGS publications not already retrieved by the main search.

on the documented DSML design. To quantify the level of agreement between these independent ratings (i.e., the inter-rater reliability; IRR), two different IRR statistics are required to accommodate different measurement scales underlying these two different ratings (selection state vs. decision options):

- *Cohen's Kappa coefficient* $\widehat{\kappa_c}$ [26] for two raters and two-level categorical ratings: The independently collected ratings on the selection state have two mutually exclusive levels (included vs. excluded). The $\widehat{\kappa_c}$ is suitable for quantifying agreement on this two-level categorical scale. We will report both the percentagewise agreement and the chance-corrected agreement. We will adopt a variant of the $\widehat{\kappa_c}$ which deals with missing ratings explicitly [42]. Ratings missing from a second rater are due to the fact that the co-rated publications only represent a subset of the totally reviewed publications (see Section 7.2).
- *Kupper-Hafner (KH) index* [66] for two raters and n-level nominal ratings: Each rater (that is, either the extracting or the checking reviewer) must assign at least one or multiple decision options per decision point (D1–D6) to a DSML design. This translates into a decision-data set having a multi-level nominal scale. The available levels of each decision point are the available decision options for this decision point according to our catalog [54]. We will report the Kupper-Hafner index [66] based on the levels actually selected by both reviewers (i.e., reflecting agreement on included levels only) and based on the total levels available for assignment (i.e., reflecting agreement on excluded levels). Again, we will report the uncorrected and chance-corrected variants of the KH index.

- Deviation: For D2, an additional, fifth level was introduced to indicate the potential of a UML metamodel modification, rather than a factual one (i.e., O2.4), due to an incomplete design documentation. For the sake of the reliability study, this nominal rating is reported as a separate level.
- Table 24 summarizes the obtained agreement levels (and corresponding variance estimates) per decision phase for 62 co-rated DSML designs.
- For 18 DSML designs, only one rating by one rater or even not a single but the final (negotiated) rating by both reviewers was available. This is, for instance, due to an initial disagreement about including or excluding a DSML design between the two raters.

To quantify the level of statistical insecurity underlying our data-generating process, for both statistics, we will additionally report the Jackknife estimate of variance.

Table 24: IRR per decision phase (D1–D6); Kupper-Hafner (KH) Index; 18 out of 80 items (DSMLs) were not co-rated (missing ratings); $\hat{\pi}$, $\hat{\pi}^*$: Percent agreement (excl., incl. missing ratings); $\hat{C}$, $\hat{C}^*$: Chance-corrected KH indices (excl., incl. missing ratings); $Var(...)$: leave-one-out Jackknife variance estimate.

	D1	D2	D3	D4	D5	D6
$\hat{\pi}$	0.77	0.91	0.60	0.83	0.95	0.64
$Var(\hat{\pi})$	*0.0009*	*0.0009*	*0.0029*	*0.0017*	*0.0008*	*0.0036*
$\hat{C}$	0.64	0.89	0.50	0.77	0.94	0.56
$Var(\hat{C})$	*0.0022*	*0.0014*	*0.0046*	*0.0029*	*0.0012*	*0.0053*
$\hat{\pi}^*$	0.85	0.96	0.84	0.94	0.98	0.87
$Var(\hat{\pi}^*)$	*0.0004*	*0.0002*	*0.0005*	*0.0002*	*0.0001*	*0.0004*
$\hat{C}^*$	0.77	0.95	0.80	0.92	0.98	0.84
$Var(\hat{C}^*)$	*0.0012*	*0.0003*	*0.0008*	*0.0003*	*0.0002*	*0.0007*

The extraction of decision data according to the catalog [54] was first performed independently by two authors in case of taking a positive selection decision on a DSML. In a second iteration, in case of an initial disagreement, a final extraction including an agreed code marking of decisions was negotiated between the two authors. An initial disagreement could result from an opposing selection decision (e.g., author A included while author B excluded the paper) and/or, for DSMLs included by both authors, from applying different codes for the decision options observed for a documented DSML design. From the 80 DSML designs under consideration, 62 DSMLs received two independent ratings initially. In 18 cases, there was only one or even no independent extraction decision recorded as decision-data rating. However, for all 80 DSMLs considered, there was a third, negotiated extraction decision and corresponding option sets.

For the 62 co-rated DSML designs, we can report the extent to which the rater pairs agreed in applying codes for decision options as defined by the catalog [54] (see Table 24). This agreement level signals how reliable the finally recorded option sets per DSML are in terms of different raters being in agreement without negotiation. For all decision phases (D1–D6), the raters showed a chance-corrected agreement $\hat{C}$ of at least or more than 0.5, indicating the agreement per DSML in terms of the overlap of decision options marked by both reviewers relative to the maximum overlap possible, on average over all 62 DSMLs. By incorporating agreement in terms of decision options excluded by both authors, an overall agreement level after removing chance $\hat{C}^*$ of equal to and greater than 0.75 for each decision phase can be reported. A closer look reveals:

- While generally starting at an acceptable level, the comparatively lowest agreement levels (i.e., based on the overlap between selected decision options) could be obtained for decision options observed for language-model constraints (D3; $\hat{C} \approx 0.5$) and platform integration (D6; $\hat{C} \approx 0.56$).
- Medium agreement levels (i.e., based on the overlap between selected decision options) were reached for language-model definition (D1; $\hat{C} \approx 0.64$) and for concrete-syntax decisions (D4; $\hat{C} \approx 0.77$).
- Comparatively high agreement levels (i.e., based on the overlap between selected decision options) are found for language-model formalization (D2; $\hat{C} \approx 0.89$) and for behavior specification (D5; $\hat{C} \approx 0.94$).

8 Data Analysis

To detect patterns in the observed DSML designs, based on their representations as decision-option sets, we apply a frequent item-set analysis [16]. In the following, we introduce key concepts (e.g., support, closedness, maximality, freeness) of this data-mining technique by giving an example from the data set actually obtained from our SLR. For the actual analysis, we processed our data set and computed the characteristic set restrictions using `arules` [44].

Table 25: A sample of ten DSMLs extracted from the SLR result set for illustration purposes. The *source* indicates the corresponding SLR publication, the DSML's application *domain(s)* encoded according to ACM CCS, the UML *diagram type(s)* tailored by a DSML, and two representations of the decision-*option sets*: atomic (option level) and aggregated (decision-point level).

DSML (Source)	Domain(s)	Diagram type(s)	Option set	
			atomic	aggregated
CompSize [69]	embedded systems, metrics, measurement, estimation	component, class	$\{1.1, 2.2, 3.5, 4.1, 4.6, 5.5, 6.6\}$	$\{d1, d2, d4\}$
EIS [77]	enterprise information systems	component, activity	$\{1.1, 2.2, 3.4, 4.1, 4.2, 4.6, 5.5, 6.6\}$	$\{d1, d2, d3, d4\}$
UACL [90]	telecommunications, availability	component, class	$\{1.1, 1.4, 2.2, 3.1, 3.4, 4.1, 4.6, 5.5, 6.6\}$	$\{d1, d2, d3, d4\}$
MoDePeMART [17]	software performance, measurement, metrics	class, state machine	$\{1.1, 1.4, 2.2, 3.5, 4.1, 4.6, 5.5, 6.6\}$	$\{d1, d2, d4\}$
UML-GUI [109]	graphical user interfaces	component, class	$\{1.1, 1.2, 2.1, 3.5, 4.7, 5.5, 6.3\}$	$\{d1, d2, d6\}$
SMF [73]	safety critical systems, software safety, fault tree analysis	use case, class, component	$\{1.1, 1.3, 1.4, 2.2, 3.5, 4.1, 4.6, 5.5, 6.6\}$	$\{d1, d2, d4\}$
BIT [2]	software testing, debugging	class	$\{1.1, 2.2, 3.1, 4.1, 4.6, 5.5, 6.1, 6.2, 6.5\}$	$\{d1, d2, d3, d4, d6\}$
UML-PMS [41]	ubiquitous and mobile computing, performance	activity	$\{1.1, 2.2, 3.4, 4.1, 4.6, 5.5, 6.6\}$	$\{d1, d2, d3, d4\}$
SECRDW [107]	data warehouses, security requirements	package, class	$\{1.1, 2.3, 3.5, 4.7, 5.5, 6.6\}$	$\{d1, d2\}$
UML4SOA [70]	service-oriented architectures	activity, class, component	$\{1.1, 2.2, 2.3, 3.1, 4.1, 4.2, 4.6, 5.5, 6.1, 6.3, 6.5\}$	$\{d1, d2, d3, d4, d6\}$

The task of identifying frequent patterns of option sets is specific to a given *base* of decision codes. In the following, the base consists of the six codes representing decision points on our catalog: $d1$–$d6$. In addition, such an analysis incorporates a collection of *observed* option sets which will be given by ten aggregated option sets from Table 25. An observed option set represents a complete DSML design. Any observed option set is a subset of the base. Given the above base, there are 16 possible, unique option sets which can be expressed using the six decision codes. The resulting design space of 16 option sets at the decision-point level is shown in Figure 7. By studying the data base of ten observed option sets alone, we obtain three initial observations:

- *Uniquely observed option sets*: In the collection of ten observed option sets, there are five *uniquely* observed option sets. See the corresponding five nodes in the Hasse graph in Figure 7, represented by solid rectangles. Conversely, there are eleven out of 16 potential option sets which cannot be found in the collection as-is (see the dashed rectangles in Figure 7).
- *Repeatedly observed option sets*: Each of the five unique option sets has at least one or more occurrences in the database. For example, there are three DSMLs sharing the aggregated option set $\{d1, d2, d4\}$ (i.e., CompSize, MoDePeMART, and SMF). Likewise, we find only one DSML (SECRDW) whose documented design reflects decisions for two decision points: $\{d1, d2\}$.
- By looking at the cardinalities of uniquely observed option sets we also learn about, e.g., the minimum number (two) or the maximum number (five) of decisions or decision points considered in the collection of ten DSMLs. In other words, no DSML design involves decisions at all six decision points.

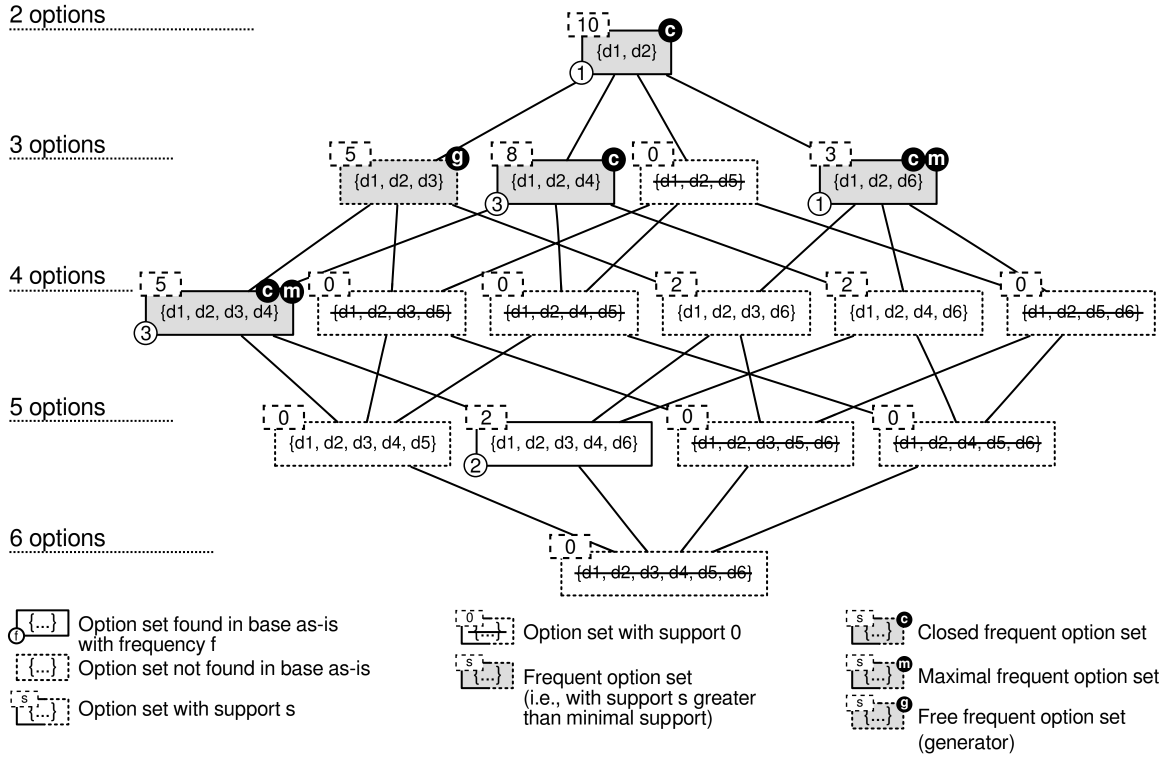


Figure 7: Visualization of the design space of decision points spawned by our decision-records catalog as a Hasse diagram of $\{O : O \in (\{d1, d2, d3, d4, d5, d6\}, \subseteq) \wedge \{d2\} \subset O \wedge \{d1\} \subset O\}$. Edges point downwards, omitting edge directions. Information from a frequent-item-set analysis using the base of ten observed option sets in Table 25 is superimposed onto the Hasse graph.

While immediately useful, we can gain additional insights from contrasting the observed option sets to the hierarchical structure of possible option (sub-)sets (see Figure 7). On the one hand, by considering the observed sets alone, we do not learn about characteristic subsets of options being recurring sets throughout the collection of ten DSMLs. For example, while the option set $\{d1, d2\}$ has been found to represent a concrete DSML design (SECRDW; see above), we omit how often this set re-appears as a proper subset of the remaining nine option sets. On the other hand, we do not learn whether there are other patterns of options re-occurring as characteristic subsets only throughout the observed base of DSMLs. In Figure 7, this is exemplified by the possible option set $\{d1, d2, d3\}$ which does not characterize a single DSML design as-is (i.e., it is not contained by the collection), but will be found shared as part of the observed option sets of five DSMLs.

The *support* s of a given option set is expressed as the number of observed option sets in the collection in which it is contained as a subset [16]. The support can be computed for all 16 possible option sets on the design space depicted in Figure 7. For example, the support of option set $\{d1, d2\}$ amounts to 10 (in absolute terms; relative support is 10/10 or 0.1). In fact, this subset is contained by all ten DSMLs while there is only one DSML which is described by this option set exactly. A support of 0 indicates that an option set is not present as-is in the collection *and* that it is not contained by any option superset residing at the next-lower levels of the design space (assuming the top-down ordering in Figure 7). Consider the example of $\{d1, d2, d5\}$. There is no DSML which is exactly characterized by these three decision-point codes and there is no DSML which contains this subset. This notion of support—in contrast to occurrence frequency of unique option sets in the collection—allows for applying a number of restrictions on the option sets: minimum support (frequency), closedness, maximality, and freeness. By combining these restrictions when filtering the total number of option sets expressible, we can identify characteristic option sets of interest (such as prototype designs; see Table 26 below).

- *Frequent option sets* [9, 16]: Typically, we are interested in finding option sets out of the total design space which have a *minimum support*. Minimum support reflects the requirement that a given option set must occur or be contained by a minimum number of option sets, i.e., DSMLs. All (observed and possible) option sets having a support s equal to or greater than the minimum support s_{min} are called *frequent* option sets. In our running example based on the DSMLs in Table 25, we apply a $s_{min} = 3$. An option is frequent if it has a support of 3 or more, that is, it is found in at least three different DSML projects. This results in five frequent option sets in our example (see also the greyed rectangles in Figure 7): $\{d1, d2\}$, $\{d1, d2, d3\}$, $\{d1, d2, d4\}$, $\{d1, d2, d6\}$, and $\{d1, d2, d3, d4\}$.

- *Frequent-closed option sets* [16]: An option set is said to be *closed* if it is frequent and if none of its proper option supersets has a support equal to or less than the support of this option set. In our example, we find four option sets out of the five frequent ones in this condition: $\{d1, d2\}$, $\{d1, d2, d4\}$, $\{d1, d2, d6\}$, and $\{d1, d2, d3, d4\}$ (see the nodes marked "c" in Figure 7). The option set $\{d1, d2, d3\}$ is not closed because its $\{d1, d2, d3, d4\}$ has the same support of 5.

  In other words, an option set is closed if no proper option superset containing a given option set is contained by the observed option sets, in which the option set is contained, in the collection. This can be the case under three conditions important to us:

  1. If a frequent option set corresponds to at least one observed option set as-is: An example is $\{d1, d2\}$ for SECRDW. In this case, none of its supersets (e.g., $\{d1, d2, d4\}$) can naturally be part of this observed set. Conversely, $\{d1, d2, d3\}$ is not closed because it does not appear as-is in the collection, only as a subset of $\{d1, d2, d3, d4\}$ (e.g., for EIS and UACL) and of $\{d1, d2, d3, d4, d6\}$ (i.e., for BIT and UML4SOA).
  2. If a frequent option set represents the least-common frequent subset for (some of) its proper option supersets as contained in the observed option sets of the collection: Consider extending the example based on Table 25 and Figure 7. In its given setting, $\{d1, d2, d3\}$ is not closed (see above). If *one* DSML is added which is described by the observed option set $\{d1, d2, d3, d5, d6\}$, then $\{d1, d2, d3\}$ would become closed because it qualifies as the least-common subset contained in both the five observed option sets containing $\{d1, d2, d3, d4\}$ and the newly added one. Technically, this would be reflected in an increased support of $\{d1, d2, d3\}$ (6), therefore, surpassing the support of $\{d1, d2, d3, d4\}$ (5).
  3. If both conditions 1) and 2) above hold for a frequent option set: This is the case for $\{d1, d2\}$. First, it appears as-is as an observed option set (SECRDW; see Table 25). Second, it turns out to be the least-common frequent subset for the observed option sets containing d3/d4 *and* d6 (i.e., BIT and UML4SOA).

  In summary: All closed option sets are frequent ones. The subset of closed option sets can potentially be smaller than the number of total frequent subsets (i.e., four of five option sets in our example), but this is not necessarily the case. Non-closed frequent option sets are subsets of one or several closed option supersets.

- *Maximal-frequent option sets* [16]: The set of frequent options sets is a subset of the design space which represents minimum support (or adoption of certain options) in the studied DSML projects. This subset, however, contains redundant information. For example, $\{d1, d2\}$ is included by all other four frequent option sets which are proper supersets of the former. Any of these supersets represent the condition of $\{d1, d2\}$ being frequent.

  A frequent option set is called *maximal* if none of its proper subsets is frequent (i.e., has equal to or more than the minimum support). This notion is suitable for removing the redundancy by upward containment between frequent option sets and to establish a potentially smaller subset of characteristic frequent option sets which is capable of representing all other frequent option sets.

  In our example, we find two maximal-frequent option sets: $\{d1, d2, d6\}$ and $\{d1, d2, d3, d4\}$ (see also the nodes marked "m" in Figure 7). The remainder of three frequent option sets are all subsets of these two option sets. The maximal subset of the set of frequent option sets, therefore, exhibits those frequent option sets with maximum cardinality (three and four decision points, respectively). From a design-space perspective, a maximal option set reflects a frequent configuration of a maximal number of decision options taken jointly—besides summarizing the entire sub-space of frequent option sets. Applied to the ten DSMLs, we can therefore state that a critical number of DSMLs take decisions at up to three and four decision points, but never at five or six decision points. In addition, based on this sample, we could summarize that most frequently DSMLs either take decisions at decision points d3 and/or d4 or—mutually exclusive—at d6, if any decision beyond d1 and d2 at all.

All maximal option sets are closed. Therefore, the set of maximal option sets is a subset of the closed subset of option sets.

- *Free frequent option sets* [9, 16]: An option set is considered a free option set (a.k.a. generator) if it is the minimal subset (i.e., the smallest in terms of options contained) among all the option subsets appearing in an observed option set. It is minimal in the sense that there are no smaller option sets (i.e., the proper subsets of the free set) which appear as-is in an observed option set. A free option set (generator) is called frequent if it has at least minimum support.

  Of particular interest to us are the free item sets which form the closed frequent item sets as found for the selected DSMLs. As stated above, the closed frequent item sets serve as a compact representation of the entire observed frequent design space (i.e., all frequent option sets can be expressed as subsets of the closed option sets). However, as the largest frequent building blocks (in terms of options contained) found in observed DSMLs, they are not as selective when characterizing observed option sets. For instance, to find the observed option sets containing $d3$, we take the closed set $\{d1, d2, d3, d4\}$ marked in Figure 7, and match it against the ten observed option sets. This will yield five option sets. This result, however, contains noise because the five option sets are also those containing $d3$ jointly with $d4$.

  According to the above definition, $\{d1, d2, d3\}$ is found to be a frequent generator of the closed set $\{d1, d2, d3, d4\}$ in the sense that it is capable of matching all observed option sets while being of smaller size in terms of options. By being smaller, it is more informative because it can be more easily combined, for example, with other smaller closed or generator sets (e.g., $\{d1, d2, d4\}$ in Figure 7) to describe the observed design space. Combining the comparatively larger closed sets as descriptors suffers from more redundancy, such as $\{d1, d2, d4\}$ and $\{d1, d2, d3, d4\}$ differing only by one option.

  It also follows from the above definition that a free option set or generator cannot correspond to an observed option set; it is a building block only.

Decision-option Sets. We performed an analysis as introduced in Section 8 to describe frequent patterns in the data base of 80 observed option sets. To run the following computations, we processed the data set of 80 option sets to exclude the no-option codes, so that absence of all option codes at a given decision point indicates absence of a decision. We, therefore, considered 24 different option codes, yielding a potential design space of $2^{24} - 1$ option sets. Under our working assumption of minimally included decision options, this is constrained to a space of $15 * 15 * 2^{16}$ potential option sets. In the following, we consider an option set or option-set pattern to be frequent if it is found for three or more DSMLs.

Unique option sets are non-duplicated observed option sets (see Section 8). The set of 80 observed option sets contains 53 unique (non-duplicated) option sets. Table 27 summarizes important observations on this option subset:

- 39 out of 53 uniquely observed option sets have exactly one occurrence, i.e., they represent exactly one DSML.
- The remaining 14 unique option sets have two or more occurrences. However, not more than five DSMLs share exactly one observed option set (see Table 27).
- The 53 unique option sets contain at minimum two options (see, e.g., SECRDW) and at maximum ten options (out of 24 possible ones; see, e.g., UML4SOA).
- The 14 unique option sets which are found shared between two or more DSMLs contain at minimum three (see, e.g., UML4SPM) and at maximum seven options (see, e.g., Aspect-SM).

In the space of $15 * 15 * 2^{16}$ potential sets of options, we found 188 frequent options sets; that is option sets which are contained partly or fully in more than three observed option sets, each specific to one DSML. These 188 frequent option sets are all contained as subsets of 40 closed and

Table 26: Overview of the option-set constructs (option subsets) considered for the analysing frequent patterns in the selected DSMLs. Each construct is defined as a set of restrictions (e.g., closedness, maximality, freeness) over the space of (frequent) options obtained from existing DSMLs.

	Option (sub-)set	Description	Restrictions: *An option set which ...*
I	Largest, highly shared option-subset	A largest (i.e., of maximal size) design fragment which is frequently observed and has a relatively high support.	• is closed *and* • is *not* maximal *and* • is *not* total
II	Largest, lowly shared option-subset	A largest (i.e., of maximal size) design fragment which is frequently observed and has a relatively low support.	• is maximal *and* • is *not* total
III	Prototype option-sets with frequent extensions	An option set which represents a highest-common, largest option-subset which was also frequently found in complete DSML design. Because for this option set frequently occurring supersets exist, this prototype design is often extended (*evolutionary* prototype) by adding other (frequently observed) options.	• is closed *and* • is *not* maximal *and* • is total *and* • is of frequency greater than or equal to minimum support
IV	Prototype option-sets with infrequent extensions	An option set which represents a lowest-common, largest option-subset which was also frequently found in complete DSML design. Because for this option set no frequent supersets exist, this prototype option-set is often employed as is. Extensions that add options to this (*evolutionary*) prototype are rarely observed.	• is maximal *and* • is total *and* • is of frequency greater than or equal to minimum support
V	Smallest common option-subset	A frequent option set which is also the smallest (i.e., of minimal size) recurring design fragment in the observed designs (e.g., a prototype option-set) and in design fragments which contain this option subset. We distinguish between two kinds of smallest common option-subsets: (1) Option subsets specific to one decision point; (2) Option subsets specific to two or more decision points.	• is free *and* • is frequent *and* • is contained by at least one largest option-subset (I, II, III, IV) *and* • no containing largest option-subsets (I, II, III, IV) has greater support

20 maximal option sets (see Section 8). 23 frequent option sets represent at least one DSML directly as an observed option set, 165 of the frequent option sets are only subsets of observed option sets which are shared between DSMLs.

We found that three options (O3.2, O3.3, and O5.4) are not contained by any of the option sets of the 80 DSMLs. Considering minimum support by three DSMLs, eight more options are found missing from the 188 frequent option sets, that is, they are not featured by any of the 188 frequent option sets: O2.4, O4.3, O4.4, O4.5, O5.1, O5.2, O5.3, and O6.4. In other words, they are not found in any expressible option set over three or more DSMLs. From a total of 24 options, 16 are frequently found (in the sense of being contained in option sets of above minimum support).

We found the following kinds of option sets, as defined in Section 8 (see I–V in Table 26). In our pool of 80 DSMLs, we found two proper option subsets which qualify as smallest recurring option-subsets, with the options in each option-subset stemming from one decision point (see Table 28). The two smallest common option-subsets *specific to one decision point* have a size of two options each and relate to decision points on constraining the language model (D3) and on platform integration (D6). For all the other four decision points, we could not find any smallest option subsets. The option subset $\{3.1, 3.4\}$ serves as the smallest common descriptor regarding language-model constraints (D3) in 18 DSML projects. It reflects that in these projects language-model constraints are consistently defined using both a constraint-expression language as well as auxiliary or complimentary textual constraint definitions in natural language. As for platform integration (D6), the smallest common descriptor shared by three DSMLs is the combination of M2T generator templates (O6.2) and M2M transformations (O6.5). This option subset reflects that there is a two-level model transformation chain being employed in the three approaches (PIM-PIM-PSM): First, platform-independent models (PIM) are transformed into another PIM representation which is then transformed into a structured textual, platform-specific (PSM) representation. For example, in UML2Alloy [3], extended UML class models (PIM) are processed into models of an Alloy metamodel (PIM) which are finally turned into textual Alloy definitions accepted by an Alloy model checker (PSM).

We extracted seven different smallest option subsets which involve minimum two options from two different decision points (see Table 29). The smallest common option-subsets *specific to two* *or more decision points* found are specific to options at decision points D1, D2, D3, D4, and D6. Four out of seven option subsets contain two to three options between language-model formalization (D2) and language-model constraining (D3). The three option subsets $\{2.2, 3.4\}$, $\{2.2, 3.1\}$, and $\{2.2, 3.1, 3.4\}$ represent that applying one or several UML profiles (O2.2) is associated with defining language-model constraints either textually only (O3.4; 30 DSMLs), or using a constraint-expression language only (O3.1; 26), or both (13). On the contrary, metamodel extensions (O2.3) are found frequently combined with both constraint-definition strategies $\{2.3, 3.1, 3.4\}$, rather than either of the two exclusively. Metamodel extensions (O2.3) are also commonly—though at the minimum-support level—applied together with diagrammatic syntax extensions (O4.2; i.e., introducing new UML diagram notational elements) and M2M transformation (O6.5). Finally, 22 of the 80 DSMLs adopt UML profiles (O2.2) for realizing a formally defined language model (O1.4) and share this proper option subset as a smallest descriptor. The 80 DSMLs contain seven distinct prototype option-sets (see Tables 30 and 31), that is, option sets which are frequent and describe entire DSML designs, with and without extensions. Six prototype option-sets come *with frequent extensions* (see Table 30). For example, the option set of UML-PMS [41] describes five observed and complete DSML designs (frequency) while it is found as a large option subset in 25 other DSMLs ($support - frequency$) in an extended form. Five prototype option-sets involve UML profiles only (O2.2), just one frequently found prototype option-set builds on metamodel extensions only (O2.3; e.g., UML4SPM [11]). All six designs involve at least one concrete-syntax decision option (see also Figure 8, indicating D4 as mandatory). The only platform-integration option found adopted in three prototype option-sets (and 12 more extensions of it) are M2T generator templates (O6.2).

A seventh prototype option-set was found which comes *with infrequent extensions*, that is, while it is realized by three DSMLs (i.e., UML-AOF, PredefinedConstraints, and UML-PMS; see Table 31) it is found extended twice ($support - frequency$). The option subset reflects a widely documented and recommended—but not necessarily frequently used—way of creating a UML-based DSML using UML profiles, by two-option strategies to define the language model (O1.1, O1.4) and the language-model constraints (O3.1, O3.4), respectively. The concrete-syntax choices (O4.1, O4.6) are inherent to UML profile usage.

The seven prototype option-sets which are realized as-is for 24 out of 80 DSMLs and are observed with extensions for up to 25 DSMLs are summarized in terms of their commonalities and differences as a feature diagram in Figure 8. The seven designs are combinations of nine options. By looking at these nine options and their characteristic combinations (see Tables 30 and 31) at least 30% (24/80) of the 80 DSMLs can be described in their entirety (prototype option-set).

The observed design space of 80 DSMLs contains 17 largest common option-subsets (see Tables 32 and 33). Eight are of comparatively high (4–10; see Table 32) and nine are of comparatively low support (3–5; see Table 33). Ten subsets result in profile-only designs (O2.2), six in DSMLs based on metamodel extensions only (O2.3). A single option subset is found for building mixed designs (3 designs; see Table 33).

In total, these 17 largest option subsets characterize 75 of the 80 DSML designs. The option subsets are made up by twelve different options specific to five different decision points (see Figure 9). Each of the 17 option subsets involves at least one decision option on language-model constraining (D3), on the concrete syntax (D4), or on platform integration (D6). The feature model of prototype option-sets in Figure 8 is a specialization of the more general feature model describing the observed largest option subsets, adding two major restrictions: First, a prototype option-set must involve a concrete-syntax decision (D4 becomes mandatory) which is not the case for a largest option subset. Second, frequently observed designs do not take decisions on platform integration at all (D6) or adopt only M2T generator templates (O6.2). The largest option subsets, which describe a larger variety of observed designs than the prototype option-sets alone, reflect all four platform-integration options (O6.1–O6.3, O6.5) which were found adopted in the 80 DSMLs.

Table 27: Overview of the degree of sharing (frequency class) of a unique option set, count of unique option sets found per frequency class, and the number of options contained by a unique option set (cardinalities); as observed for the 53 unique option sets.

	Freq. class	Count	Cardinality range (min–max)	Unique option sets (ex.; DSML)
	5	2	5–5	{1.1, 2.2, 3.4, 4.1, 4.6}; UML-PMS
	4	2	3–5	{1.1, 2.3, 4.6}; UML4SPM
	3	3	5–7	{1.1, 2.2, 4.1, 4.6, 6.2}; WS-CM
	2	7	4–7	{1.1, 1.4, 2.2, 3.1, 4.1, 4.6, 6.2}; Aspect-SM
Σ		14	3–7	
	1	39	2–10	{1.1, 2.3}; SECRDW {1.1, 2.2, 2.3, 3.1, 4.1, 4.2, 4.6, 6.1, 6.3, 6.5}; UML4SOA
Σ		53	2–10	

Table 28: Overview of the two smallest common option-subsets *specific to one decision point* (ordered by decreasing absolute support).

Decision point	Option subset	Support (abs.)
D3	{3.1, 3.4}	18
D6	{6.2, 6.5}	3

Table 29: Overview of the seven smallest common option-subsets *specific to one decision point* (ordered by decreasing absolute support).

Decision points	Option subset	Support (abs.)
D2, D3	{2.2, 3.4}	30
D2, D3	{2.2, 3.1}	26
D1, D2	{1.4, 2.2}	22
D2, D3	{2.2, 3.1, 3.4}	13
D2, D3	{2.3, 3.1, 3.4}	5
D2, D4	{2.3, 4.2}	3
D2, D6	{2.3, 6.5}	3

Table 30: Overview of the six prototype option-sets which are frequently extended (ordered by decreasing absolute support). A corresponding feature model which represents the commonalities and differences between all prototype option-sets is shown in Figure 8.

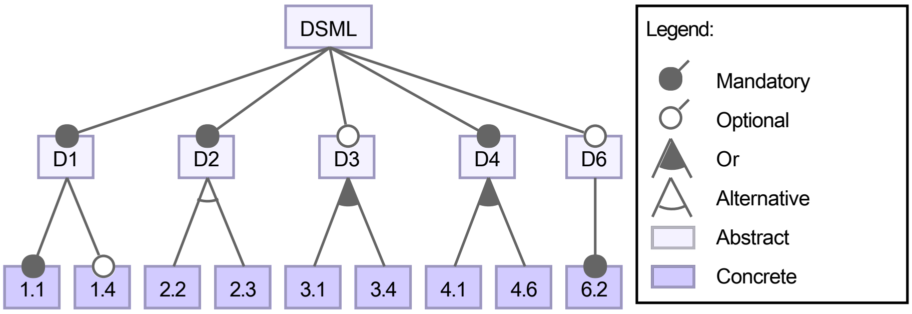
Prototype	Support (abs.)	Frequency (abs.)	DSMLs (ex.)
{1.1, 2.2, 3.4, 4.1, 4.6}	30	5	UML-AOF, PredefinedConstraints, UML-PMS
{1.1, 2.2, 3.1, 4.1, 4.6}	26	4	REMP, CUP, UML4PF
{1.1, 1.4, 2.2, 4.1, 4.6}	22	5	SPArch, MoDePeMART, RichService
{1.1, 2.2, 4.1, 4.6, 6.2}	15	3	DPL, WCAAUML, WS-CM
{1.1, 2.2, 3.1, 3.4, 4.1, 4.6}	13	3	ArchitecturalPrimitives, SHP, C2style
{1.1, 2.3, 4.6}	10	4	UML2Ext, UML4SPM, MDATC

Table 31: Overview of one prototype option-set which is infrequently extended. A corresponding feature model which represents the commonalities and differences between all prototype option-sets is shown in Figure 8.

Prototype	Support (abs.)	Frequency (abs.)	DSMLs (ex.)
{1.1, 1.4, 2.2, 3.1, 3.4, 4.1, 4.6}	5	3	UACL, SafeUML, IEC61508

Table 32: Overview of eight largest option subsets of *relatively high support* (i.e., frequently supported; ordered by decreasing absolute support).

Prototype	Support (abs.)
{1.1, 1.4, 2.2, 3.1, 4.1, 4.6}	10
{1.1, 2.3, 3.1}	6
{1.1, 2.3, 3.4}	6
{1.1, 2.2, 4.1, 4.6, 6.5}	6
{1.1, 2.2, 4.1, 4.6, 6.3}	5
{1.1, 2.2, 3.1, 4.1, 4.6, 6.5}	5
{1.1, 2.2, 4.1, 4.6, 6.1, 6.5}	4
{1.1, 1.4, 2.2, 4.1, 4.6, 6.2}	4



2.2 ⇒ 4.1 ∧ 4.6

2.3 ⇒ 4.6 ∧ ¬ 4.1

6.2 ⇒ 2.2

3.1 ∨ 3.4 ⇒ 2.2

1.4 ⇒ 2.2

6.2 ⇒ ¬ D3

1.4 ⇒ ¬ D6

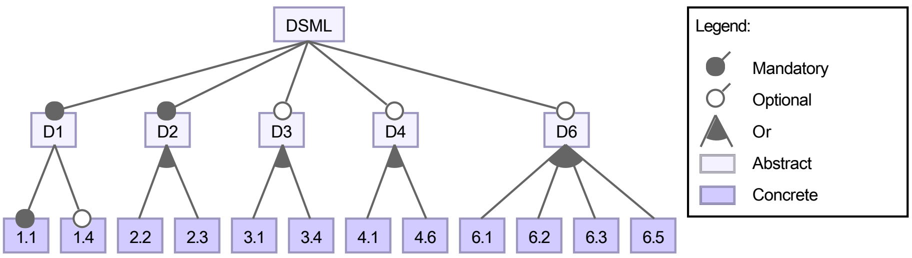
2.2 ∧ ¬ 1.4 ⇒ D3 ∨ D6

1.4 ⇒ ¬ D3 ∨ 3.1 ∧ 3.4

Figure 8: A feature model which represents the prototype option-sets found in the pool of 80 DSMLs; that is, each configuration of the feature space represents one of the seven observed prototype option-sets listed in Tables 30 and 31.

Table 33: Overview of nine largest option subsets of *relatively low support* (i.e., infrequently supported; ordered by decreasing absolute support).

Prototype	Support (abs.)
{1.1, 2.3, 3.1, 3.4}	5
{1.1, 2.3, 6.5}	3
{1.1, 2.3, 3.1, 4.6}	3
{1.1, 2.3, 3.4, 4.6}	3
{1.1, 2.2, 2.3, 4.1, 4.6}	3
{1.1, 2.2, 4.1, 4.6, 6.3, 6.5}	3
{1.1, 1.4, 2.2, 4.1, 4.6, 6.5}	3
{1.1, 2.2, 3.4, 4.1, 4.6, 6.5}	3
{1.1, 2.2, 3.1, 4.1, 4.6, 6.1, 6.5}	3



2.2 ⇒ 4.1 ∧ 4.6

6.2 ⇒ 2.2 ∧ 1.4

1.4 ⇒ 2.2

6.2 ⇒ ¬ D3

2.2 ∧ ¬ 2.3 ∧ ¬ 1.4 ⇒ D4 ∧ D6

6.1 ⇒ 6.5 ∧ ¬ 1.4 ∧ (3.1 ∨ ¬ D3)

6.2 ⇒ ¬ (6.1 ∨ 6.3 ∨ 6.5)

6.3 ⇒ ¬ (6.1 ∨ 6.2) ∧ ¬ 1.4 ∧ ¬ D3 ∧ ¬ 2.3

2.3 ∧ ¬ 2.2 ⇒ (D3 ∨ 6.5 ∧ ¬ 6.1) ∧ ¬ (D3 ∧ D6) ∧ ¬ (D4 ∧ D6)

1.4 ⇒ 3.1 ∧ ¬ D6 ∨ (6.5 ∨ 6.2) ∧ ¬ D3

2.3 ∧ ¬ 2.2 ⇒ ¬ D4 ∨ 4.6 ∧ ¬ 4.1

2.2 ∧ 2.3 ⇒ ¬ D3 ∧ ¬ D6

3.1 ∧ 3.4 ⇒ ¬ D4

Figure 9: A feature model which represents the largest option subsets found in the pool of 80 DSMLs; that is, each configuration of the feature space represents one of the 17 observed option subsets listed in Tables 32 and 33.

References

[1] S. Ali, L. C. Briand, and H. Hemmati. Modeling robustness behavior using aspect-oriented modeling to support robustness testing of industrial systems. *Softw. Syst. Model.*, 11(4):633–670, 2012.

[2] E. L. Alves, P. D. Machado, and F. Ramalho. Automatic generation of built-in contract test drivers. *Softw. Syst. Model.*, pages 1–25, 2012.

[3] K. Anastasakis, B. Bordbar, G. Georg, and I. Ray. On challenges of model transformation from UML to Alloy. *Softw. Syst. Model.*, 9(1):69–86, 2010.

[4] C. Atkinson and T. Kühne. Concepts for comparing modeling tool architectures. In L. Briand and C. Williams, editors, *Model Driven Eng. Languages Syst.*, volume 3713 of *LNCS*, pages 398–413. Springer, 2005.

[5] C. Atkinson and T. Kühne. A tour of language customization concepts. *Adv. Comput.*, 70:105–161, 2007.

[6] C. Atkinson, T. Kühne, and B. Henderson-Sellers. Systematic stereotype usage. *Softw. Syst. Model.*, 2(3):153–163, 2003.

[7] Australian Research Council. Excellence in Research for Australia (ERA) journal list. Available at `http://www.arc.gov.au/era/era_2012/era_journal_list.htm`. Last accessed: Oct 3, 2014, 2012.

[8] D. Basin, J. Doser, and T. Lodderstedt. Model driven security: From UML models to access control infrastructures. *ACM T. Softw. Eng. Meth.*, 15(1):39–91, 2006.

[9] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Proc. 1st Int. Conf. Computational Logic (CL 2000)*, volume 1861 of *LNCS*, pages 972–986. Springer, 2000.

[10] R. Behjati, T. Yue, S. Nejati, L. Briand, and B. Selic. Extending SysML with AADL concepts for comprehensive system architecture modeling. In *Proc. 7th European Conf. Modelling Found. Applicat. (ECMFA'11)*, volume 6698 of *LNCS*, pages 236–252. Springer, 2011.

[11] R. Bendraou, M.-P. Gervais, and X. Blanc. UML4SPM: A UML2.0-based metamodel for software process modelling. In *Proc. 8th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'05)*, volume 3713 of *LNCS*, pages 17–38. Springer, 2005.

[12] K. Berkenkötter and U. Hannemann. Modeling the railway control domain rigorously with a UML 2.0 profile. In *Proc. 25th Int. Conf. Comput. Safety, Reliability, Security (SAFECOMP'06)*, volume 4166 of *LNCS*, pages 398–411. Springer, 2006.

[13] S. Bernardi, F. Flammini, S. Marrone, J. Merseguer, C. Papa, and V. Vittorini. Model-driven availability evaluation of railway control systems. In *Proc. 30th Int. Conf. Comput. Safety, Reliability, Security (SAFECOMP'11)*, volume 6894 of *LNCS*, pages 15–28. Springer, 2011.

[14] S. Bernardi, J. Merseguer, and D. C. Petriu. A dependability profile within MARTE. *Softw. Syst. Model.*, 10(3):313–336, 2011.

[15] A. Blouin, B. Combemale, B. Baudry, and O. Beaudoux. Kompren: modeling and generating model slicers. *Softw. Syst. Model.*, 14(1):321–337, 2015.

[16] C. Borgelt. Frequent item set mining. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 2(6):437–456, Nov. 2012.

[17] M. Bošković and W. Hasselbring. Model driven performance measurement and assessment with MoDePeMART. In A. Schürr and B. Selic, editors, *Model Driven Eng. Languages Syst.*, volume 5795 of *LNCS*, pages 62–76. Springer, 2009.

[18] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.*, 80(4): 571–583, 2007.

[19] P. Brosch, M. Seidl, M. Wimmer, and G. Kappel. Conflict visualization for evolving UML models. *J. Object Technology*, 11(3):2:1–30, Oct. 2012.

[20] J. Bruck and K. Hussey. Customizing UML: Which technique is right for you? Available at: `http://www.eclipse.org/modeling/mdt/uml2/docs/articles/Customizing_UML2_Which_Technique_is_Right_For_You/article.html`. Last accessed: Feb 2, 2015, 2008. IBM.

[21] A. D. Brucker, J. Doser, and B. Wolff. Semantic issues of OCL: Past, present, and future. In *Proc. 6th Workshop OCL for (Meta-)Models in Multiple Application Domains*, pages 213–228, 2006.

[22] X. Burgués, X. Franch, and J. M. Ribó. Improving the accuracy of UML metamodel extensions by introducing induced associations. *Softw. Syst. Model.*, 7(3):361–379, 2008.

[23] J. Cabot. Ambiguity issues in OCL postconditions. In *Proc. 6th Workshop OCL for (Meta-)Models in Multiple Application Domains*, pages 194–204, 2006.

[24] A. Cicchetti, D. D. Ruscio, and A. Pierantonio. A metamodel independent approach to difference representation. *J. Object Technology*, 6(9):165–185, Oct. 2007. TOOLS EUROPE 2007 – Objects, Models, Components, Patterns.

[25] P. J. Clemente, J. Hernández, J. M. Conejero, and G. Ortiz. Managing crosscutting concerns in component based systems using a model driven development approach. *J. Syst. Softw.*, 84(6): 1032–1053, 2011.

[26] J. Cohen. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.*, 20(1):37–46, 1960.

[27] A. M. R. da Cruz and J. a. P. Faria. A metamodel-based approach for automatic user interface generation. In *Proc. 13th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'10)*, volume 6394 of *LNCS*, pages 256–270. Springer, 2010.

[28] J. Dingel, Z. Diskin, and A. Zito. Understanding and improving UML package merge. *Softw. Syst. Model.*, 7(4):443–467, 2008.

[29] J. Dong, S. Yang, and K. Zhang. Visualizing design patterns in their applications and compositions. *IEEE T. Softw. Eng.*, 33(7):433–453, 2007.

[30] R. Ellner, S. Al-Hilank, J. Drexler, M. Jung, D. Kips, and M. Philippsen. eSPEM – a SPEM extension for enactable behavior modeling. In *Proc. 6th European Conf. Modelling Found. Applicat. (ECMFA'10)*, volume 6138 of *LNCS*, pages 116–131. Springer, 2010.

[31] V. Englebert and P. Heymans. Towards more extensible metaCASE tools. In J. Krogstie, A. Opdahl, and G. Sindre, editors, *Advanced Information Systems Engineering*, volume 4495 of *LNCS*, pages 454–468. Springer, 2007.

[32] V. Ermagan and I. H. Krüger. A UML2 profile for service modeling. In *Proc. 10th Int. Conf. Model-Driven Eng. Languages Syst. (MoDELS'07)*, volume 4735 of *LNCS*, pages 360–374. Springer, 2007.

[33] J. Evermann. A meta-level specification and profile for AspectJ in UML. *J. Object Technology*, 6 (7):27–49, Aug. 2007.

[34] I. Feinerer, K. Hornik, and D. Meyer. Text mining infrastructure in R. *J. Statistical Softw.*, 25(5): 1–54, 2008.

[35] E. Fernández-Medina, J. Trujillo, R. Villarroel, and M. Piattini. Access control and audit model for the multidimensional modeling of data warehouses. *Decis. Support Syst.*, 42(3):1270–1289, Dec. 2006.

[36] E. Filtz. Systematic literature review and evaluation of DSML-design decisions. Bachelor thesis, Vienna University of Economics and Business, 2013.

[37] T. Fink, M. Koch, and K. Pauls. An MDA approach to access control specifications using MOF and UML profiles. *Electron. Notes in Theor. Comput. Sci.*, 142:161–179, 2006.

[38] G. Giachetti, B. Marín, and O. Pastor. Using UML as a domain-specific modeling language: A proposal for automatic generation of UML profiles. In *Proc. 21st Int. Conf. Advanced Information Systems Engineering (CAiSE'09)*, volume 5565 of *LNCS*, pages 110–124. Springer, 2009.

[39] S. Gilmore, L. Gönczy, N. Koch, P. Mayer, M. Tribastone, and D. Varró. Non-functional properties in the model-driven development of service-oriented systems. *Softw. Syst. Model.*, 10(3):287–311, 2011.

[40] C. Gonzalez-Perez and B. Henderson-Sellers. *Metamodelling for Software Engineering*. John Wiley & Sons, 2008.

[41] V. Grassi. Performance analysis of mobile systems. In M. Bernardo and A. Bogliolo, editors, *Formal Methods for Mobile Computing*, volume 3465 of *LNCS*, pages 107–154. Springer, 2005.

[42] K. L. Gwet. *Handbook of Inter-Rater Reliability*. Advanced Analytics, LLC, 3rd edition, 2012.

[43] M. Hafner and R. Breu. *Security Engineering for Service-Oriented Architectures*. Springer, 2009.

[44] M. Hahsler, B. Grün, and K. Hornik. arules—a computational environment for mining association rules and frequent item sets. *J. Statistical Softw.*, 14(15):1–25, 2005.

[45] D. Hästbacka, T. Vepsäläinen, and S. Kuikka. Model-driven development of industrial process control applications. *J. Syst. Softw.*, 84(7):1100–1113, 2011.

[46] D. Hatebur and M. Heisel. A UML profile for requirements analysis of dependable software. In *Proc. 29th Int. Conf. Comput. Safety, Reliability, Security (SAFECOMP'10)*, volume 6351 of *LNCS*, pages 317–331. Springer, 2010.

[47] D. Hatebur and M. Heisel. Making pattern- and model-based software development more rigorous. In J. S. Dong and H. Zhu, editors, *Formal Methods and Softw. Eng.*, volume 6447 of *LNCS*, pages 253–269. Springer, 2010.

[48] B. Henderson-Sellers and C. Gonzalez-Perez. Uses and abuses of the stereotype mechanism in UML 1.x and 2.0. In *Proc. 9th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'06)*, volume 4199 of *LNCS*, pages 16–26. Springer, 2006.

[49] B. Hoisl and S. Sobernig. Integrity and confidentiality annotations for service interfaces in SoaML models. In *Proc. Int. Workshop Security Aspects of Process-aware Information Systems (SAPAIS'11)*, pages 673–679. IEEE, 2011.

[50] B. Hoisl and M. Strembeck. Modeling support for confidentiality and integrity of object flows in activity models. In *Proc. 14th Int. Conf. Business Information Systems*, volume 97 of *LNBIP*, pages 278–289. Springer, 2011.

[51] B. Hoisl and M. Strembeck. A UML extension for the model-driven specification of audit rules. In *Proc. 2nd Int. Workshop Information Systems Security Engineering (WISSE'12)*, volume 112 of *LNBIP*, pages 16–30. Springer, 2012.

[52] B. Hoisl, S. Sobernig, S. Schefer-Wenzl, M. Strembeck, and A. Baumgrass. Design decisions for UML and MOF based domain-specific language models: Some lessons learned. In *Proc. 2nd Workshop Process-based approaches for Model-Driven Engineering (PMDE'12)*, pages 303–314, 2012.

[53] B. Hoisl, S. Sobernig, S. Schefer-Wenzl, M. Strembeck, and A. Baumgrass. A catalog of reusable design decisions for developing UML- and MOF-based domain-specific modeling languages. Available at: `http://epub.wu.ac.at/3578/`, 2012. Technical Reports / Institute for Information Systems and New Media (WU Vienna), 2012/01.

[54] B. Hoisl, S. Sobernig, and M. Strembeck. A catalog of reusable design decisions for developing UML/MOF-based domain-specific modeling languages. Available at: `http://epub.wu.ac.at/4312/`, 2014. Technical Reports / Institute for Information Systems and New Media (WU Vienna), 2014/03.

[55] B. Hoisl, S. Sobernig, and M. Strembeck. Modeling and enforcing secure object flows in process-driven SOAs: An integrated model-driven approach. *Softw. Syst. Model.*, 13(2):513–548, 2014.

[56] M. Huhn and A. Zechner. Analysing dependability case arguments using quality models. In *Proc. 28th Int. Conf. Comput. Safety, Reliability, Security (SAFECOMP'09)*, volume 5775 of *LNCS*, pages 118–131. Springer, 2009.

[57] W. Hummer, P. Gaubatz, M. Strembeck, U. Zdun, and S. Dustdar. Enforcement of entailment constraints in distributed service-based business processes. *Inform. Softw. Tech.*, 55(11):1884–1903, 2013.

[58] E. Jackson and J. Sztipanovits. Formalizing the structural semantics of domain-specific modeling languages. *Softw. Syst. Model.*, 8(4):451–478, 2009.

[59] V. Jain, A. Kumar, and P. R. Panda. A SysML profile for development and early validation of TLM 2.0 models. In *7th Proc. European Conf. Modelling Found. Applicat. (ECMFA'11)*, volume 6698 of *LNCS*, pages 299–311. Springer, 2011.

[60] S. Jalali and C. Wohlin. Systematic literature studies: database searches vs. backward snowballing. In *Proc. ACM-IEEE Int. Symposium Empirical Software Eng. Measurement (ESEM'12)*, pages 29–38. ACM, 2012.

[61] H. Jegadeesan and S. Balasubramaniam. An MOF2-based services metamodel. *J. Object Technology*, 7(8):71–96, 2008.

[62] J. Jürjens. *Secure Systems Development with UML*. Springer, 2005.

[63] S. Kallel, M. H. Kacem, and M. Jmaiel. Modeling and enforcing invariants of dynamic software architectures. *Softw. Syst. Model.*, 11(1):127–149, 2012.

[64] B. Kitchenham. Procedures for performing systematic reviews. Joint Technical Report (Keele University Technical Report, NICTA Technical Report) TR/SE-0401, 0400011T.1, Keele University & National ICT Ltd., July 2004.

[65] B. Kitchenham, E. Mendes, and G. Travassos. Protocol for systematic review of within- and cross-company estimation models. Review Protocol, Version 14, last accessed: Feb 2, 2015, 2006.

[66] L. L. Kupper and K. B. Hafner. On assessing interrater agreement for multiple attribute responses. *Biometrics*, 45(3):957–967, 1989.

[67] F. Lagarde, F. Terrier, C. André, and S. Gérard. Constraints modeling for (profiled) UML models. In *Proc. 3rd European Conf. Model Driven Architecture – Foundations and Applications*, volume 4530 of *LNCS*, pages 130–143. Springer, 2007.

[68] F. Lagarde, E. Huáscar, F. Terrier, C. André, and S. Gérard. Leveraging patterns on domain models to improve UML profile definition. In *Proc. 11th Int. Conf. Fundamental Approaches to Softw. Eng. (FASE'08)*, volume 4961 of *LNCS*, pages 116–130. Springer, 2008.

[69] K. Lind and R. Heldal. A model-based and automated approach to size estimation of embedded software components. In *Proc. 14th Int. Conf. Model Driven Eng. Languages Syst.*, pages 334–348. Springer, 2011.

[70] P. Mayer, A. Schroeder, and N. Koch. MDD4SOA: Model-driven service orchestration. In *Proc. 12th Int. Conf. Enterprise Distributed Object Computing (EDOC'08)*, pages 203–212. IEEE, 2008.

[71] M. Mernik, J. Heering, and A. Sloane. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, 2005.

[72] Microsoft Corporation. Microsoft academic search journal list. Available at `http://academic.research.microsoft.com/?SearchDomain=2&SubDomain=4&entitytype=4`. Last accessed: Feb 2, 2015, 2013.

[73] M. Miguel, J. Briones, J. Silva, and A. Alonso. Integration of safety analysis in model-driven software development. *IET Softw.*, 2(3):260–280, 2008.

[74] N. Moreno, P. Fraternali, and A. Vallecillo. WebML modelling in UML. *IET Softw.*, 1(3):67–80, 2007.

[75] L. Nascimento, D. L. Viana, P. A. M. S. Neto, D. A. O. Martins, V. C. Garcia, and S. R. L. Meira. A systematic mapping study on domain-specific languages. In *Proc. 7th Int. Conf. Softw. Eng. Advances (ICSEA'12)*, pages 179–187. IARIA, 2012.

[76] F. Naumann and M. Herschel. *An Introduction to Duplicate Detection*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[77] M. Nikolaidou, N. Alexopoulou, A. Tsadimas, A. Dais, and D. Anagnostopoulos. Accommodating EIS UML 2.0 profile using a standard UML modeling tool. In *Proc. Int. Conf. Softw. Engineering Advances*, 2007.

[78] Object Management Group. OMG unified modeling language (OMG UML), superstructure. Available at: `http://www.omg.org/spec/UML`, 2011. Version 2.4.1, formal/2011-08-06.

[79] Object Management Group. OMG meta object facility (MOF) core specification. Available at: `http://www.omg.org/spec/MOF`, Apr. 2014. Version 2.4.2, formal/2014-04-03.

[80] R. Paige, J. Ostroff, and P. Brooke. Principles for modeling language design. *Inform. Softw. Tech.*, 42(10):665–675, 2000.

[81] R. Panesar-Walawege, M. Sabetzadeh, and L. Briand. A model-driven engineering approach to support the verification of compliance to safety standards. In *Proc. 22nd Int. Symposium Software Reliability Engineering*, pages 30–39, 2011.

[82] R. K. Panesar-Walawege, M. Sabetzadeh, and L. Briand. Using UML profiles for sector-specific tailoring of safety evidence information. In *Proc. 30th Int. Conf. Conceptual Modeling*, pages 362–378. Springer, 2011.

[83] J. Pardillo. A systematic review on the definition of UML profiles. In *Proc. 13th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'10)*, volume 6394 of *LNCS*, pages 407–422. Springer, 2010.

[84] J. Pardillo and C. Cachero. Domain-specific language modelling with UML profiles by decoupling abstract and concrete syntaxes. *J. Syst. Softw.*, 83(12):2591–2606, 2010.

[85] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *Proc. 12th Int. Conf. Evaluation and Assessment in Softw. Eng. (EASE'08)*, EASE'08, pages 68–77. British Computer Society, 2008.

[86] M. Pinto, L. Fuentes, and L. Fernández. Deriving detailed design models from an aspect-oriented ADL using MDD. *J. Syst. Softw.*, 85(3):525–545, 2012.

[87] S. Robert, S. Gérard, F. Terrier, and F. Lagarde. A lightweight approach for domain-specific modeling languages design. In *Proc. 35th EUROMICRO Conf. Softw. Eng. and Advanced Applications (SEAA'09)*, pages 155–161. IEEE, 2009.

[88] A. Rodríguez, E. Fernández-Medina, J. Trujillo, and M. Piattini. Secure business process model specification through a UML 2.0 activity diagram profile. *Decis. Support Syst.*, 51(3):446–465, 2011.

[89] J. Ryoo and H. Saiedian. AVDL: A highly adaptable architecture view description language. *J. Syst. Softw.*, 79(8):1180–1206, 2006.

[90] P. Salehi, A. Hamoud-Lhadj, P. Colombo, F. Khendek, and M. Toeroe. A UML-based domain specific modeling language for the availability management framework. In *Proc. 12th Int. Symposium High-Assurance Systems Engineering*, pages 35–44, 2010.

[91] S. Schefer. Consistency checks for duties in extended UML2 activity models. In *Proc. Int. Workshop Security Aspects of Process-aware Information Systems (SAPAIS'11)*, pages 680–685. IEEE, 2011.

[92] S. Schefer and M. Strembeck. Modeling support for delegating roles, tasks, and duties in a process-related rbac context. In *Proc. Int. Workshop Information Systems Security Engineering (WISSE'11)*, volume 83 of *LNBIP*, pages 660–667. Springer, 2011.

[93] S. Schefer-Wenzl and M. Strembeck. Modeling process-related duties with extended UML activity and interaction diagrams. *Electron. Commun. EASST*, 37, 2011.

[94] S. Schefer-Wenzl and M. Strembeck. An approach for consistent delegation in process-aware information systems. In *Proc. 15th Int. Conf. Business Information Systems (BIS'12)*, volume 117 of *LNBIP*, pages 60–71. Springer, 2012.

[95] S. Schefer-Wenzl and M. Strembeck. Modeling context-aware RBAC models for business processes in ubiquitous computing environments. In *Proc. 3rd Int. Conf. Mobile, Ubiquitous and Intelligent Computing*, pages 126–131. IEEE, 2012.

[96] S. Schefer-Wenzl and M. Strembeck. Modelling context-aware RBAC models for mobile business processes. *Int. J. Wireless Mobile Comput.*, 6(5):448–462, 2013.

[97] S. Schefer-Wenzl and M. Strembeck. Generic support for RBAC break-glass policies in process-aware information systems. In *Proc. 28th Annual ACM Symposium Applied Computing (SAC'13)*, pages 1441–1446. ACM, 2013.

[98] S. Schefer-Wenzl, S. Sobernig, and M. Strembeck. Evaluating a UML-based modeling framework for process-related security properties: A qualitative multi-method study. In *Proc. 21st European Conf. Information Systems (ECIS'13)*. AIS, 2013.

[99] Scimago Lab. SCImago journal & country rank. Available at http://www.scimagojr.com/journalrank.php. Last accessed: Feb 2, 2015, 2013.

[100] B. Selic. A systematic approach to domain-specific language design using UML. In *Proc. 10th IEEE Int. Symposium Object-Oriented Real-Time Distributed Computing (ISORC'07)*, pages 2–9. IEEE, 2007.

[101] B. Selic. The theory and practice of modeling language design for model-based software engineering – a personal perspective. In *Proc. Int. Summer School Generative Transformational Techniques Softw. Eng. (GTTSE'09)*, volume 6491 of *LNCS*, pages 290–321. Springer, 2009.

[102] S. Sen, N. Moha, B. Baudry, and J.-M. Jézéquel. Meta-model pruning. In *Proc. 12th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'12)*, volume 5795 of *LNCS*, pages 32–46. Springer, 2009.

[103] K. Siau and Y. Tian. A semiotic analysis of unified modeling language graphical notations. *Requir. Eng.*, 14(1):15–26, 2009.

[104] A. Silva, J. Saraiva, D. Ferreira, R. Silva, and C. Videira. Integration of RE and MDE paradigms: the ProjectIT approach and tools. *IET Softw.*, 1(6):294–314, 2007.

[105] C. Snook and M. Butler. UML-B: Formal modeling and design aided by UML. *ACM T. Softw. Eng. Meth.*, 15(1):92–122, 2006.

[106] S. Sobernig, B. Hoisl, and M. Strembeck. Protocol for a systematic literature review on design decisions for UML-based DSMLs. Available at: http://epub.wu.ac.at/4311/, 2014. Technical Reports / Institute for Information Systems and New Media (WU Vienna), 2014/02.

[107] E. Soler, J. Trujillo, E. Fernández-Medina, and M. Piattini. Building a secure star schema in data warehouses by an extension of the relational package from CWM. *Comp. Stand. Inter.*, 30(6): 341–350, 2008.

[108] S. S. Somé. A meta-model for textual use case description. *J. Object Technology*, 8(7):87–106, 2009.

[109] C. Song, E. Cho, and C. Kim. An integrated GUI-business component modeling method for the MDD- and MVC-based hierarchical designs. *Int. J. Softw. Eng. Know.*, 21(3):447–490, 2011.

[110] D. Spinellis. Notable design patterns for domain-specific languages. *J. Syst. Softw.*, 56(1):91–99, 2001.

[111] M. Staron and C. Wohlin. An industrial case study on the choice between language customization mechanisms. In *Proc. 7th Int. Conf. Product-Focused Software Process Improvement (PROFES'06)*, volume 4034 of *LNCS*, pages 177–191. Springer, 2006.

[112] M. Staron, L. Kuzniarz, and C. Wohlin. Empirical assessment of using stereotypes to improve comprehension of UML models: A set of experiments. *J. Syst. Softw.*, 79(5):727–742, 2006.

[113] M. Strembeck and J. Mendling. Modeling process-related RBAC models with extended UML activity models. *Inform. Softw. Tech.*, 53(5):456–483, 2011.

[114] M. Strembeck and U. Zdun. Modeling interdependent concern behavior using extended activity models. *J. Object Technology*, 7(6):143–166, 2008.

[115] M. Strembeck and U. Zdun. An approach for the systematic development of domain-specific languages. *Softw. Pract. Exper.*, 39(15):1253–1292, 2009.

[116] J. Suryadevara and R. K. Shyamasundar. UML-based approach to specify secured, fine-grained concurrent access to shared resources. *J. Object Technology*, 6(1):107–119, 2007.

[117] J. Suryadevara, L. Chung, and R. K. Shyamasundar. cmUML – a UML based framework for formal specification of concurrent, reactive systems. *J. Object Technology*, 7(8):187–207, 2008.

[118] C. Tibermacine, R. Fleurquin, and S. Sadou. A family of languages for architecture constraint specification. *J. Syst. Softw.*, 83(5):815–831, 2010.

[119] A. Torres, R. de Matos Galante, and M. S. Pimenta. A synergistic model-driven approach for persistence modeling with UML. *J. Syst. Softw.*, 84(6):942–957, 2011.

[120] J. Webster and R. T. Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS Quart.*, 26(2):pp. xiii–xxiii, 2002.

[121] I. Weisemöller and A. Schürr. A comparison of standard compliant ways to define domain specific languages. In *Workshops Proc. 10th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'07)*, volume 5002 of *LNCS*, pages 47–58. Springer, 2008.

[122] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requir. eng. paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.*, 11(1):102–107, Dec. 2005.

[123] T. Xie. Softw. eng. conf. Available at `http://web.engr.illinois.edu/~taoxie/seconferences.htm`. Last accessed: Feb 2, 2015, Feb 2013.

[124] S. Zachariadis, C. Mascolo, and W. Emmerich. The satin component system – A metamodel for engineering adaptable mobile systems. *IEEE T. Softw. Eng.*, 32(11):910–927, 2006.

[125] R. Zbib, A. Jain, D. Bassu, and H. Agrawal. Generating domain specific graphical modeling editors from meta models. In *30th Annu. Int. Comput. Softw. Applicat. Conf. (COMPSAC'06)*, pages 129–138. IEEE, 2006.

[126] U. Zdun and P. Avgeriou. Modeling architectural patterns using architectural primitives. In *Proc. 20th Annu. ACM SIGPLAN Conf. Object-oriented Programming, Syst., Languages, Applicat. (OOPLSA'05)*, pages 133–146. ACM, 2005.

[127] U. Zdun and M. Strembeck. Modeling composition in dynamic programming environments with model transformations. In *Proc. 5th Int. Symposium Software Composition (SC'06)*, volume 4089 of *LNCS*, pages 178–193. Springer, 2006.

[128] U. Zdun and M. Strembeck. Reusable architectural decisions for DSL design: Foundational decisions in DSL development. In *Proc. 14th European Conf. Pattern Languages of Programs (EuroPLoP'09)*, pages 1–37. ACM, 2009.

[129] H. Zhang, M. A. Babar, and P. Tell. Identifying relevant studies in software engineering. *Inform. Softw. Tech.*, 53(6):625–637, June 2011.

[130] Y. Zhang, Y. Liu, L. Zhang, Z. Ma, and H. Mei. Modeling and checking for non-functional attributes in extended UML class diagram. In *Proc. 32nd Annual IEEE Int. Conf. Computer Softw. and Applications*, pages 100–107, Washington, DC, USA, 2008. IEEE.

[131] G. Zoughbi, L. Briand, and L. Labiche. A UML profile for developing airworthiness-compliant (RTCA DO-178B), safety-critical software. In *Proc. 10th Int. Conf. Model Driven Eng. Languages Syst. (MoDELS'07)*, volume 4735 of *LNCS*, pages 574–588. Springer, 2007.

[132] G. Zoughbi, L. Briand, and Y. Labiche. Modeling safety and airworthiness (RTCA DO-178B) information: Conceptual model and UML profile. *Softw. Syst. Model.*, 10(3):337–367, 2011.

A Abbreviations

Table 34: Journal, conferences, and beyond.

Journal names	
AES	Advances in Engineering Software
BIS	International Conference on Business Information Systems
CJA	Chinese Journal of Aeronautics
ComSIS	Computer Science and Information Systems
CSI	Computer Standards & Interfaces
DETC	International Design Engineering Technical Conferences
DKE	Data & Knowledge Engineering
DSS	Decision Support Systems
ENTCS	Electronic Notes in Theoretical Computer Science
IETSoftw	IET Software
IJCIS	International Journal of Cooperative Information Systems
IJICIC	International Journal of Innovative Computing, Information and Control
IJOR	International Journal of Operational Research
IJSEKE	International Journal of Software Engineering and Knowledge Engineering

IJWIS	International Journal of Web Information Systems
ISeB	Information Systems and e-Business Management
ISF	Information Systems Frontiers
ISSE	Innovations in Systems and Software Engineering
IST	Information and Software Technology
JOT	Journal of Object Technology
JRPIT	Journal of Research and Practice in Information Technology
JSA	Journal of Systems Architecture
JSS	Journal of Systems and Software
JSUSE	Shenzhen Daxue Xuebao (Ligong Ban)/ Journal of Shenzhen University Science and Engineering
JSW	Journal of Software
JUCS	Journal of Universal Computer Science
PRC	Plastics, Rubber and Composites: Macromolecular Engineering
SCP	Science of Computer Programming
SOCA	Service Oriented Computing and Applications
SoSyM	Software & Systems Modeling
SP&E	Software: Practice and Experience
TC	IEEE Transactions on Computers
TEC	IEEE Transactions on Evolutionary Computation
TII	IEEE Transactions on Industrial Informatics
TOMACS	ACM Transactions on Modeling and Computer Simulation
TOSEM	ACM Transactions on Software Engineering and Methodology
TSE	IEEE Transactions on Software Engineering
TTBE	WIT Transactions on The Built Environment
VLC	Journal of Visual Languages & Computing

Conference names	
AAMS	International Conference on Autonomous Agents and Multiagent Systems
ACMSE	ACM Annual Southeast Regional Conference
AGTIVE	International Symposium on Applications of Graph Transformations with Industrial Relevance
AMOST	Workshop on Advances in Model-Based Testing
AOSE	International Workshop Agent-Oriented Software Engineering
APSCC	IEEE Asia-Pacific Services Computing Conference
ASE	ACM/IEEE International Conference on Automated Software Engineering
ASRU	IEEE Workshop on Automatic Speech Recognition and Understanding
ATIO	Aviation Technology, Integration and Operations Conference
BMFA	Workshop on Behavioral Modelling – Foundations and Applications
CAiSE	International Conference on Advanced Information Systems Engineering
CCCM	International Colloquium on Computing, Communication, Control, and Management
CEA	International Conference on Computer Engineering and Applications
CEC	Conference on Commerce and Enterprise Computing
CIT	IEEE International Conference on Computer and Information Technology
COMPSAC	Computer Software and Applications Conference
COP	International Workshop on Context-Oriented Programming
CRIWG	International Workshop on Groupware
CSEDU	International Conference on Computer Supported Education
CSMR	European Conference on Software Maintenance and Reengineering
DAC	Design Automation Conference
DATE	Design, Automation, and Test in Europe
DaWaK	International Conference on Data Warehousing and Knowledge Discovery
DEBS	International Conference on Distributed Event-based Systems
DSM	Domain-Specific Modeling
DSVIS	International Conference on Interactive Systems: Design, Specification, and Verification
ECBS	International Conference and Workshops on the Engineering of Computer-Based Systems
ECDL	European Conference on Digital Libraries
ECMDA-FA	European Conference on Model Driven Architecture – Foundations and Applications
ECMFA	European Conference on Modelling Foundations and Applications
ECSA	European Conference on Software Architecture

EDOC	International Enterprise Distributed Object Computing Conference
EDOCW	International Enterprise Distributed Object Computing Conference Workshops
EFTA	IEEE International Conference on Emerging Technologies and Factory Automation
EGOVIS	International Conference on Electronic Government and the Information Systems Perspective
EICS	Symposium on Engineering Interactive Computing Systems
EMSOFT	International Conference on Embedded Software
ER	International Conference on Conceptual Modeling
ESAW	International Workshop on Engineering Societies in the Agents World
ESEC	European Software Engineering Conference
ESWC	European Semantic Web Conference on The Semantic Web: Research and Applications
FASE	International Conference on Fundamental Approaches to Software Engineering
FDL	Forum on Specification Design Languages
FML	International Workshop on Formalization of Modeling Languages
GPC	International Conference on Grid and Pervasive Computing
GPCE	Generative Programming and Component Engineering Conference
HASE	IEEE International Symposium on High Assurance Systems Engineering
HICSS	Hawaii International Conference on System Sciences
IAT	International Conference on Intelligent Agent Technology
ICALT	International Conference on Advanced Learning Technologies
ICCSE	International Conference on Computer Science Education
ICEIS	International Conference on Enterprise Information Systems
ICINCO	International Conference on Informatics in Control, Automation and Robotics
ICIT	IEEE International Conference on Industrial Technology
ICOODB	International Conference on Objects and Databases
ICSE	International Conference on Software Engineering
ICSEA	International Conference on Software Engineering Advances
ICSOC	International Conference on Service-oriented Computing
ICSTW	International IEEE Conference on Software Testing, Verification and Validation
ICWE	International Conference on Web Engineering
ICWS	IEEE International Conference on Web Services
INDIN	IEEE International Conference on Industrial Informatics
ISC	IEEE Southeastcon Conference
ISEC	India Software Engineering Conference
ISORC	IEEE International Symposium on Object and Component-oriented Real-Time Distributed Computing
ISSRE	IEEE International Symposium on Software Reliability Engineering
ITNG	International Conference on Information Technology
ITSIM	International Symposium in Information Technology
MoDELS	International Conference on Model Driven Engineering Languages and Systems
MUSIC	International Conference on Mobile, Ubiquitous and Intelligent Computing
OOPSLA	Object-Oriented Programming, Systems, Languages & Applications Conference
OTM	On the Move Confederated International Conferences (CoopIS, DOA, GADA, IS, and ODBASE)
PERCOM	IEEE International Conference on Pervasive Computing and Communications
QSIC	International Conference on Quality Software
REFSQ	International Working Conference on Requirements Engineering: Foundation for Software Quality
RTSS	Real-Time Systems Symposium
SAC	ACM Symposium on Applied Computing
SAFECOMP	International Conference on Computer Safety, Reliability, and Security
SAM	System Analysis and Modelling Conference
SC	International Symposium on Software Composition
SCC	IEEE International Conference on Services Computing
SCCC	International Conference of the Chilean Computer Science Society
SEAA	Euromicro Conference on Software Engineering and Advanced Applications
SEFM	International Conference on Software Engineering and Formal Methods
SEKE	International Conference on Software Engineering and Knowledge Engineering
SenSys	ACM Conference on Embedded Networked Sensor Systems
SERP	International Conference on Software Engineering Research and Practice

SFM	International Conference on Formal Methods for the Design of Computer, Communication, and Software Systems
SIGMOD	ACM International Conference on Management of Data
SLE	International Conference on Software Language Engineering
SOCC	IEEE International System-on-Chip Conference
SoMeT	Conference on New Trends in Software Methodologies, Tools and Techniques
SOSE	International Symposium on Service-oriented System Engineering
SPLASH	ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity
SPLC	International Software Product Line Conference
SYNASC	International Symposium on Symbolic and Numeric Algorithms for Scientific Computing
SysCon	IEEE Systems Conference
VL/HCC	Visual Languages and Human-Centric Computing Symposium
WaGe	Third International Workshop on Workflow Management and Applications in Grid Environments
WISES	International Workshop on Intelligent Solutions in Embedded Systems
WSC	Winter Simulation Conference
	Various
AITO	Association Internationale pour les Technologies Objets
ICPS	International Conference Proceeding Series

B Engine-specific Search Strings

Listing 4: ACMDL, submitted via direct, batched HTTP requests.

```
("SysML profile" OR "UML profile" OR "UML2 profile" OR UML2 OR UML2.0
OR "UML 2" OR "UML 2.0" OR MOF2 OR "MOF 2" OR "MOF 2.0" OR MOF2.0 OR
UML OR "unified modeling language" OR MOF OR "meta object facility" OR
SysML OR "systems modeling language" OR OCL OR "object constraint
language")
AND
(customisation OR customization OR metamodel OR "meta-model" OR "meta
model" OR stereotype OR profile OR "domain-specific language" OR
"domain specific language" OR "code generation" OR "meta-level" OR
metalevel OR "meta level" OR extension OR "abstract syntax" OR "visual
language" OR "concrete syntax" OR "domain-specific modeling" OR
"domain specific modeling" OR DSM OR specialization OR specialisation
OR refinement OR "model-to-model transformation" OR "model to model
transformation" OR "M2M transformation" OR "model-to-text
transformation" OR "model to text transformation" OR "M2T
transformation" OR "model transformation" OR "conceptual model" OR
"domain model" OR "model generation")
AND
(PublishedAs:journal OR PublishedAs:proceeding OR
PublishedAs:transaction)
AND
((Keywords:UML OR Keywords:"unified modeling language" OR Keywords:MOF
OR Keywords:"meta object facility" OR Keywords:SysML OR
Keywords:"systems modeling language")
OR
(Abstract:UML OR Abstract:"unified modeling language" OR Abstract:MOF
OR Abstract:"meta object facility" OR Abstract:SysML OR
Abstract:"systems modeling language"))
```

Listing 5: IEEEXplore, submitted and refined via the web UI (form).

```
(("SysML profile" OR "UML profile" OR "UML2 profile" OR UML2 OR UML2.0
OR "UML 2" OR "UML 2.0" OR MOF2 OR "MOF 2" OR "MOF 2.0" OR MOF2.0 OR
UML OR "unified modeling language" OR MOF OR "meta object facility" OR
SysML OR "systems modeling language" OR OCL OR "object constraint
language")
AND
(customisation OR customization OR metamodel OR "meta-model" OR "meta
model" OR stereotype OR profile OR "domain-specific language" OR
"domain specific language" OR "code generation" OR "meta-level" OR
metalevel OR "meta level" OR extension OR "abstract syntax" OR "visual
language" OR "concrete syntax" OR "domain-specific modeling" OR
"domain specific modeling" OR DSM OR specialization OR specialisation
OR refinement OR "model-to-model transformation" OR "model to model
transformation" OR "M2M transformation" OR "model-to-text
transformation" OR "model to text transformation" OR "M2T
transformation" OR "model transformation" OR "conceptual model" OR
"domain model" OR "model generation"))
```

The actual URL used for dispatching and for reproducing the IEEEXplore search is shown below, containing the URL-encoded search string from Listing 5:

```
http://ieeexplore.ieee.org/search/searchresult.jsp?bulkSetSize=1845&refinements%
3D4291944822%2C4291944246%26ranges%3D2005_2013_p_Publication_Year%26matchBoolean%3Dtrue%
26searchField%3DSearch_All%26queryText%3D%28%28%22SysML+profile%22+OR+%22UML+profile%
22+OR+%22UML2+profile%22+OR+UML2+OR+UML2.0+OR+%22UML+2%22+OR+%22UML+2.0%22+OR+MOF2+OR+
%22MOF+2%22+OR+%22MOF+2.0%22+OR+MOF2.0+OR+UML+OR+%22unified+modeling+language%22+OR+
MOF+OR+%22meta+object+facility%22+OR+SysML+OR+%22systems+modeling+language%22+OR+OCL+OR+
%22object+constraint+language%22%29+AND+%28customisation+OR+customization+OR+metamodel+
OR+%22meta-model%22+OR+%22meta+model%22+OR+stereotype+OR+profile+OR+%22domain-specific+
language%22+OR+%22domain+specific+language%22+OR+%22code+generation%22+OR+%22meta-level%
22+OR+metalevel+OR+%22meta+level%22+OR+extension+OR+%22abstract+syntax%22+OR+%22visual+
language%22+OR+%22concrete+syntax%22+OR+%22domain-specific+modeling%22+OR+%22domain+
specific+modeling%22+OR+DSM+OR+specialization+OR+specialisation+OR+refinement+OR+
%22model-to-model+transformation%22+OR+%22model+to+model+transformation%22+OR+%22M2M+
transformation%22+OR+%22model-to-text+transformation%22+OR+%22model+to+text+transformation%
22+OR+%22M2T+transformation%22+OR+%22model+transformation%22+OR+%22conceptual+model%22+OR+
%22domain+model%22+OR+%22model+generation%22%29%29
```

Listing 6: Scopus, submitted via web UI (form).

```
("SysML profile" OR "UML profile" OR "UML2 profile" OR uml2 OR uml2.0
OR "UML 2" OR "UML 2.0" OR mof2 OR "MOF 2" OR "MOF 2.0" OR mof2.0 OR
uml OR "unified modeling language" OR mof OR "meta object facility" OR
sysml OR "systems modeling language" OR ocl OR "object constraint
language")
AND
(customisation OR customization OR metamodel OR "meta-model" OR "meta
model" OR stereotype OR profile OR "domain-specific language" OR
"domain specific language" OR "code generation" OR "meta-level" OR
metalevel OR "meta level" OR extension OR "abstract syntax" OR "visual
language" OR "concrete syntax" OR "domain-specific modeling" OR
"domain specific modeling" OR dsm OR specialization OR specialisation
OR refinement OR "model-to-model transformation" OR "model to model
transformation" OR "M2M transformation" OR "model-to-text
transformation" OR "model to text transformation" OR "M2T
transformation" OR "model transformation" OR "conceptual model" OR
"domain model" OR "model generation")
AND
(LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR
LIMIT-TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR
LIMIT-TO(PUBYEAR, 2009) OR LIMIT-TO(PUBYEAR, 2008) OR
LIMIT-TO(PUBYEAR, 2007) OR LIMIT-TO(PUBYEAR, 2006) OR
LIMIT-TO(PUBYEAR, 2005)) AND (LIMIT-TO(DOCTYPE, "cp") OR
LIMIT-TO(DOCTYPE, "ar")) AND (LIMIT-TO(LANGUAGE, "English"))
```

Listing 7: SpringerLink, submitted via web UI (form).

```
("SysML profile" OR "UML profile" OR "UML2 profile" OR UML2 OR
UML2.0 OR "UML 2" OR "UML 2.0" OR MOF2 OR "MOF 2" OR "MOF 2.0" OR
MOF2.0 OR UML OR "unified modeling language" OR MOF OR
"meta object facility" OR SysML OR "systems modeling language" OR
OCL OR "object constraint language")
AND
(customisation OR customization OR metamodel OR "meta-model" OR
"meta model" OR stereotype OR profile OR "domain-specific language" OR
"domain specific language" OR "code generation" OR "meta-level" OR
metalevel OR "meta level" OR extension OR "abstract syntax" OR
"visual language" OR "concrete syntax" OR "domain-specific modeling" OR
"domain specific modeling" OR DSM OR specialization OR
specialisation OR refinement OR "model-to-model transformation" OR
 "model to model transformation" OR "M2M transformation" OR
"model-to-text transformation" OR "model to text transformation" OR
 "M2T transformation" OR "model transformation" OR "conceptual model" OR
 "domain model" OR "model generation")
```

C Venue Lists

Table 35: MS/AS: Microsoft Academic Search ([72]; subset of 53 journals in category "Computer Science"/"Software Engineering"; retrieved on Feb 4, 2013).

Journal	Publication count	MS/AS rating
TSE - IEEE Transactions on Software Engineering	3928	183
SOFTWARE - IEEE Software	4071	108
ACM Sigsoft Software Engineering Notes	5021	85
SPE - Software - Practice and Experience	3247	80
TOSEM - ACM Transactions on Software Engineering and Methodology	332	69

Journal	Publication count	MS/AS rating
JSS - Journal of Systems and Software	3168	61
IET Software/IEE Proceedings - Software	594	57
FAC - Formal Aspects of Computing	1037	56
BIT - Bit Numerical Mathematics	2790	51
INFSOF - Information & Software Technology	2904	46
ISJ - Information Systems Journal	395	45
ENVSOFT - Environmental Modelling and Software	1866	44
AES - Advances in Engineering Software	2901	42
FMSD - Formal Methods in System Design	615	42
STTT - International Journal on Software Tools for Technology Transfer	535	41
ESE - Empirical Software Engineering	491	36
RE - Requirements Engineering	363	36
STVR - Software Testing, Verification & Reliability	497	33
ASE - Automated Software Engineering	445	33
SMR - Journal of Software Maintenance and Evolution: Research and Practice	527	32
SOSYM - Software and System Modeling	357	31
CONSTRAINTS - Constraints - An International Journal	433	29
CSI - Computer Standards & Interfaces	2272	28
EWC - Engineering With Computers	742	28
SOPR - Software Process: Improvement and Practice	505	28
ANSOFT - Annals of Software Engineering	375	28
Quality and Reliability Engineering International	1988	26
JOT - Journal of Object Technology	752	26
IJSEKE - International Journal of Software Engineering and Knowledge Engineering	728	26
Concurrent Engineering: R&A - Concurrent Engineering: Research and Applications	654	26
TAPOS - Theory and Practice of Object Systems	127	26
SQJ - Software Quality Journal	478	22
Artificial - Advanced Engineering Informatics	458	22
CEE - Computers & Electrical Engineering	1507	21
STP - Software - Concepts and Tools / Structured Programming	182	21
CSSE - Computer Systems: Science & Engineering	336	19
ISEM - Information Systems and E-business Management	230	15
IJCIA - International Journal of Computational Intelligence and Applications	263	14
JETC - ACM Journal on Emerging Technologies in Computing Systems	137	14
JSI - Journal of Systems Integration	233	11
STT - Softwaretechnik-trends	189	11
ACM Sigcpr Computer Personnel	274	10
KES Journal - International Journal of Knowledge-based and Intelligent Engineering Systems	223	10
ISSE - Innovations in Systems and Software Engineering	178	9
ACM Sigapp Applied Computing Review	159	9
Robotica	68	8
SOFO - Software Focus	66	7
ISI - Ingénierie Des Systèmes D'information	342	6
ECEASST - Electronic Communication of The European Association of Software Science and Technology	166	6
LOGIN - Log in	671	5
ICOM - Zeitschrift Für Interaktive Und Kooperative Medien	399	4
ACM Sigsoc Bulletin	197	4
CSI - Computer Standards & Interfaces	1	0

Table 36: MS/AS: Microsoft Academic Search ([72]; subset of 284 conferences in category "Computer Science"/"Software Engineering"; retrieved on Feb 4, 2013).

Conference	Publication count	MS/AS rating
ICSE - International Conference on Software Engineering	4715	118
ITC - International Test Conference	4476	89
CAV - Computer Aided Verification	1071	88
OOPSLA - Conference on Object-Oriented Programming Systems, Languages, and Applications	1822	59
SPIN - International Workshop on Model Checking of Software	644	59
ICSM - International Conference on Software Maintenance	1489	57
TACAS - Tools and Algorithms for Construction and Analysis of Systems	669	56
CP - Principles and Practice of Constraint Programming	1245	54
IFIP - World Computer Congress	2831	49
Requirements Engineering	406	47
Dagstuhl Seminars	3061	46
SAS(WSA) - Static Analysis Symposium/Workshop on Static Analysis	556	46
AOSD - Aspect-Oriented Software Development	347	46
WCRE - Working Conference on Reverse Engineering	650	45
ASE - Automated Software Engineering	1208	44
ESEC - European Software Engineering Conference	691	44
SAC - ACM Symposium on Applied Computing	4617	43
UML - The Unified Modeling Language	476	42
FM - World Congress on Formal Methods	785	39
ISSRE - International Symposium on Software Reliability Engineering	870	37
ICPC - International Conference on Program Comprehension	634	37
RTA - Rewriting Techniques and Applications	698	36
TAPSOFT - Theory and Practice of Software Development	393	35
ISSTA - International Symposium on Software Testing and Analysis	359	35
COMPSAC - International Computer Software and Applications Conference	3409	34
Fall Joint Computer Conference	310	34
METRICS - IEEE International Software Metrics Symposium	337	33
INRIA - INRIA	290	32
FASE - Fundamental Approaches to Software Engineering	397	31
TOOLS - Technology of Object-Oriented Languages and Systems	1424	30
WETICE - Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises	1206	30
FoSSaCS - Foundations of Software Science and Computation Structure	399	30
FMCAD - Formal Methods in Computer-Aided Design	348	30
VMCAI - Verification, Model Checking and Abstract Interpretation	278	30
ISMM - International Symposium on Memory Management	231	30
ICSR - International Conference on Software Reuse	335	29
CSMR - Conference on Software Maintenance and Reengineering	724	28
SPLC - Software Product Lines	396	28
SEKE - Software Engineering and Knowledge Engineering	1832	27
FGCS - Fifth Generation Computer Systems	346	27
WOSP - Workshop on Software and Performance	231	27
Generative Programming and Component Engineering	226	27
AMAST - Algebraic Methodology and Software Technology	472	26
COORDINATION - Coordination Models and Languages	295	26
APSEC - Asia-Pacific Software Engineering Conference	1308	25
WICSA - Working IEEE/IFIP Conference on Software Architecture	414	25
SOFTVIS - Software Visualization	186	25
EUROMICRO - Conference on Software Engineering and Advanced Applications	1153	23
LOPSTR - Logic Program Synthesis and Transformation	402	23
MPC - Mathematics of Program Construction	220	23
IWSSD - International Workshop on Software Specifications & Design	148	23
VEE - International Conference on Virtual Execution Environments	111	23
ICSP - International Conference on the Software Process	87	23

Conference	Publication count	MS/AS rating
CMG - Computer Measurement Group Conference	3155	22
MODELS - Model Driven Engineering Languages and Systems	536	22
SCM - System Configuration Management	216	22
FIW - Feature Interactions in Telecommunications and Software Systems	208	22
PASTE - Workshop on Program Analysis For Software Tools and Engineering	110	22
ICECCS - International Conference on Engineering of Complex Computer Systems	710	21
IEEE International Conference on Formal Engineering Methods	428	21
MSR - Mining Software Repositories	267	21
CBSE - Component-Based Software Engineering	229	21
ISESE - International Symposium on Empirical Software	207	21
CSC - ACM Annual Computer Science Conference	1756	20
QEST - Quantitative Evaluation of Systems	388	20
RE - IEEE Int. Conf. on Requirements Engineering	104	20
Australian Software Engineering Conference	540	19
HASE - High-Assurance Systems	415	19
IFM - Integrated Formal Methods	207	19
FORMATS - Formal Modeling and Analysis of Timed Systems	184	19
IWPSE - International Workshop on Principles of Software Evolution	172	19
Formal Methods for Components and Objects	133	19
FroCos - Frontiers of Combining Systems	132	19
XP Universe - Extreme Programming	646	18
SAFECOMP - International Conference on Computer Safety, Reliability and Security	471	18
SEFM - Conference on Software Engineering and Formal Methods	337	18
SCAM - Source Code Analysis and Manipulation	229	18
Meta-Level Architectures and Reflection	57	18
ISCIS - International Symposium on Computer and Information Sciences	870	17
Ershov Memorial Conference	405	17
WIKIS - International Symposium on Wikis	177	17
SSR - ACM SIGSOFT Symposium on Software Reusability	68	17
ICSP - International Software Process Workshop	628	16
QSIC - International Conference on Quality Software	556	16
PROFES - Product Focused Software Process Improvement	399	16
OOIS - Object Oriented Information Systems	394	16
EWSPT - European Workshop on Software Process Technology	244	16
REFSQ - Requirements Engineering: Foundation for Software Quality	200	16
SOCO - Software Composition	114	16
FATES/RV - International Workshop on Formal Approaches to Testing of Software	74	16
WOSS - Workshop on Self-Healing Systems	49	16
SERP - Software Engineering Research and Practice	1041	15
SEW - Annual Software Engineering Workshop	279	15
ICCBSS - International Conference on COTS-Based Software Systems	195	15
ECMDAFA - European Conference on Model Driven Architecture - Foundations and Applications	155	15
SELMAS - Software Engineering for Large-Scale Multi-Agent Systems	99	15
CD - IFIP/ACM Working Conference on Component Deployment	86	15
Software Product Family Engineering	76	15
AMAST - Algebraic Methodology and Software Technology	50	15
Automatic Verification Methods for Finite State Systems	35	15
COMPOS - Compositionality: The Significant Difference	26	15
ACS/IEEE International Conference on Computer Systems and Applications	1232	14
ATVA - Automated Technology for Verification and Analysis	256	14
DFG Projects	182	14
STEP - Software Technology and Engineering Practice	180	14
CTRS - Conditional Term Rewriting Systems	126	14
ISAS - International Service Availability Symposium	105	14

Conference	Publication count	MS/AS rating
School on Formal Methods for the Design of Computer, Communication and Software Systems	80	14
GCSE - Generative and Component-Based Software Engineering	55	14
ICFPC - Formalization of Programming Concepts	49	14
ICGSE - IEEE International Conference on Global Software Engineering	242	13
ISOTAS - International Symposium on Object Technologies for Advanced Software	53	13
IRI - Information Reuse and Integration	973	12
ICST - International Conference on Software Testing, Verification, and Validation	604	12
ISOLA - Leveraging Applications of Formal Methods	313	12
OSS - Open Source Software	237	12
FMICS - Formal Methods for Industrial Critical Systems	119	12
SEE - Software Engineering Environments	105	12
PDSE - International Symposium on Software Engineering for Parallel and Distributed Systems	103	12
VISSOFT - Visualizing Software for Understanding and Analysis	89	12
EWSA - European Workshop on Software Architecture	65	12
GTTSE - Generative and Transformational Techniques in Software Engineering	40	12
Ontologies in Agent Systems	31	12
SNPD - Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing	1702	11
SE - Software Engineering	558	11
ESEM - Empirical Software Engineering and Measurement	470	11
FDL - Forum on specification & Design Languages	288	11
Modellierung	218	11
IESA - Interoperability for Enterprise Software and Applications	213	11
JCKBSE - Joint Conference on Knowledge-Based Software Engineering	160	11
SAM - System Analysis and Modeling	73	11
Software Engineering and Middleware Engineering Distributed Objects	67	11
ETX - Eclipse Technology eXchange	66	11
Learning Software Organizations	61	11
AGILEDC - Agile Development Conference	49	11
WSE - Website Evolution	47	11
FMSP - Formal Methods in Software Practice	33	11
SAIG - Semantics, Applications, and Implementation of Program Generation	29	11
MDAFA - Model Driven Architecture Foundations and Applications	29	11
SEAA - Software Engineering and Advanced Applications	364	10
SCCC - International Conference of the Chilean Computer Science Society	333	10
IASTEDSEA - Software Engineering and Applications	148	10
EPEW - European Performance Engineering Workshop	121	10
TAICPART - Testing: Academic & Industrial Conference - Practice And Research Techniques	109	10
APAQS - Asia-Pacific Conference on Quality Software	88	10
QOSA - Quality of Software Architectures	77	10
SOQUA - Software Quality	75	10
TAP - Tests and Proofs	60	10
WIFT - Workshop on Industrial-Strength Formal Specification Techniques	40	10
CASSIS - Construction and Analysis of Safe, Secure, and Interoperable Smart Devices	29	10
FCA - Formal Concept Analysis	19	10
ACISICIS - ACIS International Conference on Computer and Information Science	966	9
IASTEDSE - Software Engineering	269	9
JCIT - Jerusalem Conference on Information Technology	219	9
EUROSPI - European Conference on Software Process Improvement	110	9
HVC - Haifa Verification Conference	109	9
VSTTE - Verified Software: Theories, Tools, Experiments	58	9
Program Construction	34	9
Reflection and Software Engineering	18	9

Conference	Publication count	MS/AS rating
Software Engineering Research and Applications	444	8
WER - Workshop em Engenharia de Requisitos	250	8
TASE - Theoretical Aspects of Software Engineering	218	8
SOCA - Service-Oriented Computing and Applications	218	8
ECSA - European Conference on Software Architecture	188	8
Formale Beschreibungstechniken	116	8
IWFM - Irish Workshop in Formal Methods	69	8
FSEN - Fundamentals of Software Engineering	64	8
ICMT - International Conference on Model Transformation	56	8
ACL2 - International Workshop on the ACL2 Theorem Prover and Its Applications	54	8
KORSO	42	8
ESPRIT ARES Workshops	34	8
IW-SAPF - International Workshop on Software Architectures for Product Families	28	8
UNI/IIST - Anniversary Colloquium of UNU/IIST	27	8
SDE - Software Development Environments	25	8
SEM - Software Engineering and Middleware	23	8
Semantics of Concurrent Computation	19	8
Temporal Logic in Specification	16	8
SCESM - Scenarios and state machines: models, algorithms, and tools	16	8
Object Modeling with the OCL	13	8
CSSE - International Conference on Computer Science and Software Engineering	1593	7
ICSOFT - International Conference on Software and Data Technologies	723	7
ICETET - International Conference on Emerging Trends in Engineering & Technology	713	7
CAINE - Computer Applications in Industry and Engineering	602	7
ICSEA - International Conference on Software Engineering Advances	481	7
SWSTE - IEEE International Conference on Software - Science, Technology and Engineering	79	7
International Computing Symposium	71	7
BDIM - International Workshop on Business-Driven IT Management	48	7
European Conference on Software Quality	45	7
Radical Innovations of Software and Systems Engineering in the Future	26	7
RODIN - RODIN Project	25	7
A-MOST - Advances in Model-Based Software Testing	14	7
NEMS - Nano/Micro Engineered and Molecular Systems	1944	6
DEPCOS - International Conference on Dependability of Computer Systems	196	6
Fault-Tolerant Computing Systems / Fehlertolerierende Rechensysteme	192	6
IASSE - International Conference on Intelligent and Adaptive Systems and Software Engineering	166	6
LMO - Langages et Modèles à Objets	137	6
IWSM - International Workshop on Software Measurement	99	6
C3S2E - Canadian Conference on Computer Science & Software Engineering	87	6
RISE - Rapid Integration of Software Engineering Techniques	46	6
SEAFOOD - Software Engineering Approaches for Offshore and Outsourced Development	45	6
Global Constraint Optimization and Constraint Satisfaction	32	6
SPLST - Symposium on Programming Languages and Software Tools	25	6
Performance Engineering	24	6
ROOM - Rigorous Object-Oriented Methods	20	6
Component-Based Software Quality	18	6
ESERNET - Experimental Software Engineering Network	16	6
Algebraic and Coalgebraic Methods in the Mathematics of Program Construction	9	6
PAP - International Conference and Exhibition on Practical Applications of Prolog	8	6
SESPSDE - Software Engineering Symposium on Practical Software Development Environments	8	6
Verification and Validation of Enterprise Information Systems	136	5

Conference	Publication count	MS/AS rating
FIFF Jahrestagung	127	5
IFIP TC2 Publications	60	5
BCS-FACS Publications	44	5
GSEM - Grid Services Engineering and Management	43	5
VaMoS - Variability Modelling of Software-Intensive Systems	35	5
Semantics of Specification Languages	19	5
Larch - International Workshop on Larch	17	5
RTSE - Requirements Targeting Software and Systems Engineering	15	5
Formal Methods and Testing	12	5
Fachtagung Prozessrechner	304	4
SEDE - Software Engineering and Data Engineering	271	4
ITEE - Information Technologies in Environmental Engineering	163	4
Conference Internationale Associant Chercheurs Vietnamiens et Francophones en Informatique	82	4
Software Quality and Productivity	61	4
ISEC - India Software Engineering Conference	46	4
PROSPECTRA	41	4
TEX for Scientific Documentation	38	4
ISIM - International Conference on Information System Implementation and Modeling	35	4
TFM - Teaching Formal Methods	31	4
IPSEN	29	4
RIMS Symposia on Software Science and Engineering	29	4
The Analysis of Concurrent Systems	27	4
ISCNZ - Information Systems Conference of New Zealand	25	4
Portability of Numerical Software	24	4
ASWSD - Automotive Software Workshop	23	4
WWV - Automated Specification and Verification of Web Sites	23	4
Problems and Methodologies in Mathematical Software Production	22	4
COODBSE - Colloquium on Object Orientation in Databases and Software Engineering	19	4
VISSAS - Verification of Infinite-State Systems with Applications to Security	17	4
WOODPECKER - Workshop on Open Distribute Processing: Enterprise, Computation, Knowledge, Engineering and Realisation	10	4
FMSB - Formal Methods in Systems Biology	9	4
WACC - Work Activities Coordination and Collaboration	8	4
Umwelt - Informatik für den Umweltschutz	278	3
JIISIC - Ibero-American Symposium on Software Engineering and Knowledge Engineering	187	3
SoMeT - Software Methodologies, Tools and Techniques	130	3
EUNIS - European University Information Systems	82	3
ISESS - International Symposium on Environmental Software Systems	35	3
ACIT - Automation, Control, and Information Technology	29	3
iStar - International i* Workshop	21	3
PSSE - Pernambuco Summer School on Software Engineering	18	3
Personal Computing	15	3
ICEISSAM - Software Audit and Metrics	12	3
CIP-Project	11	3
COEA - Component-Oriented Enterprise Applications	10	3
WISER - Workshop on Interdisciplinary Software Engineering Research	6	3
Publications of the German Chapter of the ACM	141	2
EBUSINESS - E-Business	141	2
SEUH - Software Engineering im Unterricht der Hochschulen	119	2
SETP - Software Engineering Theory and Practice	64	2
International Working Conference on Model Realism	54	2
CAL - Conférence francophone sur les Architectures Logicielles	44	2
HINC - History of Nordic Computing	41	2
Managing Information Technology's Organisational Impact	21	2
Software-Entwicklung	21	2
Software Management	20	2

Conference	Publication count	MS/AS rating
The IOTA Programming System	18	2
Objective Software Quality	18	2
German-Argentinian Workshop on Information Technology	13	2
MASSA - Multiagent Systems and Software Architecture	10	2
CIbSE - Conferencia Iberoamericana de Software Engineering	73	1
WIMAW - Management der Anwendungsentwicklung und -wartung	68	1
GMMEMV - Elektromagnetische Verträglichkeit in der KFZ-Technik	47	1
Frauenarbeit und Informatik	36	1
UH - Unternehmen Hochschule	26	1
Software-Architektur	13	1
ADIS - Apoyo a la Decisión en Ingeniería del Software / Decision Support in Software Engineering	12	1
DDOPS - Development and Deployment of Product Software	6	1
IR Workshop	1	1
SETA - Symposium on Environments and Tools for Ada	1	1
SCSS - International Conference on Complex, Intelligent and Software Intensive Systems	2	1
Testen, Analysieren und Verifizieren von Software	17	0
Portable Software	12	0
ENASE - International Conference on Evaluation of Novel Approaches to Software Engineering	2	0
ACM SIGSOFT FSE - ACM SIGSOFT International Symposium on the Foundations of Software Engineering	1	0

Table 37: SJR: SCImago Journal Rank ([99]; subset of 184 journals for subject area "Computer Science", subject category "Software", 2012).

Journal
Foundations and Trends in Machine Learning
Proceedings of the Annual ACM Symposium on Theory of Computing
ACM Transactions on Database Systems
Briefings in Bioinformatics
Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems
IEEE Transactions on Software Engineering
Theory and Practice of Logic Programming
Mathematical Programming
Proceedings of the IEEE International Conference on Computer Vision
Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms
Empirical Software Engineering
ACM Transactions on Software Engineering and Methodology
ACM Transactions on Programming Languages and Systems
Journal of Statistical Software
Journal of Mathematical Imaging and Vision
Journal of Functional Programming
ACM Transactions on Graphics
Communications of the ACM
Proceedings - International Conference on Data Engineering
IEEE Transactions on Multimedia
Random Structures and Algorithms
Transactions on Data Privacy
IEEE Transactions on Computational Intelligence and AI in Games
IEEE Software
Artificial Intelligence and Law
IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers
IEEE Micro
SIGMOD Record
Software and Systems Modeling
ACM Transactions on Mathematical Software

Journal
Software Testing Verification and Reliability
IEEE Internet Computing
ACM Transactions on Autonomous and Adaptive Systems
Journal of Scientific Computing
Algorithmica
Journal of Computer Security
IEEE Transactions on Reliability
Conference Record of the Annual ACM Symposium on Principles of Programming Languages
Journal of Systems and Software
Journal of Software Maintenance and Evolution
Science of Computer Programming
IEEE Transactions on Visualization and Computer Graphics
Automated Software Engineering
International Journal of Web and Grid Services
Discrete Optimization
Journal of Visual Communication and Image Representation
Journal of Logic and Algebraic Programming
Proceedings - International Conference on Software Engineering
IEEE Transactions on Autonomous Mental Development
Formal Aspects of Computing
International Journal of Innovative Computing, Information and Control
Computer Graphics Forum
Journal of Artificial Intelligence
CMES - Computer Modeling in Engineering and Sciences
Computer Standards and Interfaces
Computer Supported Cooperative Work
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition
IEEE Transactions on Affective Computing
Computer Journal
Optimization Methods and Software
ACM Transactions on Design Automation of Electronic Systems
ACM SIGPLAN Notices
Natural Language Engineering
Advances in Engineering Software
Transactions on Embedded Computing Systems
Performance Evaluation
Concurrency Computation Practice and Experience
Journal of Web Engineering
Proceedings - Graphics Interface
Journal of Simulation
International Journal for Numerical Methods in Biomedical Engineering
Journal of Ambient Intelligence and Smart Environments
Transactions on Architecture and Code Optimization
Proceedings of the Annual Symposium on Computational Geometry
Service Oriented Computing and Applications
Computer Aided Geometric Design
Software Quality Journal
Zidonghua Xuebao/Acta Automatica Sinica
BIT Numerical Mathematics
Multimedia Tools and Applications
Computer Methods and Programs in Biomedicine
IEEE Multimedia
Software - Practice and Experience
Fundamenta Informaticae
Computer Languages, Systems and Structures
Journal of Computer Science
ACM Transactions on Modeling and Computer Simulation
Entertainment Computing
International Journal of Data Warehousing and Mining
Computers and Graphics

Journal
IET Software
Information Management and Computer Security
Presence: Teleoperators and Virtual Environments
Jisuanji Xuebao/Chinese Journal of Computers
Innovations in Systems and Software Engineering
Journal of Computer Science and Technology
Proceedings of the National Conference on Artificial Intelligence
Ruan Jian Xue Bao/Journal of Software
IEEE Computer Graphics and Applications
ACM Journal on Emerging Technologies in Computing Systems
Visual Computer
Simulation
Proceedings - Symposium on Computer Arithmetic
IET Image Processing
IET Computers and Digital Techniques
International Journal of Agent-Oriented Software Engineering
International Journal of Ad Hoc and Ubiquitous Computing
IET Information Security
Microprocessors and Microsystems
Proceedings - IEEE International Conference on Robotics and Automation
International Journal of Web Based Communities
Neural Network World
Computer Animation and Virtual Worlds
Journal of Object Technology
International Journal of Digital Content Technology and its Applications
Journal of Research and Practice in Information Technology
International Journal of Electronic Government Research
Jisuanji Yanjiu yu Fazhan/Computer Research and Development
IEEE International Conference on Fuzzy Systems
ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb
Informatica
Xitong Fangzhen Xuebao/Acta Simulata Systematica Sinica
International Journal of Digital Earth
IET Computer Vision
Journal of Software
International Journal of Software Engineering and Knowledge Engineering
IEICE Transactions on Communications
Peer-to-Peer Networking and Applications
International Journal of Computational Science and Engineering
IEICE Transactions on Information and Systems
Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics
International Journal of Modelling and Simulation
International Journal of Communication Networks and Information Security
Computers in Cardiology
Pollack Periodica
Journal of Decision Systems
Webology
Proceedings of the International Workshop on Rapid System Prototyping
International Geoscience and Remote Sensing Symposium (IGARSS)
International Journal of Computer Games Technology
International Journal of Digital Crime and Forensics
Proceedings of the International Joint Conference on Neural Networks
International Journal of Network Management
International Journal of Embedded Systems
Programming and Computer Software
International Journal of Computers for Mathematical Learning
Design Automation for Embedded Systems
Journal of Communications Software and Systems
Journal of WSCG

Journal
International Journal of Ambient Computing and Intelligence
International Journal of Information and Computer Security
Journal of Physical Agents
Proceedings of the IEEE International Conference on Industrial Technology
International Journal of Automation and Control
Proceedings of the IEEE Symposium on Computer-Based Medical Systems
International Journal of Cognitive Informatics and Natural Intelligence
Computer Software
WSEAS Transactions on Signal Processing
Informatologia
Synthesis Lectures on Network Simulation
Information Security Journal
International Journal of Mobile Network Design and Innovation
Cognitive Technologies
Understanding Complex Systems
Reliable Computing
Cutter IT Journal
International Journal of Open Source Software and Processes
Journal of Digital Forensic Practice
Shu Ju Cai Ji Yu Chu Li/Journal of Data Acquisition and Processing
IEEE International Professional Communication Conference
Computer Assisted Mechanics and Engineering Sciences
ABB Review
IEEE Conference Record of Annual Pulp and Paper Industry Technical Conference
Synthesis Lectures on Mobile and Pervasive Computing
Modelling, Measurement and Control C
International Journal of High Performance Computing and Networking
Systems Science
Modelling, Measurement and Control B
Advances in Modeling and Analysis C
Ada User Journal
HP Laboratories Technical Report
International Journal of Simulation: Systems, Science and Technology
International Journal of Imaging and Robotics