A Computational Infrastructure for Research Synthesis in Software Engineering

Paulo Sérgio M. dos Santos, Ian E. do Nascimento, and Guilherme H. Travassos

Federal University of Rio de Janeiro,
ESE Group, PESC/COPPE,
Rio de Janeiro – RJ, Brazil
{pasemes, ght}@cos.ufrj.br, ianesteves@poli.ufrj.br

Abstract. Research synthesis is an essential instrument to consolidate scientific knowledge regarding the benefits and drawbacks of software technologies. However, conducting a research synthesis can be challenging given the amount of data and information to manage and aggregate. Furthermore, comparing and contrasting evidence besides keeping track of all aggregation decisions can be cumbersome whether manually performed. Therefore, in this paper, we present a computational infrastructure to support research synthesis. The tool offers a graphical and formal notation to represent Software Engineering qualitative and quantitative evidence. The infrastructure's design and navigational concerns besides the details about its internal algorithms are presented aiming at supporting the explanations on how the formal representation is used in the aggregation procedure and how uncertainty formalisms are implemented. The infrastructure usefulness is shown through its use on aggregating evidence regarding software inspection techniques.

Keywords: research synthesis, structured synthesis method, theory building, CASE, evidence-based software engineering

1 Introduction

With the establishment of the evidence-based practice in Software Engineering (SE), one of the main challenges for its further development is how to put knowledge (*i.e.*, produced evidence) to use. This issue is usually addressed under the topic of *knowledge translation*, which has as one of its main concerns the synthesis and application of research knowledge for software development processes improvement taking into account the intricacies of research and professional communication [9].

In knowledge translation, the synthesis of knowledge, the appropriate form for communicating knowledge, and support for interaction between researchers and practitioners are essential to make knowledge reach its users. In Medicine, where knowledge translation was first conceptualized [10] and from which the case for knowledge translation in SE was adapted [9], knowledge users are usually professionals and knowledge translation is done through the application of *guidelines* for extracting *recommendations* from synthesized research.

In our proposal for research synthesis [3], we focus in the way of communicating knowledge by using a diagrammatic evidence representation that can be used for qualitative and quantitative evidence. Our long-term assumption is that a unique evidence format can foster the communication between stakeholders regardless their main interest – research or practice. This assumption is based on three observations:

- The determination of controlled experiments and meta-analysis as 'gold standards' in Medicine seems to represent an important factor in bringing a common understanding on how statistical methods and techniques can support the building of its body of knowledge [11]. This common understanding was beneficial for sharing a common jargon to disseminate knowledge in the area.
- Graphical representations seems to be well accepted by both communities [12] as they are capable of simplifying and aggregating complex information into meaningful patterns. An example from Medicine is the *Forest Plot* diagram.
- A formal representation, with well-defined semantics such as found in statistics, seems to be essential for the organization of any scientific body of knowledge. This is particularly important in the case of evidence-based practice, which is highly dependent on the synthesis of produced evidence. But as scientific contributions usually involve some transformation, expansion or refutation of existing conceptual and propositional networks, any formalization at that level seems to be useful [13].

Still, any research synthesis method can be a complex task whether we consider the amount of extracted information to manage and data to aggregate manually. In addition, the effective communication of evidence and research, in general, can be immensely amplified with web platforms – as is the case of digital libraries and specialized online networks (e.g., www.cochrane.org).

In this work, we present a web computational infrastructure to support research synthesis activities based on the conceptual proposal of a method for aggregation of evidence in SE [3]. The proposal uses a diagrammatic representation created to graphically describe theories in SE and uses belief functions uncertainty formalism. Both topics are briefly introduced in Section 2. The research synthesis method implemented in the infrastructure is presented in Section 3 and Section 4 describes its architecture, design, and some algorithms and formalization used to support evidence representation and aggregation. A worked synthesis example is demonstrated in Section 5 explaining how the infrastructure supports it. And Section 6 enumerates related and future work.

2 Background

2.1 Theoretical Structures

As generally agreed in theories formulation [1], the conceptualization from [2] is defined by constructs (i.e., concepts) connected by propositions (i.e., relations) and a scope defining its boundaries. We choose to use Sjøberg et al. [2] theory conceptualization as it is already tailored to SE and because it defines a visual representation with specific semantics. The proposed representation contains just ten semantic constructs, which is one of the reasons for which we believe in its relative simplicity – but it also can be a limiting factor in its capacity of representing different evidence aspects.

The syntactic diagrammatic representation was derived from [2] and most of the ten semantic constructs are presented in Fig. 1. There are three possible types of structural relationships in the representation: *is a*, *part of* and *property of*. All of them have counterparts in UML, respectively: generalization, composition and class attributes. The *is a* and *part of* relationships use the same UML notation for generalization and composition. Properties are denoted by dashed connections. The relationships are used to link two types of concepts – *value* and *variable* –, which are classified into five subtypes.

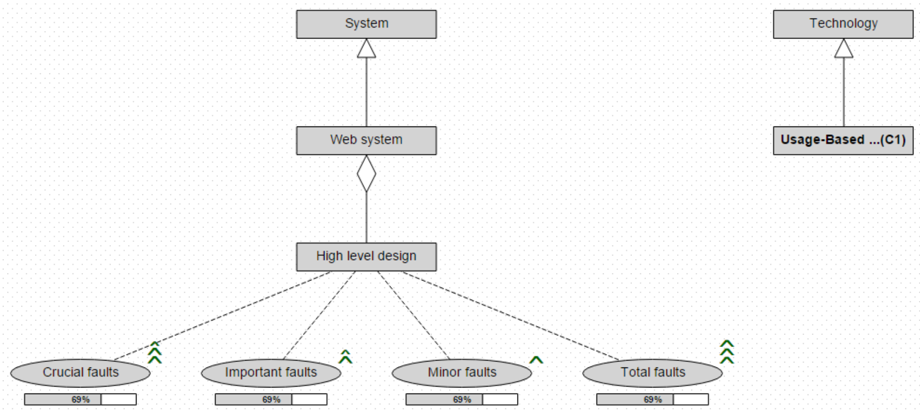

Fig. 1. Partial diagrammatic representation of evidence related to Usage-Based Reading inspection technique [15]

A *value* concept represents a particular variable value, usually an independent variable. *Value* concepts are represented by rectangles and they are classified in *archetypes* (the root of each hierarchy), *causes* (indicated by the use of a bold font and a ‘C1’ following the name denoting that it is the ‘cause 1’) and *contextual aspects* (e.g., ‘web system’). The four archetypes – activity, actor, system, and technology – were suggested by [2] in an attempt to capture the typical scenario in SE described as an actor applying a technology to perform activities in a software system.

A *variable* concept focuses on value variations usually associated with a dependent variable. *Variable* concepts are represented by ellipses or parallelograms symbolizing *effects* and *moderators*, respectively. In addition, *effects* have implicit *cause-effect* relation with its cause and can have *moderations*. To indicate the effect size, we added to the representation a seven-point Likert scale. The scale ranges from strongly negative to strongly positive and is indicated above the ellipse (e.g., ‘’ indicates that ‘minor faults’ are weakly positively affected by ‘usage-based reading’ – ‘’ would be used for strongly positive). The other type of *variable* concepts, *moderators*, indicates that some positive or negative effect is moderated (i.e., reduced) when it increases or decreases. For instance, a common moderator in SE studies is how ‘experience’ influences observed effects. A last aspect related to variable concepts is the association of a belief value (ranging from 0% to 100%) to estimate the confidence in the observed effects and moderations. The ‘bar’ under each element represents the belief value.

2.2 Dempster-Shafer Theory

Theoretical structures by themselves would not be sufficient for integrative synthesis purposes. It was necessary to provide some way to allow evidence aggregation and understanding of what they 'say' together. One of key requirements of the research synthesis method described in the next section is the ability to deal with any kind of evidence. Therefore, it was important to choose an approach that could cope with the inherent *uncertainty* associated with any piece of evidence and handle the *ignorance* (i.e., lack of knowledge) that may come out from more unsystematic observations. For the same reason, it would also be unfeasible to choose a probability theory dependent on any kind of occurrence frequencies (e.g., known distributions). These features are considered one of several methodological advantages of Dempster-Shafer theory [4], which is also regarded for its consistency with classical probability theory, its compatibility with Boolean logic and its manageable computational complexity.

In the mathematical theory of evidence, aggregation is achieved by the *Dempster's Rule of Combination* [4], which takes two pieces of evidence and produces new evidence representing the consensus of the two original pieces. To that end, each evidence is expressed in terms of belief values – using the *basic probability assignment function* – assigned to subsets of propositions of distinct, exhaustive possibilities – called the *frame of discernment*. Uncertainty and ignorance are associated to the *cause-effect* and *moderation* relationships. Therefore, the Likert scale associated with relationships (e.g., 'strongly positive' or 'indifferent') forms the frame of discernment. Additionaly, given the evidence characteristics and its findings, probability is committed to some of these possible qualifications. Further details on how Dempster-Shafer Theory is used can be obtained in [3].

3 Conceptual framework: Structured Synthesis Method (SSM)

As the focus in proposing SSM is in the evidence representation and its formalization, SSM inherits most of its extraction and translation procedures from other methods, which are not detailed here, namely: (i) *thematic synthesis*, (ii) *meta-ethnography*, (iii) *case survey*, and (iv) *qualitative comparative analysis*. These methods share many similar procedures and guidelines indicating how evidence should be extracted from papers and how the synthesis is done. SSM procedures consist in five major steps:

1. <u>Planning and definition</u>: the study objectives are defined, including a research question, and the inclusion/exclusion criteria formalized.
2. <u>Selection</u>: primary studies are collected in systematic manner.
3. <u>Quality assessment</u>: the quality of primary studies is assessed using specific instruments.
4. <u>Extraction and translation</u>: evidence is extracted from studies and translated to theoretical structures.
5. <u>Aggregation and analysis</u>: based on the extracted theoretical structures compatible evidence is aggregated by pooling their effects and moderators. Then, the results are analyzed together.

The first three steps compose the basic arrangement for any systematic literature review. In fact, almost any type of research synthesis depends on this sort of preparation. There are, however, some particular aspects to these initial steps considering SSM. First, the research question in SSM is generally more open, since theoretical structures are able to capture multiple cause-effect relationships and, thus, it does not need to be constrained in this regard. Second, the quality assessment is used as input for estimating the belief values for each effect/moderator of each evidence, which in turn is used as basis for the *Dempster's Rule of Combination*. In the initial proposal [3], the estimation was solely based on two scores schema (or checklists), which are used for unsystematic [5] and systematic [6] studies, respectively. This estimation was refined to include the evidence hierarchy classification from GRADE [7]. In this refined procedure, evidence is first graded into high (randomized controlled trials – RCTs), moderate (*quasi*-experiments), low (observational studies) and very low (unsystematic observations). Based on this, we define four belief ranges: [0.0, 0.25] for unsystematic observations up to [0.75, 1] for RCTs – even though RCTs are seldom conducted in SE it is kept in the scale to make explicit the relatively less confidence in *quasi*-experiments and observational studies. Then, in each range, the total score for each quality checklist is converted to the 0.25 range of each classification level.

The fourth step is heavily based in the four methods previously cited in the beginning of this section. The idea is to follow the basic steps suggested in [2] for describing the concepts, relations and explanations, but also to combine with the guidance used in existing research synthesis methods to extract information from evidence and represent them through theoretical structures. The major orientation in creating the theoretical structures comes from the *thematic synthesis* and its increasing abstraction level (Fig. 2). SSM also contains recommendations from *meta-ethnography* such as how the text should be coded and papers translated into one to another to identify concepts and relations. The inductive approach from *qualitative comparative analysis*, where concepts are identified inductively from the collection of studies, complements these recommendations. To improve the synthesis reliability, the participation of more than one researcher is recommended as is in *case survey* and many other qualitative methods. At last, instructions for identifying cause-effect relationships are also included, since they put qualitative and quantitative evidence in the same perspective in SSM.

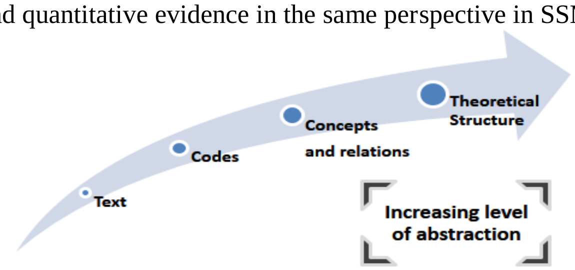

Fig. 2. SSM abstraction levels for evidence extraction (adapted from [8])

The last step starts with the identification of compatible evidence to be aggregated. This is accomplished by verifying whether the theoretical structures 'match'. That is, if all value concepts are equivalent denoting that evidence have similar independent variables and context and, thus, can be aggregated. It is interesting to observe that this procedure can be done iteratively within the step four. Therefore, value concepts that initially seem to be incompatible can be further refined and translated to one another so that new concepts can be developed to capture the findings together. Once compatible evidence is identified, its effects and moderators are pooled and then analyzed.

4 Tool support for research synthesis

All infrastructure's requirements, architecture and design decisions were defined after the SSM proposal and mostly before its construction. Thus, it should be noticed that SSM was conceived to be used independently of tool support like any other research synthesis method. Nevertheless, the method use of a formal model, besides aiming at enhancing the understandability with a diagrammatic representation, was also motivated by its potential straightforward translation to a computational infrastructure.

Although not detailed here, functional requirements are categorized in four types:

- Storage and processing: these are the basic requirements upon which the infrastructure facilities are constructed. They are associated with knowledge formal representation for the theoretical structures (section 2.1), the implementation of the Dempster-Shafer Theory uncertainty formalism (section 2.2), and the support to determine theoretical structures (i.e., evidence) aggregation facilities. In this category, there is also concern with how evidence can be searched, such as keyword based or using a theoretical structure fragment as template.
- Facilities for researchers: include the needs for supporting the execution of a research synthesis, which are basically associated with the five steps enumerated in section 3. The tool shall also maintain all provenance data about whom created the syntheses and evidence instances, besides to preserve the traces among terms and evidence in which they were used or evidence and technical papers from which it was extracted. Collaborative synthesis with more than one researcher is also an important addition for large syntheses processes.
- Facilities for practitioners: the requirements associated with facilities for practitioners define how professionals can take their experiences into and from the computational infrastructure as part of their continuous improvement cycles. This results from our intention that the representation can be used by software engineers. However, this part is out of the scope of this paper.
- Visualization, information provision and social network: this category contains requirements defining the functionalities necessary to display and model evidence in the infrastructure. Additionally, as the body of knowledge in any scientific area is a collective work, knowledge shall be provisioned to and discussed by the community as a whole. Instruments such as *wikis* and *forums* are defined to this end. Furthermore, social mechanisms are expected to favor the establishment of a community as, for instance, the maintenance of the glossary of terms used in evidence or the support to have more than one representation instance for some evidence and letting the community choose the adequate one.

In summary, three main usage scenarios influenced the specification of these requirements: (i) evidence search, (ii) research synthesis control and organization, and (iii) support for continuous improvement activities. The most important non-functional requirements for that end are: (i) be constructed as a Web application and (ii) offer an application-programming interface as web services for the most important facilities, such as search and aggregation, to facilitate tool's integration.

At the current stage, some of the defined requirements are not implemented yet: search using theoretical structure fragment as template, collaborative synthesis, all facilities for practitioners, and most social and informational mechanisms. The infrastructure is implemented as a Web application using the Java programming language and a graph database (Neo4j – www.neo4j.org). It has about 12000 lines of code (excluding web pages and meta-model generated code) in 183 classes, of which 44 are related to domain (research synthesis) concepts and some are listed in Fig. 3.

4.1 Architecture and formal evidence model

The infrastructure was constructed as typical web application architecture inspired on a Model-View-Controller style (Fig. 3). A particularly important design decision was to decouple the knowledge representation model and the uncertainty component from the rest of the system. This allows representations and inferences to evolve independently and was essential to let us first focus on the knowledge evidence representation and then consider how inferences can be obtained from it. This is an indicated strategy for building knowledge-based systems in general [14]. It is also interesting to notice the evidence editor component, which was implemented using web technologies to run on web browsers. Another important feature present in the architecture is the web services API for some systems' information.

As previously mentioned, one of the main components of the architecture is the knowledge representation model and the associated validator. The evidence meta-model, defining the representation abstract syntax, was formalized using Eclipse Modeling Framework (EMF – www.eclipse.org/emf). The model validation is obtained from this formalization. To do that, meta-model classes are instantiated and, then, using EMF programming interface, the validation is executed against the instance. All elements of the concrete syntax described in Section 2.1 and its restrictions are present in the model shown in Fig. 4. Both model concepts and relationships are represented as meta-model classes. The meta-model's classes associations define the evidence model structural restrictions. For instance, variable concepts can only be property of other concepts. This is represented in the model by the link between *VariableConcept* and *PropertyOfRelationship*. It is important to notice the usage of inheritance in the meta-model so that, in this case, the 'fromConcept' of a variable concept is defined in its parent, but its relationship target ('outPropertyRelationship') is defined in *VariableConcept* itself. Another example is the case of structural relationships. The source ('fromConcept') of a structural relationship can be any concept, but its target can only be value concepts ('toValueConcept'). In addition to the structural meta-model restrictions there are two logical restrictions – not shown in the diagram – limiting sub properties and properties that are also type of or part of other concepts.

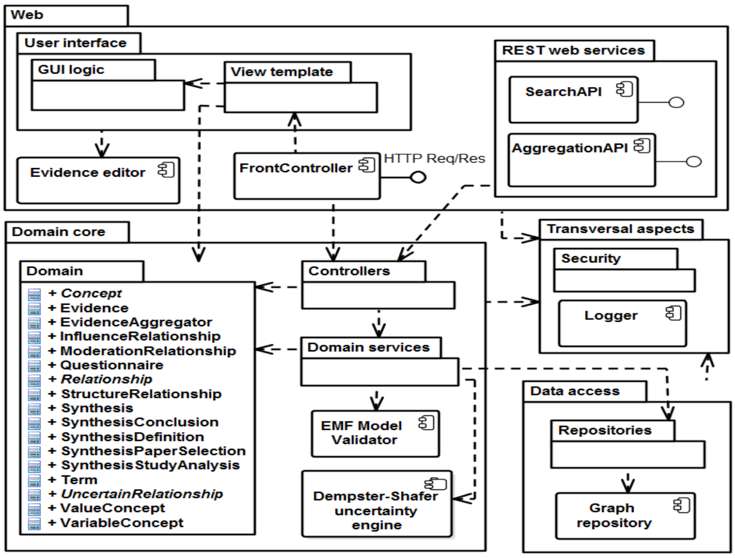

Fig. 3. Infrastructure architecture with its main elements

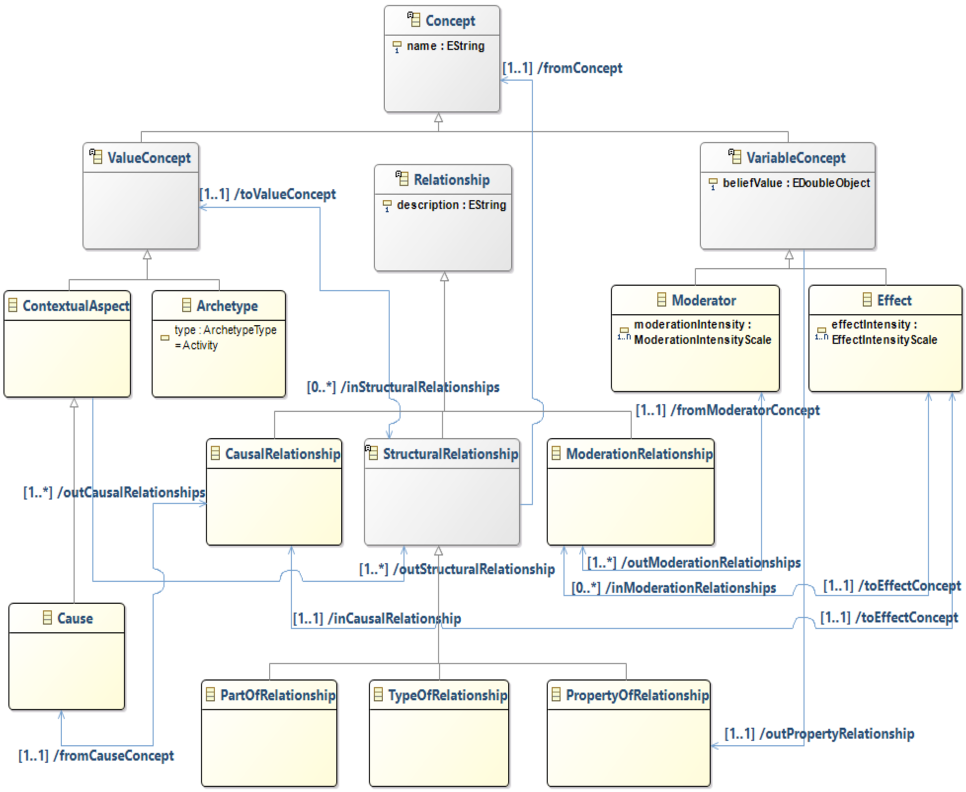

Fig. 4. Evidence representation meta-model

4.2 Navigational structure and supporting facilities

As described in the requirements categories definition, most of the infrastructure functionalities are associated with the SSM method. Fig. 5 shows the navigational structure from which is possible to identify the application main screens and functionalities. In the same Fig. 5 it is possible to observe the three main tool's 'use cases': (1) glossary maintenance, (2) evidence search and aggregation, and (3) synthesis creation.

The glossary contains all term definitions and their synonyms, which are used to detect evidence compatibility. Currently, terms can be defined by any user and cannot have more than one definition. This means that if someone defines 'software quality', it is not possible to have an alternative definition, and it cannot be changed after it has been used in evidence. Evidence search is done by keywords in all parts of its definition (concepts, scope description and explanation detail). From the evidence result list, it is possible to select which one will be aggregated using the procedure described in the next section. The other main use of the tool is the synthesis creation, which is the system part whose functionalities are most influenced by the SSM method orientations.

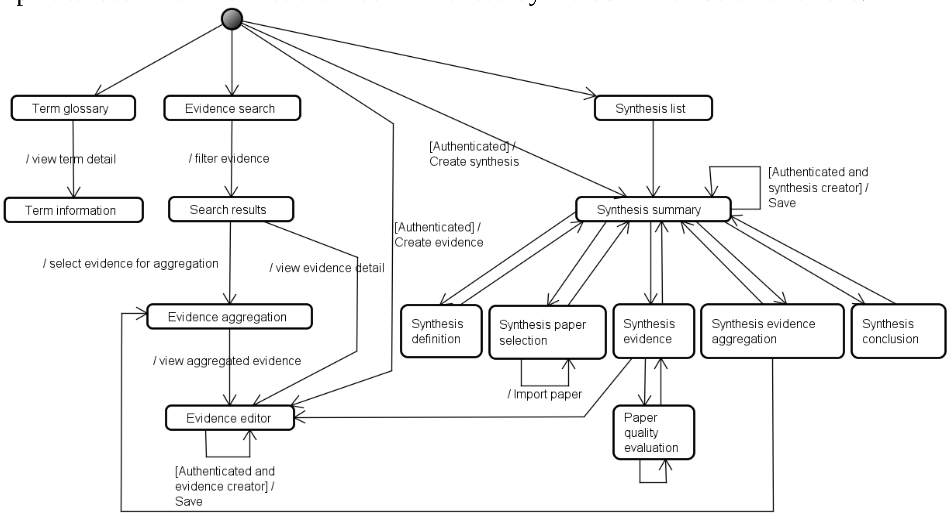

Fig. 5. Infrastructure navigational structure (the start activity symbol represents the accessible menu items and the final symbol was omitted)

4.3 Aggregation conflict resolution

The aggregation algorithm takes as an input the selected evidence and a set of instructions that will guide the resolution process. The algorithm starts with the first two selected evidence, producing an aggregated evidence. It then successively combines other evidence with the previous partial result, until the final aggregated evidence is produced.

The algorithm at first only considers the evidence main structure, ignoring causal and moderation relationships. Then, starting by their archetypes, it generates a tree representing the differences between the two evidence. Each node of the tree keeps record of the presence of a concept in both evidence or not. After differences between evidence are calculated, the algorithm then applies a list of instructions on the resulting data structure in order to solve them. An instruction specifies a pair of each piece of evidence concepts and a resolution for that conflict. A resolution can be one of three: addition, removal and combination. The addition of a node indicates that the concept should be included in the resulting evidence (Fig. 6b). The removal takes the specified node off the tree and places its children under the parent of the removed node (Fig. 6c). The combination of two nodes joins two concepts into one taking one of the two concepts as the new concept definition and adding the children from both concepts (Fig. 6d).

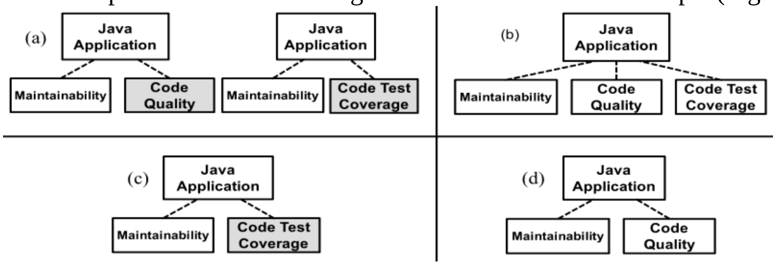

Fig. 6. Possible resolutions for the conflicts on evidence fragment (a): add both concepts (b), remove one (c), or join them (d)

Once all the differences are solved, the resulting difference tree have all related concepts using structural relationships (*is a, part of* and *property of*). This is sufficient information to instantiate the evidence representation model without *cause-effect* and *moderation* relationships. Thus, having the structure of the resulting evidence, the algorithm finally calculates the *cause-effect* and *moderation* relationships combined values.

5 Worked example

In this section, we describe the infrastructure main functionalities through a worked example of a research synthesis regarding Usage-Based Inspection technique [15]. The example uses the same evidence from [3]. However, it is important to notice that some results are different because the belief value estimation has been refined and the criteria for converting the effect intensity from quantitative data are now systematic. The detailed results of the example shown in this section can be accessed at the infrastructure address: http://evidencefactory.lens-ese.cos.ufrj.br – the synthesis name is UBR Synthesis. The name Evidence Factory is a reference to the Experience Factory and symbolizes the place where evidence are constructed (*i.e.*, modeled) and made available.

The infrastructure's first page for synthesis shows a resume of all steps to execute or already executed. From this page, the researcher is able to access all the other pages related to the synthesis in the order that they are defined. The page associated with the study definition (Fig. 7) just register some basic information about it. Paper selection page allows the researcher to import papers and associate them with the evidence using Bibtex format (Fig. 8). In this point, the researcher can also indicate what papers will be included in the synthesis, which in the case of the example are four papers.

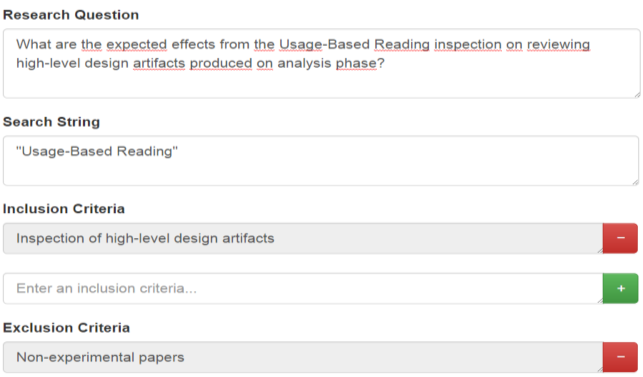

Fig. 7. Research definition for Usage-Based Reading software inspection

Insert new papers: Upload a bibtex file

Title	Authors	Year	Type	Source
A replicated experiment of usage-based and checklist-based reading	Thelin, T. and Andersson, C. and Runeson, P. and Dzamashvili-Fogelstrom, N.	2004	Proceedings	10th International Symposium on Software Metrics, 2004. Proceedings
An experimental comparison of usage-based and checklist-based reading	Thelin, T. and Runeson, P. and Wohlin, C.	2003	Article	IEEE Transactions on Software Engineering
Investigating the effect of expert ranking of use cases for design inspection	Winkler, D. and Halling, M. and Biffl, S.	2004	Proceedings	Euromicro Conference, 2004. Proceedings. 30th
Usage-based reading—an experiment to guide reviewers with use cases	Thelin, Thomas and Runeson, Per and Regnell, Björn	2001	Article	Information and Software Technology

Fig. 8. Paper selection page with the four papers included in the synthesis

The third page offers the researcher the possibility of defining the study type, create the evidence model in the editor and answer the quality questionnaire (Fig. 9). The two quality questionnaires previously cited [5][6] were entirely embedded in the tool, where the research is able to answer them. It is also important to notice that when created at this point, evidence can be searched by any other user and is available to be used in other aggregations. The Fig. 2 already had shown the evidence concrete diagrammatic syntax and how it is rendered in the infrastructure.

Title	Authors	Study analysis
A replicated experiment of usage-based and checklist-based reading	Thelin, T. and Andersson, C. and Runeson, P. and Dzamashvili-Fogelstrom, N.	Quasi Experiment

Fig. 9. Evidence study type definition and links for quality evaluation and modeling

The evidence aggregation page is one of the main system parts. At this moment, the researcher can group compatible or at least most similar evidence. This is done by creating groups and then assigning one or more group category to the considered evidence. In the example of Fig. 10, all evidence were known to be very similar, so only one group 'all evidence' was created. When evidence is not totally compatible, but the researcher understands that is close enough for the synthesis objective, a group can be created and the algorithm from Section 4.3 can be used for their matching. Groups can also be formed to hold most contradictory or divergent evidence, so that differences can also be analyzed and explained in the last step. The aggregation group page also shows the aggregation status of each group using the red color to indicate that the aggregation has not yet been done, green color to indicate that evidence were fully matched or yellow to indicate that the informed resolutions are not sufficient to match evidence. After the aggregation is completed, the generated aggregated evidence is shown in the same way as in Fig. 1 and is stored in the database. The aggregated evidence, then, becomes available to other users.

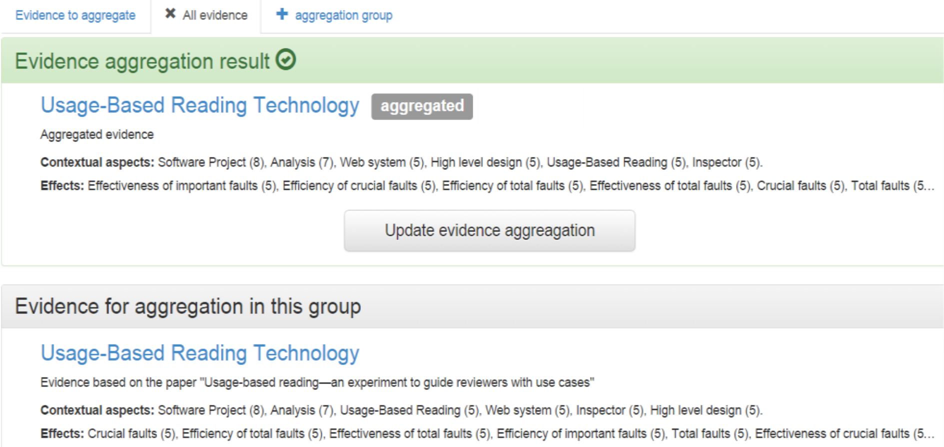

Fig. 10. Aggregation grouping and organization (one of the four evidence is shown)

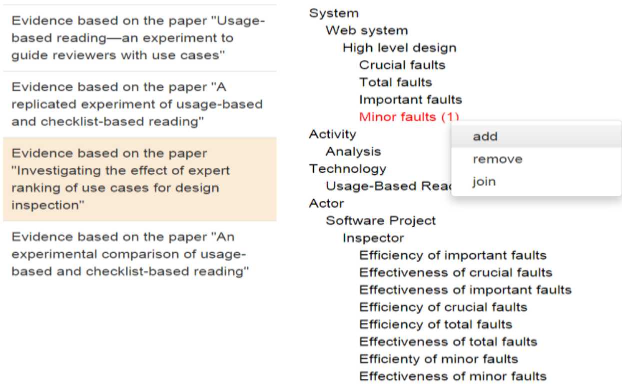

Fig. 11. Evidence aggregation conflict resolution interface and its available actions

In the last synthesis step, the researcher is able to register its analysis and conclusions about the aggregation. This finishes the synthesis with all information structured and traced between aggregation, evidence, technical papers and criteria defined for their inclusion.

As it could be seen in this section, the infrastructure offers support for the key steps of research synthesis. All the infrastructure design decisions were based on the research synthesis method proposed by the authors (SSM). Given the method focus on evidence representation and synthesis, we expect that knowledge translation can be supported and facilitated – at least based on the hypothesis stated on the introduction of this paper. We also assume that the tool can help in this regard since the infrastructure can be thought as a sort of knowledge engineering environment, with support for acquiring knowledge (*i.e.,* modeling evidence) and making inferences from it (*e.g.,* calculating the aggregated evidence confidence or answering questions based on the concepts relations). Although in its initial stage of development, the example intended to indicate its usefulness considering evidence knowledge organization and its synthesis.

6 Related and Future Work

Even though not identifying other tools for research synthesis, we relate to the works of [16] and [17] for their aim at representing SE evidence in a computational environment. In [16], a textual Research Schema Modelling Language is proposed. Differently from this work, in which representation is based on theoretical structures' concepts and relationships, the language constructs and semantics are essentially related to common papers' structure and studies descriptions such as problems, observations, and artifacts. This categorization is what allows the organization of a body of knowledge. The work of [17] uses knowledge engineering as a theoretical base to propose an approach to continuously evolve a representation model based on new knowledge (i.e., evidence) incorporated into the knowledge base. Unlike our Evidence Factory, their proposal depends on the role of a knowledge engineer that is able to prepare knowledge metamodel for new information. Once the information is organized, the authors claim that the knowledge can be synthesized by defining queries against the base.

Future works are essentially associated with the limitations of current stage of this work. Regarding the infrastructure there are several specified functionalities not implemented yet. In addition, we have not conducted any evaluation yet. Our current focus is on the collaborative synthesis, the detailing of Dempster-Shafer computations describing why a proposition of the frame of discernment (*i.e.,* the relationship intensity) was selected, and on developing more facilities related to the step 4 of SSM (Section 3) on evidence diagram editor page. After these short-term improvements, another important research effort will be on experimentally evaluating the infrastructure. We plan to conduct a study about the system usefulness and perceived ease-of-use using the TAM (Technology Acceptance Model) definition of these two variables.

On the other hand, with respect to the SSM and evidence representation, we also intend to perform experimental studies to evaluate its applicability as a research synthesis method. In fact, we already have conducted one study with that end with some positive results and new hypotheses. In that study, ten graduate students used SSM to aggregate four studies about test-driven development. Another important line of investigation will be on identifying limitations of the evidence representation and possible extensions or reformulations. We intend to concentrate on 'low-level' representation model for SE mechanistic explanations.

Acknowledgements

This work has been developed into the context of CNPq (304795/2010-0) ESE project, Mr. Nascimento is a PIBIC UFRJ student. Prof. Travassos is a CNPq Researcher.

References

1. Bacharach, S.B.: Organizational theories: Some criteria for evaluation. Academy of Management Review. 14, 496–515 (1989).
2. Sjøberg, D.I.K., Dybå, T., Anda, B.C.D., Hannay, J.E.: Building Theories in Software Engineering. In: Shull, F., Singer, J., and Sjøberg, D.I.K. (eds.) Guide to Advanced Empirical Software Engineering. pp. 312–336. Springer London (2008).
3. Santos, P.S.M., Travassos, G.H.: On the Representation and Aggregation of Evidence in Software Engineering: A Theory and Belief-based Perspective. Electronic Notes in Theoretical Computer Science. 292, 95–118 (2013).
4. Shafer, G.: A Mathematical Theory of Evidence. Princeton Univ. Press (1976).
5. Shull, F., Feldmann, R., Shaw, M.: Building decision support in an imperfect world. Int. Symp. on Empirical Software Engineering. 33-35 (2006).
6. Kitchenham, B.A., Sjøberg, D.I.K., Dybå, T., Pfahl, D., Brereton, P., Budgen, D., Höst, M., Runeson, P.: Three empirical studies on the agreement of reviewers about the quality of software engineering experiments. Information and Software Technology. 54, 804–819 (2012).
7. Atkins, D. et. al.: GRADE Working Group: Grading quality of evidence and strength of recommendations. BMJ. 328, 1490 (2004).
8. Cruzes, D.S., Dyba, T.: Recommended Steps for Thematic Synthesis in Software Engineering. Int. Symp. on Empirical Soft. Eng. and Measurement. pp.275–284 (2011).
9. Budgen, D., Kitchenham, B., Brereton, P.: The Case for Knowledge Translation. Int. Symp. on Empirical Soft. Eng. and Measurement. pp. 263–266 (2013).
10. Davis, D., Davis, M.E., Jadad, A., Perrier, L., Rath, D., Ryan, D., Sibbald, G., Straus, S., Rappolt, S., Wowk, M., Zwarenstein, M.: The case for knowledge translation: shortening the journey from evidence to effect. BMJ. 327, 33–35 (2003).
11. Booth, A.: Evidence-based practice: triumph of style over substance? Health Information & Libraries Journal. 28, 237–241 (2011).
12. Worren, N.A., Moore, K., Elliott, R.: When theories become tools: Toward a framework for pragmatic validity. Human Relations. 55, 1227 –1250 (2002).
13. Thagard, P., Nowak, G.: The conceptual structure of the geological revolution. In: Shrager, J., Langley, P. (eds.) Computational models of discovery and theory formation. pp. 27-72. Morgan Kaufman, San Mateo, CA (1990).
14. Freiling, M., Alexander, J., Messick, S., Rehfuss, S., Shulman, S.: Startting a knowledge engineering project: a step-by-step approach. AI Magazine. (1985)
15. Thelin, T., Runeson, P., Regnell, B.: Usage-based reading—an experiment to guide reviewers with use cases. Information and Software Technology. 43, 925–938 (2001)
16. Jordan, H., Beecham, S., Botterweck, G.: Modelling Software Engineering Research with RSML. Proc. of the Int. Conf. on Evaluation and Assessment in Soft. Eng.. pp. 44:1–44:10. ACM, New York, NY, USA (2014)
17. Ekaputra, F.J., Serral, E., Biffl, S.: Building an Empirical Software Engineering Research Knowledge Base from Heterogeneous Data Sources. Proc. of the Int. Conf. on Knowledge Technologies and Data-driven Business. pp. 13:1–13:8. ACM, New York, NY, USA (2014).
18. Dixon-Woods, M. et. al.: Synthesising qualitative and quantitative evidence: a review of possible methods. Journal of Health Services Research & Policy. 10, 45–53B (2005).