COPPE
UFRJ
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

EVIDENCE REPRESENTATION AND AGGREGATION IN SOFTWARE ENGINEERING USING THEORETICAL STRUCTURES AND BELIEF FUNCTIONS

Paulo Sérgio Medeiros dos Santos

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro
Dezembro de 2015

EVIDENCE REPRESENTATION AND AGGREGATION IN SOFTWARE ENGINEERING USING THEORETICAL STRUCTURES AND BELIEF FUNCTIONS

Paulo Sérgio Medeiros dos Santos

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

____

Prof. Guilherme Horta Travassos, D.Sc.

____

Prof^{a}. Ana Regina Cavalcanti da Rocha, D.Sc.

____

Prof. Jano Moreira de Souza, Ph.D.

____

Prof. Leonardo Gresta Paulino Murta, D.Sc.

____

Prof. Fábio Queda Bueno da Silva, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2015

Santos, Paulo Sérgio Medeiros dos

Evidence Representation and Aggregation in Software Engineering Using Theoretical Structures and Belief Functions / Paulo Sérgio Medeiros dos Santos – Rio de Janeiro: UFRJ/COPPE, 2015.

XV, 281 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos.

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 217-239.

1. Research Synthesis. 2. Knowledge Engineering. 3. Scientific Theories. 4. Mathematical Theory of Evidence. I. Travassos, Guilherme Horta II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais, pelo amor e carinho. Não por um acaso, sempre colocaram a educação em primeiro lugar. O resultado é este.

Agradecimentos

Em primeiro lugar à Vida, pelo simples motivo de estar com saúde e lúcido para desfrutar desta jornada intelectual que tanto gosto e me desafia.

À minha mãe, Maria do Carmo, que apesar de não estar mais entre nós, é a responsável pela minha sensibilidade emocional e paciência diante dos obstáculos da vida.

Ao meu pai, Heraldo, quem despertou minha curiosidade lógica e racional ao sempre me levar como ajudante dos seus trabalhos como bombeiro hidráulico, eletricista, pintor, carpinteiro, pedreiro, técnico em eletrônica e as dezenas de outras coisas que ele fazia.

Ao meu irmão, Daniel, que mesmo sem entender bem o que eu fazia, sempre admirou o irmão 'nerd' que tem.

Ao meu orientador, Prof. Guilherme, que confiou em mim desde a primeira conversa para Iniciação Científica. Não sabia o que era pesquisa e tão pouco o que era de fato Engenharia de Software. Ao longo de uma década, pude não apenas compreender estas duas coisas à fundo, mas principalmente ter um exemplo de ética, justiça e bom caráter. Um aprendizado para o futuro orientador, que espero poder ser. Pude observar também como extrair o máximo de cada aluno sempre respeitando as diferenças individuais. No meu caso particular, sempre respeitou ciclos relativamente longos sem discussões, mas que me permitiam amadurecer internamente as ideias. Ou a passar confiança mesmo quando tudo não passava de uma ideia 'mirabolante'.

Aos professores Ana Regina, Fábio, Jano e Leonardo por participarem da minha banca de defesa de doutorado. Em particular à Ana Regina e Fábio pelas atenciosas críticas e delimitação do escopo deste trabalho durante o exame de qualificação.

Aos alunos de Iniciação Científica, que trabalharam comigo na construção da Evidence Factory, Luiz Carlos Ciafrino e Ian Nascimento pelas ideias e trabalho dedicado ao projeto.

Ao meus amigos e companheiros da COPPE, alguns já egressos, Jobson Massolar, Tayana Conte, Rodrigo Spínola, Arilo Claudio, Marcos Kalinowski, Victor Vidigal, Vitor Lopes, Marco Antônio, Taisa Guidini, Breno França, Rafael Mello, Karen Nakazato, Rodrigo Santos, Wallace Martinho, Rafael Espírito Santo, Verônica Taquette, Talita Ribeiro, Luciana Nascimento, Thiago Souza, Wladmir Chapetta, Fabio Farzat, Felyppe Rodrigues e Rebeca Motta pelo ótimo ambiente de trabalho e por todos os excelentes momentos vividos durantes esses anos.

Aos meus amigos e companheiros da Petrobras, cito particularmente aqueles que trabalharam diretamente comigo, mas a companhia de todos no ambiente de trabalho foi sempre prazerosa: Cristine Dantas, Amanda Varella, Daniel Borges, Luciane Nobre, João Alonso, Diogo Krejci, Francisco Carvalho, Carlos Barbosa, Leonardo Torres, Rodrigo Maia, Bruno Giovaninni, Alexandre Papanis, Matheus Fernal e Stefano Rodrigues.

Ao pessoal administrativo do PESC, Gutierrez da Costa, Claudia Prata, Maria Mercedes, Solange Santos e Sônia Galliano pelo carinho e atenção.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

REPRESENTAÇÃO E AGREGAÇÃO DE EVIDÊNCIAS EM ENGENHARIA DE SOFTWARE POR MEIO DE ESTRUTURAS TEÓRICAS E FUNÇÕES DE CERTEZA

Paulo Sérgio Medeiros dos Santos

Dezembro/2015

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

Com a consolidação da prática baseada em evidência em Engenharia de Software, a tradução de conhecimento vem ganhando relevância devido à importância da identificação de meios para disponibilizar evidências do estado da arte e da prática de forma a acelerar a captura dos benefícios e limitações destes achados. Entretanto, ainda que seja um elemento essencial para a tradução de conhecimento, métodos de síntese de pesquisa voltados às especificidades da área ainda não foram investigados. Neste trabalho, o Método de Síntese Estruturada é proposto com base no diagnóstico da elevada heterogeneidade dos estudos primários em Engenharia de Software e da dificuldade da comunicação entre profissionais e acadêmicos para um melhor aproveitamento do corpo de conhecimento da área. A fundamentação para a proposição do método é estabelecida com foco na estruturação formal do conhecimento cuja definição é feita a partir da noção de teoria científica e complementada pelo arcabouço da teoria matemática de evidência. Um estudo de viabilidade do método proposto indica a possibilidade do seu uso por pesquisadores da área, enquanto que a condução de dois estudos de síntese reais sugere que o método é capaz de revelar tendências (*i.e.*, efeitos positivos ou negativos) dos resultados presentes em estudos primários. Além disto, a estruturação formal do conhecimento possibilitou a construção de uma infraestrutura computacional que apoia o método. A associação entre método de pesquisa e modelagem de conhecimento emergiu, neste sentido, como contribuição adicional deste trabalho.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

EVIDENCE REPRESENTATION AND AGGREGATION IN SOFTWARE ENGINEERING USING THEORETICAL STRUCTURES AND BELIEF FUNCTIONS

Paulo Sérgio Medeiros dos Santos

December/2015

Advisor: Guilherme Horta Travassos

Department: Computer Science and Systems Engineering

With the evidence-based practice consolidation in Software Engineering, knowledge translation is receiving increased attention due to its importance to the identification of means to accelerate the capture of benefits and drawbacks from evidence of the state of the art and practice. However, although it is an essential knowledge translation element, research synthesis methods addressing Software Engineering specificities are still not available. Thus, we propose the Structured Synthesis Method (SSM) based on the diagnosis of a high heterogeneity of primary studies in Software Engineering along with a common perception of communication issues between practitioners and researchers when trying to leverage from the body of knowledge. The view for the SSM proposition was established upon a formal knowledge representation, which was designed from the scientific theories notion and supplemented with the mathematical theory of evidence. Concerned with the method evaluation, an experimental study indicates that researchers are able to follow the method's guidance whereas the two research synthesis studies conducted with SSM shows its capability in revealing the trends of software technologies benefits and drawbacks reported in primary studies. Furthermore, the definition of a formal knowledge representation for evidence was decisive for the construction of a computational infrastructure supporting SSM. As a result, the association between research synthesis methods with knowledge modeling emerged as an additional contribution of this work.

INDEX

INDEX OF FIGURES

INDEX OF TABLES

1 Introduction

This chapter describes what motivated this investigation and defines the main questions regarding it. The research goals are also enumerated along with the research methodology used to achieve them.

1.1 Motivation

As a community work, Science is highly dependent on methodological and technological instruments to objectively describe and effectively disseminate its knowledge, so that researchers have the opportunity to interpret and exploit it to advance the understanding on different matters. Scientific contributions are usually built incrementally, involving some transformation, expansion or refutation of existing conceptual and propositional networks. As the body of knowledge increases, scientists concentrate more effort on ensuring that new hypotheses and observations are needed and consistent with previous findings. The knowledge accumulation is mainly grounded on the organization and systematization of facts that have some relation with others, aiming at identifying and characterizing patterns of relationships amongst phenomena and processes of the observed world (Overton, 1991).

To practitioners, on the other hand, what forms the body of knowledge is the understanding obtained from observing practical consequences of ordinary daily activities. In their worldview, there is not a genuine interest in obtaining 'the best explanation' to a phenomenon. And, similarly to a pragmatic view, in cases that two explanations lead to the same practical consequences, then they are considered equivalent in some extent (James, 1907). This practical disposition is inherent to most engineering disciplines as 'engineering design is always a *contingent* process, subject to unforeseen complications and influences as the design develops' (Ferguson, 1994). Thus, in Software Engineering (SE) as the engineering can not be formalized as a sequential process that can be summarized in a block diagram, the pragmatic knowledge organization could be reduced to make explicit the expected effects of specific action in a well-defined context.

The apparent paradox between scientific and practical perspectives are in fact complementary and essential to SE – and engineering in general. Although scientists would like that all formal engineering knowledge could be derived from science, it seems intuition use in SE is inevitable (Glass, 2008) granted this subjectivity is carefully used (Strigini, 1996).

The capacity to have knowledge about a subject is directly proportional to the capacity this knowledge can be described in terms of a set of (objective) rules (Cilliers, 2005). Such rules include, for instance, formal mathematical-based models, taxonomies to classify and characterize nature elements, or even diagrammatic schemes to make relations between concepts explicit. This set of rules is usually defined within a representation format, which is an essential knowledge property since it is the means that must be understood in order to correctly inform anyone who is interested in a piece of knowledge. In science, the representation format has major influence for the progress of any discipline as it helps dealing with or even abstracting away the content complexity of the body of knowledge. There are several examples where scientists were able to address or even to discover new problems as a result from insights produced by new understanding revealed by new ways of expressing knowledge. For instance, the intimacy between Mathematics and Physics is well known. The creation and development of modern calculus was mostly motivated by the need of computing areas, volumes or lengths of arcs (Rosenthal, 1951).

Apart from having a *representation* format, a body of knowledge have to be maintained by *activities* that are more or less well defined depending if they are related to maintenance of scientific or practical knowledge. These activities range from actions related to the building and development of knowledge itself (e.g., primary studies and experience reports) to efforts directed to its synthesis (e.g., secondary studies) which usually results in guidelines and recommendations.

Arguably, most recent progression in SE empirical research may be rather concerned with activities on building than representing scientific knowledge. This can be noticed by the explicit focus in the field on introducing and offering guidance for alternative research methods such as case study, survey, ethnography, action research and simulation (Harrison *et al.*, 1999, Wohlin *et al.*, 2003, Zelkowitz, 2007, Easterbrook *et al.*, 2008, Runeson and Höst, 2009, Santos and Travassos, 2011, Mello *et al.*, 2014, França and Travassos, 2015). In fact, this usually happens in most scientific disciplines that concentrate efforts towards the proposition or adaptation of existing research methods, without giving enough attention for the scientific knowledge structure (Rosen, 1996). Nevertheless, although the formalization of scientific knowledge can be sometimes impractical, alternatives to improve its representation should be sought (Suppes, 2001) and may even be associated to the building activities. For instance, the determination of controlled experiments and meta-analysis as the ‘gold standard’ in Medicine (Sackett *et al.*, 1996) seems to have represented an important factor in bringing a common understanding on how statistical methods and techniques can support the building of the discipline’s body of knowledge (Booth, 2011). This common understanding was beneficial for both researchers and practitioners due to the sharing of a common jargon to disseminate knowledge in the area.

As a result of the increasing heterogeneity of primary studies methods in SE and, thus, the difficulty in applying statistical meta-analytical techniques (Dybå *et al*., 2007), secondary studies also undergo over this expansion. The technical literature already has some examples of the application of research synthesis methods other than meta-analysis, such as comparative analysis (Dieste and Juristo, 2011) and meta-ethnography (Silva *et al.*, 2013). Thus, the investigation of alternative research syntheses methods for SE combined with better means for representing evidence in the area, constitutes an important challenge for the long-term evidence-based SE success.

1.2 Problem definition

The main problem addressed in this thesis is in the topic of *knowledge translation*. The problem of knowledge translation was recently discussed in SE (Budgen *et al.*, 2013). The authors highlight the importance of the knowledge translation in evidence-based practice and indicate that it still an open research problem in SE. By adapting the description of knowledge translation of Davis *et al.* (2003), who characterize the problem in Medicine, Budgen *et al.* (2013) define knowledge translation in SE as being ‘the exchange, *synthesis* and ethically sound application of knowledge – within a complex system of interactions between *researchers and users* – to accelerate the capture of the benefits of research through better quality software and software development processes’[1]. Still according to the authors, its three key elements are: the outcomes of a systematic literature review; interpretations of what these mean in particular contexts; and *appropriate forms for communicating them*.

There are different dimensions in knowledge translation each of which with its own activities and concerns. The two most important are knowledge creation and knowledge application. Knowledge creation is related to evidence production, which commonly originate from primary and secondary studies (including research synthesis

[1] Emphasis added.

studies). Domain interpretation is relevant to knowledge creation as well since it is what can lower the barrier for knowledge use. Knowledge application, on the other hand, is associated with activities and definitions that are regarded necessary to put 'knowledge to action' for its users (Graham *et al.*, 2006, apud Budgen *et al.*, 2013), such as creation of guidelines, presentation of knowledge in appropriate forms, and monitoring of knowledge use.

The interest in knowledge translation is grounded on a clear vision that 'knowledge creation (first generation research), distillation (creation of systematic reviews or second-generation research), and dissemination (appearance in journals) are not usually sufficient on their own to ensure appropriate knowledge use' (Straus *et al.*, 2013). Thus, its focus, as a research program, is to identify the supplementary requirements for knowledge translation and develop activities, directives or even services, including specialized institutions or tools, to support knowledge use.

In fact, a significant part of evidence-based practice success in Medicine can be attributed to an infrastructure dedicated for professionals (and other researchers), which provides services concerned with the abstraction and synthesis of scientific advancements, including specialized groups committed to conduct meta-analysis studies and a web platform (http://www.cochrane.org) with advanced evidence search facilities (Kitchenham *et al.*, 2004). Currently in SE, however, in the absence of an infrastructure like this, most professionals tend to ignore evidence in place of expert opinion in most part of their decisions (Rainer *et al.*, 2005, Kitchenham *et al.*, 2007). Furthermore, in the case of practitioners, the lack of a centralized guidance about the best practices causes most part of the state of the practice body of knowledge to be lost or scattered in informal discussion forums or not even come out from software projects where they are conceived and used.

Still, it is undeniable that the adoption of evidence-based practice is conditioned to the involvement of both academics and practitioners. Thus, knowledge translation within evidence-base practice is commonly addressed with the development of knowledge translation *guidelines* showing how to systematically derive recommendations for practice and the provision of *guidance* for organizations on how to deploy the recommendations (Budgen *et al.*, 2013). As a result, this approach focuses on the *steps* necessary to report recommendations and *how* they can be used to inform decisions, which again indicates that attention is more concentrated on *activities* related to knowledge translation than on *representation* formats for it.

In this thesis, we will focus on the later: a representation format for evidence, which we will try to argue that it is an important aspect for knowledge translation. As knowledge translation has many organizational and policy aspects, the scope of this research is related to three more technical aspects of knowledge translation: synthesis of knowledge, appropriate form for communicating the body of knowledge, and support for interaction between researchers and users (professionals and other researchers). It should be noted here, however, that the last aspect – support for interaction – is not entirely addressed in this work as only the researcher-to-researcher cycle is covered. The idea is that knowledge is translated not only for practitioners, but also for researchers themselves in order to, for instance, investigate the state of the art or consider new hypotheses. This is why we put both practitioners and scientists among 'knowledge users'. Moreover, another aspect that forms the problem defined here, and is usually neglected in the knowledge translation discussions, is that knowledge translation exclusively based on reporting recommendations by specialized organizations can be hindered by the huge amount of publications and evidence that are made available every day. In other words, systematic reviews, which usually are used as basis for knowledge translation, tend to get outdated in a relatively short time imposing the constant incorporation of new primary studies' results to the corresponding existing synthesized results. This will be associated with the representation for evidence that we will introduce later.

Based on the knowledge translation research topic described in this section and the context of evidence-based practice in SE, the problem addressed in this thesis is defined as follows. To investigate formal representation for evidence in SE that both researchers and practitioners can use and that can support the body of knowledge to be accumulated and synthesized in such a way that new research hypotheses can be identified and evidence-informed decisions for software improvement can be made in practice.

According to Worren *et al.* (2002), there are three main representational modes for expressing knowledge: (a) *Propositional*, which is the researchers' preferred mode since it favors to explicit the research rationale in the most objective way; (b) *Narrative*, which is the most chosen by practitioners to tell about their daily experiences due to its flexibility in interpretation and application; and (c) *Visual*, which is usually well accepted by both communities given its capacity in simplifying and aggregating complex information into meaningful patterns. Still, despite of some examples such as in Cruzes and Dybå (2011a) for thematic synthesis and in Silva *et al.* (2013) to explicit the results of a meta-ethnography study, graphical representations are not frequently applied to evidence representation and aggregation.

This mismatch in the orientation of how practitioners and researchers perceive evidence usefulness in terms of its representation is fundamental to the discussion of what constitutes an appropriate format for representing scientific knowledge in SE. It is hypothesized in this work that putting industrial experiences and research evidence under the same perspective can make all stakeholders benefit from it regardless it is going to be used to support scientific inquiries or practical decision-making in software projects (Figure 1). So, the first step in moving toward this direction is to expand the concept of evidence, as it is commonly conceived by the academia, and start to accept all categories of evidence from the weak and incomplete (*e.g.*, usually originated from lessons learned and experiences from practice) to the rigorous and well-documented (i.e., commonly produced as result of experimental studies) (Shull *et al.*, 2006). By doing so, software engineers will have the opportunity to make use of evidence even when it is not so "strong" (Shull *et al.*, 2006), for instance, if there are no statistically significant conclusions associated to it. On the other hand, they can have the chance to aggregate the evidence and obtain a "consensual opinion" on the use of software technologies so that insights can be gained from the current body of "weak" (Shull *et al.*, 2006) evidence.

The suggested model in Figure 1 represents a possible conception to address this problem. As previously mentioned, a more common view within the evidence-based paradigm commend the extraction of objective summaries of evidence – usually from systematic literature reviews – about technologies and methodologies, which in turn can influence the practice and can be used as input to new policies and standards (Charters *et al.*, 2009, Budgen *et al.*, 2013). This model seems to have worked out in other disciplines, as the previously cited case of Medicine. However, it appears that it also came from a shared common view of how knowledge should be represented in the field (Booth, 2011) – in the case of Medicine, the 'gold standard' associated with randomized controlled trials and quantitative/statistical hypotheses testing. Moreover, depending on how this model is interpreted, it can implicitly imply some kind of hierarchy between academics and practitioners putting the former as producers of knowledge and the latter as consumers. However, as stated by Barley *et al.* (1988), any view that tries to impose a restricted way of relationship between these communities is subject to a high risk of simplifying the theme to the point of not having practical utility.

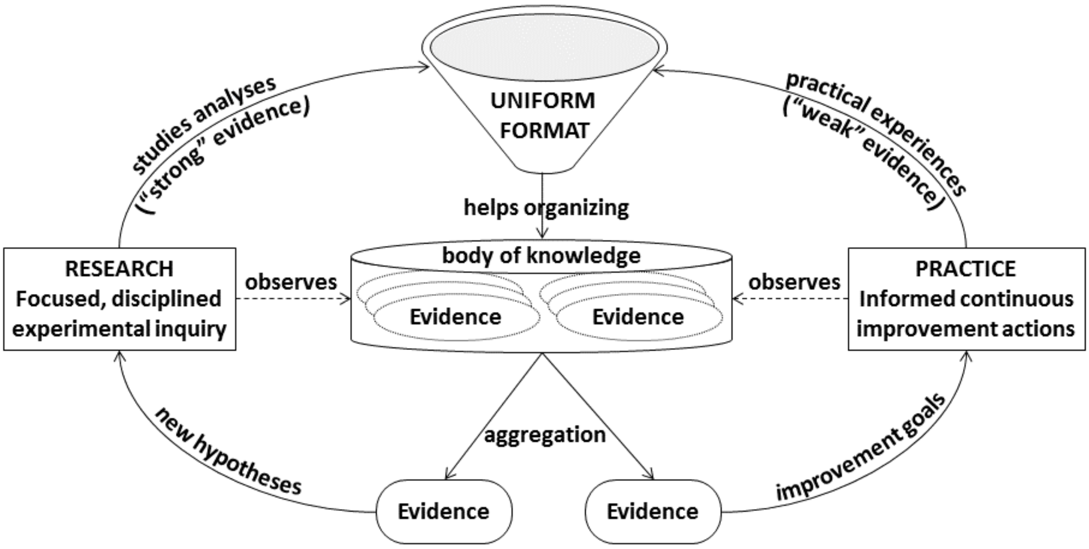

UNIFORM FORMAT
studies analyses ("strong" evidence)
practical experiences ("weak" evidence)
helps organizing
body of knowledge
Evidence
Evidence
RESEARCH Focused, disciplined experimental inquiry
observes
observes
PRACTICE Informed continuous improvement actions
aggregation
new hypotheses
improvement goals
Evidence
Evidence

Figure 1 - A model for evidence-based practice in SE based on a unified evidence representation

As we are addressing a topic related to knowledge, it is critical to state our epistemological view about knowledge before defining our research questions, particularly considering the defined model for evidence-based practice. First, as may be noticed in the initial discussion of this Section, it is interesting to observe that we establish a correspondence between evidence and knowledge throughout this Thesis. Therefore, unless otherwise stated, the terms knowledge or scientific knowledge should be interpreted as evidence. This association is not by chance, as this work is based on the evidentialist theory of knowledge[2]. 'Evidentialism is the view that the epistemic justification of a belief is determined by the quality of the believer's evidence for the belief' (Feldman and Conee, 1985). Evidentialists assume that knowledge is formed through a careful evaluation of facts or experiences, which are used for justifying beliefs. Moreover, because it uses experiences as the basis for justification it is possible to connect it with the empiricism. For Wang (2012), in empiricism knowledge is gained by direct experience and experiment while evidentialism considers the formal analysis of such knowledge to determine if it is valuable.

[2] To define knowledge, philosophers commonly use the tripartite theory of knowledge (http://www.theoryofknowledge.info/). As the name states, this account define that justified, true belief is necessary and sufficient for knowledge. More formally, *S* knows that *p* iff (i) *p* is true, (ii) *S* believes that *p*, and (iii) *S* is justified in believing that *p* (Ichikawa and Steup, 2014).

1.3 Research questions

Regarding the problem described in the previous section, we adopt a dualistic view based on how formal representations can be used both for representing evidence and for organizing the body of knowledge in SE. Given this dualistic view, two research questions are formulated in this thesis:

- **RQ1**: Is it possible to have a formal representation capable of translate (i.e., represent) results of primary studies in Software Engineering?
    - What are the necessary semantic and syntactic (i.e., diagrammatic) constructs to that end?
    - What are the necessary procedures and steps to guide the usage of such representation?
- **RQ2**: Does such formal representation can help organizing evidence in SE by supporting the management, search and aggregation of primary studies' findings?
    - What uncertainty formalism can be used to aggregate qualitative and quantitative studies?
    - How computational support can be designed to that end?

1.4 Research goals

The research goal in its wider scope is to investigate whether a unified view about evidence representation and aggregation in SE can stimulate a closer approximation of the industrial and academic cultures, by establishing a common way of making available the results of empirical evaluations and experimental studies conducted in the field. The idea is that the connection of practical and experimental universes can bring up a shared awareness of the current state of available evidence about software technologies, which in turn may form the basis for a two-way communication between practitioners and researchers.

Interestingly enough, the notion of evidence and evidence-based practice as a whole is not generally agreed among researchers. There is intense debate about what counts as evidence – *e.g.*, only scientific findings? –, what are its main constituent parts and characteristics, what are the evidence-based practice main criticisms, how much interested are practitioners in it, and even the absence of evidence about the evidence-based practice (Upshur, 2000, Upshur *et al.*, 2001, Cohen *et al.*, 2004, Rycroft-Malone *et al*., 2004, Scott-Findlay and Pollock, 2004, Tarlier, 2005, Walters *et* *al.*, 2009, O'Grady, 2012). Since this thesis aims at investigating evidence representation and its synthesis in SE, the last aspect is particularly important as it has a considerable impact on how the goals of this thesis were set. The point in question is that most research methodologies and paradigms are usually not subject to methodological evaluations themselves as they are analytically examined according to the degree of the explanatory power of new results found using them, which is expected to further confirm the explanatory power of the paradigm itself (Guerra *et al.*, 2012). Following this line of argument, it would be unfeasible to set this wider scope as the only goal of this thesis. Thus, the sub goals are not defined as investigating the knowledge translation topic itself, but rather on contributing to identify and justify the necessary building blocks for an alternative view of this research topic:

- Propose (design or adapt) a formal diagrammatic representation for qualitative and quantitative evidence in SE that is at least supposed to be used ("written" and "read") by researchers and practitioners;
- Determine an uncertainty formalism to be used in evidence aggregation that can be applied to estimate the confidence in each piece of evidence and obtain their combined confidence given the findings' conflict and agreement level;
- Explore research synthesis methods' shared characteristics and activities, particularly regarding aggregating both qualitative and quantitative evidence;
- Define a research synthesis method using the proposed formal diagrammatic representation and the selected uncertainty formalism;
- Design and construct a computational infrastructure that supports the defined research synthesis method, facilitates the manipulation of the proposed diagrammatic representation, and automates the uncertainty formalism computations;
- Explore methodologies focused on designing knowledge-based systems and investigate what is the state of the art for this kind of systems in the scientific domain;
- Evaluate the method applicability and utility, both experimentally (*e.g.,* controlled studies) and empirically (*i.e.,* conducting real synthesis studies).

Furthermore, in the absence of consensus about what constitutes an evidence, we adopt a definition taken from dictionary[3]: "something which shows that something else exists or is true". Guyatt *et al.*, 2008 also give a similar general definition: "any observation about an apparent relation between two events constitutes a potential evidence", although it manifest more explicitly the concern with (empirical or experimental) observation. It should also be noticed that these definitions are aligned with the empiricist view adopted as theory of knowledge.

1.5 Research methodology

We describe this thesis research methodology through the Stol and Fitzgerald (2013)[4] Research Path Schema, which indicates the possible emphasis and order given to the main elements present on any research methodology. The Research Path Schema is based on the Validity Path Schema from Brinberg and McGrath (1985) who argue that any research design involves at least three elements: (a) some content of interest, (b) some ideas that give meaning to that content, and (c) some techniques or procedures by means of which those ideas and content can be studied.

In Research Path Schema, these three elements are classified in *substantive*, *conceptual* and *methodological* domains, respectively. Depending on how these domains are combined different paths are established. In this thesis, we follow the *method-driven study design* path. In the *study design* path, the goal is to build a study design based on the *conceptual* and *methodological* domains, and use it on one or more elements of the *substantive* domain (Stol and Fitzgerald, 2015). If the primary interest is on the *methodological* domain, then the study design path is called *method-driven* study design path. Still according to Stol and Fitzgerald (2015), a common scenario found in SE research that follows the *method-driven study design* path is when a conceptual model or framework is taken as basis to develop a method, technique or tool, which is the primary research interest. In this scenario, the conceptual domain works as a lens through which the methodological domain is developed and addressed. As a result, the *substantive* domain is relatively less important since the implementation is the result, which serves as an initial validation of the researcher's proposed idea.

[3] http://www.merriam-webster.com/dictionary/evidence.

[4] Later extended in Stol and Fitzgerald (2015).

As shown in Figure 2, the primary domain investigated in this thesis is the methodological domain. This primary focus of the research methodology is aligned with the research questions previously defined given their design nature. In Easterbrook *et al.* (2008), design questions are put within the *non-empirical* research as they concerned with designing better ways to do something (e.g., research or software engineering). This definition is also aligned with the absence of empirical studies on most research methodologies and paradigms as discussed in the previous section.

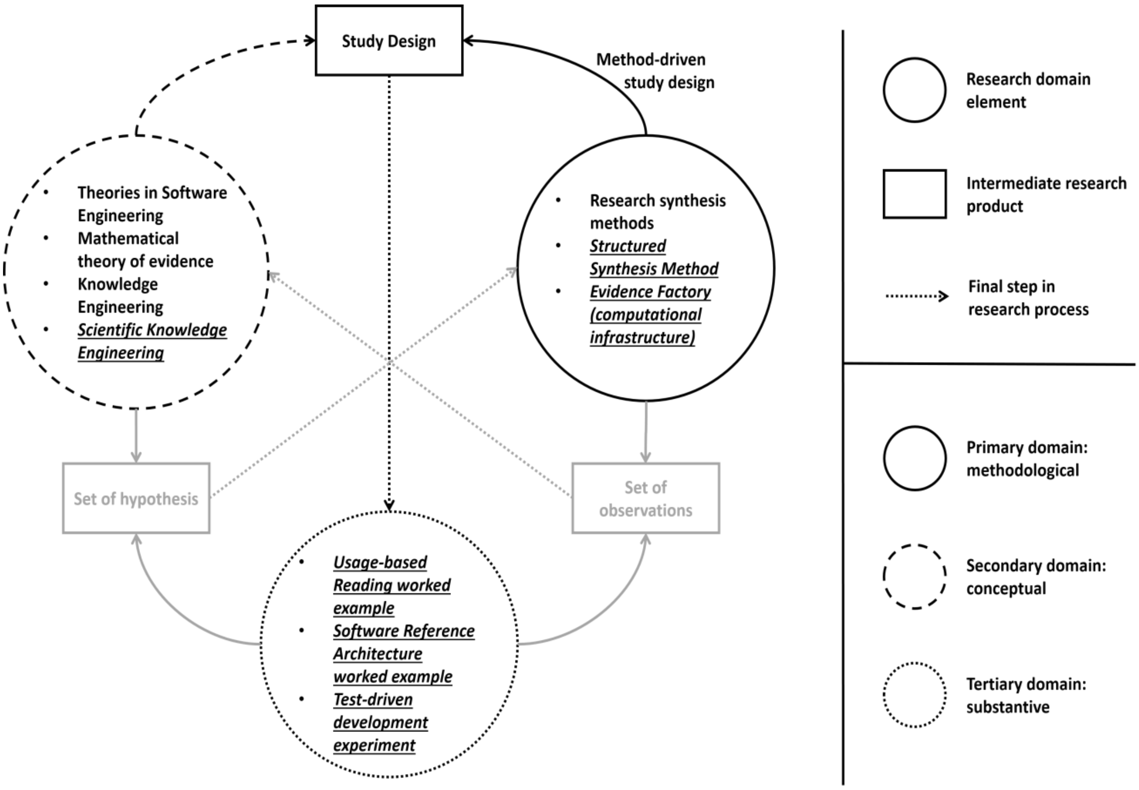

Study Design
Method-driven study design
Theories in Software Engineering
Mathematical theory of evidence
Knowledge Engineering
Scientific Knowledge Engineering
Research synthesis methods
Structured Synthesis Method
Evidence Factory (computational infrastructure)
Set of hypothesis
Set of observations
Usage-based Reading worked example
Software Reference Architecture worked example
Test-driven development experiment
Research domain element
Intermediate research product
Final step in research process
Primary domain: methodological
Secondary domain: conceptual
Tertiary domain: substantive

Figure 2 - Research methodology path and the topics addressed (underlined items are original research presented in this thesis)

Although the methodological domain is the primary domain of this thesis, the first research step begun in the conceptual domain. At the initial phase of the research, it was necessary to delimit the conceptual domain scope from which the methodological aim could be developed. Three conceptual domain topics were explored in order to define the research synthesis method proposed in this Thesis (denominated the Structured Synthesis Method) and design the computational infrastructure to support it (denominated Evidence Factory). Scientific theories form the basis for the elaboration of the diagrammatic representation for theories in SE (Sjøberg *et al.*, 2008), which we formalize in this work as a representation for evidence. The Mathematical Theory of Evidence, also known as belief functions, was selected for evidence aggregation since it does not distinguish qualitative and quantitative evidence and can combine evidence in any order (*i.e.*, accumulate incrementally) (Shafer, 1976). Knowledge Engineering topic (Studer *et al.*, 1998) was also an important foundation upon which we could guide the representation formalization and the computational infrastructure design. Some definitions were also taken from the methodological domain itself, as is the case of research synthesis methods (Dixon-Woods *et al.*, 2005) used to compose Structured Synthesis Method.

Apart from the two main contributions of this thesis in the methodological domain, we also have concentrated efforts on better delineating in what consists Knowledge Engineering in the case of scientific knowledge. We have conducted a literature review on Knowledge Engineering works focusing on scientific knowledge. The idea was also to have a comparison baseline with which the Structured Synthesis Method and the computational infrastructure could be contrasted. Clearly, many other research gaps could be addressed in the conceptual domain such as a comprehensive investigation of other diagrammatic representations for evidence than the used in this research or a thorough analysis of uncertainty formalisms to be used for evidence aggregation. However, we tried to keep attention to the aim of this research in the methodological domain, which is to contribute to identify and justify the necessary building blocks of an alternative view for knowledge translation.

As expected by following the chosen research path, the remaining contributions are empirical investigations to demonstrate the usefulness of the proposal in the substantive domain. The first is an experimental study where subjects were asked to use the Structured Synthesis Method to aggregate evidence from four papers in the Test-Driven Development topic. This experimental evaluation was followed by two worked examples. One related to Usage-Based Reading inspection and the other about Software Reference Architecture. The worked examples were also basis for a preliminary evaluation of the Evidence Factory tool.

1.6 Thesis organization

The thesis is organized following the research methodology described in the previous section. The first three chapters introduces de conceptual domain. Then, chapters four, five and seven are related to the methodological domain. The substantive domain is detailed on chapters six and eight.

It is possible to notice the intersection between chapters related to the methodological and substantive domains. This is because we decided to describe the experimental study immediately after describing the method, then present the Evidence Factory tool which support the method, and only later detail the working examples.

Thus, if readers want to focus on a specific subject, we suggest two reading paths according to the main topics addressed in this work. For the ‘research synthesis path’, readers should read Chapters 2, 3, 5, 6, 8, and 9, while for the ‘knowledge engineering path’, Chapters 4, 5, 7, 8, and 9.

The next chapters are organized as follows:

- **Chapter 2 – Theory building in Software Engineering**: discusses the notion of theories and justifies why it is a relevant topic for evidence representation. In addition, it presents the diagrammatic representation used for evidence.
- **Chapter 3 – The Mathematical Theory of Evidence**: presents the mathematical formalism used in the evidence aggregation.
- **Chapter 4 – Scientific knowledge engineering**: delineates the intersection between knowledge engineering and evidence representation and synthesis.
- **Chapter 5 – The Structured Synthesis Method**: introduces the main proposal of this thesis.
- **Chapter 6 – An experimental study of knowledge translation using theoretical structures**: describes our attempt of a controlled investigation about the method proposed.
- **Chapter 7 – Tool support for the Structured Synthesis Method**: details the design decisions and the facilities of the computerized infrastructure constructed to support the method proposed in this thesis.
- **Chapter 8 – A worked examples**: presents real application of Structured Synthesis Method and discusses the experiences using it.
- **Chapter 9 – Conclusion**: concludes this thesis with final considerations regarding the research questions posed in this chapter and pointing to future works.

2 Theory building in Software Engineering

In this chapter, we explore the theory notion indicating why it has interesting features to be thought out as a representation for evidence. More specifically, a proposal for theory representation is detailed, which will be later used as a formal model for evidence.

2.1 Introduction

Theory building is a research subject concerned with how scientific knowledge is organized and represented to use it to explain real world phenomena. Amongst the most relevant characteristics for investigating theories as a tool for evidence representation, we can cite:

- **Generalist nature**: theories are regarded as a device used to organize a complex empirical world (Bacharach, 1989). In most scientific disciplines, theories support scientists in providing an organized description about some real world phenomenon in a way that part of the real phenomenon complexity is reduced in the representation. For this reason, some researchers (*e.g.*, Suppe, 2000) understand theories as a collection of models for describing the world;
- **Epistemological grounding**: a formal representation depends on a well-defined conceptual basis to be developed. In that regards, it is not difficult to see that the notion of theories is widely discussed in the technical literature. There are entire books (*e.g.*, Reynolds, 1971, L'Abate, 2012) dedicated to the topic on what constitutes a theory, what its main elements are and what paradigms and epistemological views can be used for its construction. The importance of this topic is apparent on several works that use epistemological grounding as an approach to develop knowledge representations in scientific domain (*e.g.*, Hars, 2001, Travassos *et al.*, 2008, Lopes and Travassos, 2009). This rationale led us to expect theories as an alternative for representation aligned with our research goal;
- **Knowledge tool**: 'Theories are practical because they allow knowledge to be accumulated in a systematic manner and this accumulated knowledge enlightens professional practice' (Gregor, 2006, apud Stol and Fitzgerald, 2015);

- **Empirical heterogeneity manageability**: Given its generalist nature, theories can be instantiated within different levels of abstraction (Yang, 2002), which we believe may help linking different findings and putting them in a coherent perspective. This is particularly important in SE where research studies are more heterogeneous with a wide range of research approaches, methods and techniques, both quantitative and qualitative.

It is important to say at this point that, although our purpose is to use the notion of theories as evidence representation, the theory concept is not going to be applied to its full extension. Given its comprehensiveness, there are definitions and usages of theories that are different from the used here, from a wide scope defining it as a tool for describing general laws of nature (Reynolds, 1971) to the specificity of using it to test hypotheses within the positivist quantitative paradigm (Pawar, 2009). Thus, our aim will be more directed to the theoretical structure itself. That is, to the concepts and relations forming a theory. As it will be discussed in this chapter, these are the basic elements of any theory, so we are not making any deviation from its definition, just taking the perspective aligned to the defined goals. The idea of focusing on the theoretical structures is to have evidence organized in a way that aggregation is facilitated. For this reason, we will use the term 'theoretical structure' on any occasion that emphasis is more on the representation than on the 'complete' theory conception. This does not mean, however, that other topics, besides the related to theory structure and representation, will not be discussed in this chapter.

On the next section, we define the theory concept. Then, in Section 2.3 we enumerate and describe theories' main features. A proposal for building theories in SE is presented in Section 2.4. The proposal defines a diagrammatic representation for theories in SE, which we will use for evidence aggregation. To clearly set our view about theories, we discuss different epistemological perspectives in Section 2.5 and examine related works about theories in SE in Section 2.6. Section 2.7 concludes this chapter.

2.2 Definition

It is not trivial to define the necessary and sufficient conditions for what constitutes a theory. Still, despite this difficulty in precisely delimit what forms a theory, in most mature sciences the use of theory tends to be taken for granted and discussions tend to focus on how, rather than whether, to use theory (Hannay *et al.*, 2007). This is because theories tend to facilitate the communication of ideas as they offer a conceptual framework upon which knowledge can be structured in a precise and concise way (Reynolds, 1971).

The concept of theories is broadly used in several scientific disciplines, even though with different formulations (Reynolds, 1971). For that reason, this theme is frequently object of analysis and discussions in Philosophy of Science aiming at the definition of its own terminology, the delineation of the basic conditions for characterizing a theory, and the investigation of how to evolve theories along the time (Suppe, 1977). Some formulations are more abstract (ignoring temporal or contextual aspects) while others are more realist (depending on a more detailed description), but a rather general definition is found in (Gioia e Pitre, 1990, apud Lynham, 2002): ‘a coherent description, explanation and representation of observed or experienced phenomena’.

This definition, although useful for a basic understanding of what can be considered a theory, is excessively generic and do not offer support for practical application. Nevertheless, even if there is not a consensus about theory characteristics in a detailed level, discussions about this topic commonly involves the following issues (Reynolds, 1971, Bacharach, 1989, Lynham, 2002, Walker and Avant, 2004, Gregor, 2006, Sjøberg *et al.*, 2008, Pawar, 2009): (i) its utility and purpose, (ii) its basic elements, (iii) its development levels along its formation, and (iv) how it is evaluated. The next section briefly address each of these issues – more comprehensive discussions can be obtained in the referenced authors.

2.3 Essential theory features

2.3.1 Utility

The utility of theories is associated to the purposes of a systematically organized body of knowledge. According to Reynolds (1971), scientists are usually interested in a piece of theory-based knowledge that provides at least one of the following aspects:

- **Typologies:** also known as taxonomies or ontologies, they offer a form of classification or general organization of a domain allowing to form an initial understanding of a phenomenon or matter. In comparison to other purposes, typologies are usually more easily to be obtained as it basically depends on observation. The difficulty lies in providing a classification that is complete (all ‘things’ are classified) and mutually exclusive (absence of ambiguity on the classification of each ‘thing’) (Reynolds, 1971);
- **Prediction and explanation:** it is possible to say that except from a temporal perspective, predicting future events or explaining what happened on the past represent essentially the same notion. Both perspectives are concerned with the justification of an event. An interesting aspect of predictions and explanations is that a reason can be provided for an (past or future) event without necessarily explaining why this is observed or expected (Gregor, 2006).

Theories combining these two features of predicting and explaining are considered the ‘standard’ conception of theories (Sjøberg *et al.*, 2008);

- **Sense of understanding:** is present in theories that, regardless used for prediction or explanation, completely describes the mechanisms that links the changes in one or more concepts with changes in other concepts. The causal links unveils the causal process that both can explain an event but also can predict it. To that end, it is also necessary the given description to contain the concepts used to organize and classify the phenomenon of interest. Thus, to provide of a sense of understanding theories must also have characteristics of typologies and prediction/explanation. When all these elements are found together in a theory, it can be said that it provides a ‘sense of understanding’ (Reynolds, 1971).

2.3.2 Basic elements

The basic elements that form a theory are (i) a system of constructs linked by propositions and (ii) a scope defining its applicability (Bacharach, 1989).

Sjøberg *et al.* (2008) put the theories’ basic elements into a perspective on how they are used. According to them, theories should support the description of *what* (defining constructs to represent observable or non-observable entities), *how* (determining propositions to indicate the relationships between entities), *why* (providing explanations to describe why the propositions hold), and *where/when* (establishing the conditions under which the theory constructs and propositions are supposedly applicable).

A more detailed description of what elements form a theory can obtained in Pawar (2009). Mentioning Dubing (1969, 1976), Kerlinger (1988), Bacharach (1989), and Whetten (1989), he decomposes a theory into the following elements: *domain*, *concepts*, *variables*, *definitional statements*, *assumptions*, *propositions* and *hypothesis*. The first observation from the enumerated elements is the division of *scope* into *domain* and *assumptions*. This brings the attention to the importance of not only describing the real domain where the phenomenon manifests itself, but also defining the expected requirements or preconditions about the nature of the phenomenon and the way that the entities and their relations can be observed and measured.

Another interesting association is established between *propositions* and *hypotheses*. In Pawar (2009), *hypotheses* are defined as conjectures derived from the observations that can be conceived based on *propositions*. Taking this interpretation to its conceptual limit, some researchers use *propositions* and *hypotheses* interchangeably, even though with a preference to use *propositions* for more comprehensive generalizations while associating *hypotheses* to relatively more specific statements. In addition, *hypotheses* can also be generated in situations where *concepts* are operationalized by *variables* that were not used to measure these *concepts*.

The relation and differentiation between *constructs*, *concepts*, and *variables* is also essential for understanding the theories' basic elements. *Constructs* and *variables* are clearly on different abstraction levels, and usually it is said that a *variable* operationalizes a *construct* (Yang, 2002) – that is, it measures or is used to observe a *construct*. This differentiation is necessary to distinguish terms representing abstract entities or ideas, which cannot be directly measured (*e.g.,* software quality), and terms related to features of the empirical world (*e.g.*, software number of defects). Apart from the more evident distinction between *constructs* and *variables*, some authors also try to establish a distinction between *constructs* and *concepts* as a gradual spectrum of abstraction levels from *construct*, to *concepts* and to *variables*. Still, according to Pawar (2009), it is not uncommon to see the same term labelled as both *constructs* and *concepts*, and even as *variables*. It seems this usually happens when scientists are more interested in using theories as a knowledge tool to organize ideas and structure the phenomenon description than using it as part of the research methodological explanation. Thus, in this work, as attention will be given to knowledge representation and how it can be used to represent evidence, this distinction will not be regarded as imperative and the usage of *construct* and *concept* will be considered interchangeable.

The last basic element enumerated by Pawar (2009) is *definitional statements*. As the name implies, *definitional statements* are simply the collection of definitions and descriptions about the theories' *concepts*, *variables* and *propositions*. This element is commonly absent from the theory definitions found in the technical literature, since it is implicitly assumed to be present in any theory description.

2.3.3 Development

It seems to be consensual among scientists that theories can have different development levels. According to Bacharach (1989), 'implied in the notion of generalizability are different levels on which one can theorize. This implicit continuum stretches from empirical generalizations (rich in detail but strictly bounded in space and/or time) to grand theoretical statements (abstract, lacking in observational detail, but relatively unbounded in space and/or time)'. Therefore, these levels can represent not only milestones in the theory building process, but also 'full' theories by themselves depending on their utility (Sjøberg *et al.*, 2008). Analyzing these development levels, Walker and Avant (2004) distinguish among four levels.

In the first level (level 1) are the theories with strong focus on practical utility, which are also denominated as small-range theories. This kind of theories have concrete relationships, which are directly based on observations, and usually have a causal nature, but do not provide a sense of understanding (Jacox, 1974 apud Walker and Avant, 2004). To summarize, the definitive feature of the small-range theories is their specificity to a particular situation (Higgins and Shirley, 2000). The next level (level 2) is formed by the middle-range theories. For Lens *et al.* (1995), middle-range theories are those sufficiently specific to guide practice and research, and still generic enough for different populations so it can explain similar phenomena. The third level (level 3) categorizes the so-called grand theories, which are highly abstract and have concepts and propositions transcending specific populations or events. And in the last level we find the meta-theories, which focus on methodological and philosophical issues and do not represent theories themselves.

When constructed from the beginning, theories are usually created based on only one initial observation (*i.e.*, evidence), from which its preliminary constructs, propositions and explanations are extracted. From that point, theories that advance throughout the different levels undergo over a continuous refinement process, which involves reiterated development cycles of definition or adaptation of the theory elements based on new evidence obtained in the field (Lynham, 2002). During this process, with the incorporation of new evidence, the confidence on the theories' explanations and predictions increases while, at the same time, the theories become more independent from specific context and population. As a result, theories' explanation coverage is extended over different phenomena, which is precisely the most discernible aspect of theory evolution over the different development levels.

The link between evidence and theories is so important that Higgins and Shirley (2000) try to suggest specific evidence types for each theory development level. For instance, they indicate middle-range theories are usually associated with replications in controlled conditions (*e.g.*, controlled experiment), while small-range theories would be associated with non-systematic observations. This categorization could also point that a possible source of level 1 theories would be practitioners trying to describe their experiences or researchers in initial exploratory studies, while level 2 and 3 would be preferably produced by researchers in more advanced stage of their investigations.

The development process described until now is similar to what Reynolds (1971) denominated as *research-then-theory* approach. As it is possible to presume, the complementary approach is called *theory-then-research*, which consists in the situation where new ideas or hypotheses are analytically described with a theory and subsequently evaluated through observations or controlled studies. Theories constructed in this way are denominated *hypothetical* theories – at least initially while they are not supported by evidence.

Regardless the used approach, *research-then-theory* or *theory-then-research*, the theory development process must be guided by strategies that are suitable to the technical literature type and available data. Walker and Avant (2004) define three strategies for theories development levels summarized here:

- **Synthesis**: used to extract and accumulate constructs and propositions from a data set or pieces of evidence. Synthesis allows researchers to combine isolated pieces of information that are yet theoretically unconnected. Synthesis can work well when a researcher is collecting data or trying to interpret data without an explicit theoretical framework;
- **Derivation**: used in the cases where a scientist wants to transpose and redefine a concept, statement, or theory from one context or field to another. Can be especially useful when there is not theoretical basis in the field for that matter, bringing a theoretical soundness to the explanations. A possible result from these situations is that new understandings or theories can be developed from that point;
- **Analysis**: this strategy allows the understanding of the whole phenomenon with the analysis of its constituent parts. This strategy is usually applied to the cases where there is a solid theoretical basis in the field so that their constructs and propositions can be reformulated to represent the new understanding about the investigated phenomenon.

It is interesting to add here that these strategies are not only used to support the definition of the theories' parts, mainly its constructs and propositions, but also the theory as a whole. In fact, Walker and Avant (2004) describe separately in detail how each of these strategies are used to each theory elements and the theory itself. For instance, in the case of the synthesis strategy, they show how qualitative methods, such as Grounded Theory, can be used to identify concepts and how quantitative methods, such as correlation analysis, to characterize propositions based on a dataset. In the theory level, still related to the synthesis strategy, they indicate the use of graphical representation to facilitate the understanding when making explicit the relationships among the concepts. In SE context, the importance of theory representation can also be found in Sjøberg *et al.* (2008), where the authors mention how it helps to put all evidence related to the theory into the same perspective.

2.3.4 Evaluation Criteria

Theory evaluation is an important mechanism to assess its 'quality level' or how well-formed we estimate it is. There is a wide discussion about how to evaluate a theory and what criteria can be used to that end. Sjøberg *et al.* (2008), citing the works of Bunge (1967), Dubin (1978) and Cohen (1989), enumerate six criteria they consider most relevant to evaluate theories in SE:

- **Testability:** degree to which a theory can be refuted;
- **Experimental support:** degree to which a theory is supported by evidence confirming its validity;
- **Explanatory power:** degree to which a theory explain or predict known observations in a specified scope;
- **Parsimony:** degree to which a theory is economically constructed using the minimum set of constructs and propositions;
- **Generality:** capture the breadth of scope and the degree to which a theory is independent of specific settings;
- **Utility:** degree to which the theory support relevant areas in the state of the practice.

The evaluation criteria have a correspondence with the theories' development levels described in the previous section. For instance, a theory with a high explanatory power, parsimony and generality are usually classified in the third development level. Another example is the hypothetical theories, which have no experimental support. In summary, the evaluation criteria are essential to compare candidate theories to explain the same phenomenon, indicating the strongest given a specific research or practice goal.

2.4 The adopted representation for theories

The proposal for evidence aggregation presented in this thesis has as one of its main parts the diagrammatic theory representation from Sjøberg *et al*. (2008). We choose to use this theory conceptualization as it is already tailored to SE and defines a visual representation with specific notational semantics. Moreover, it is regarded as well suited for mid-range generalizations and it is appropriate to make explicit the underlying phenomenon mechanisms, which are decisive properties for the development of SE theories (Wieringa *et. al*, 2011).

Figure 3 shows an example of a theory represented using the diagrammatic notation proposed by Sjøberg *et al*. (2008). This theory was extracted from a real action research study regarding the use of source code refactoring in a medium-to-large scale Web software project (Santos and Travassos, 2011). The notational semantics are partly based on UML.

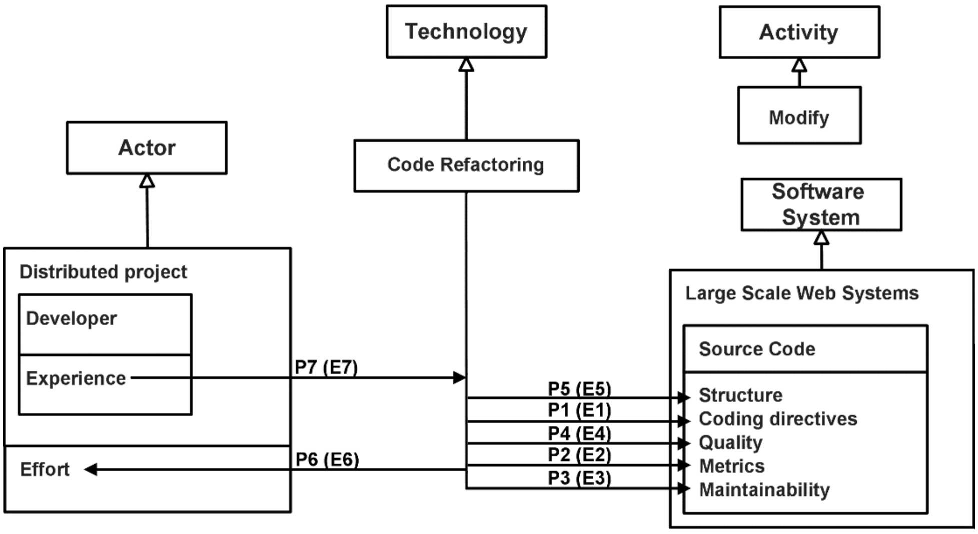


Figure 3 – Refactoring theory diagrammatical representation from (Santos and Travassos, 2011)

A *concept* (or *construct*) is represented as a class or class attribute. A class is represented by a box with its name written at the top such as, for instance, 'Distributed Project'. A class can have a subclass (using the same generalization notation as in UML) or a component class (drawn as a box inside another box such as, for instance, 'Source Code'). Usually, if the concept represents a particular variable value, then the concept is modelled as a subclass or component class (e.g., 'Large Scale Web Systems'). Otherwise, if the focus concerns the values variations, then the concept is a variable modelled as a class attribute, such as 'Effort'. An attribute is placed at the bottom of a class box (below the horizontal line).

A relation representing a *proposition* is drawn with an arrow. An arrow from *concept* A to B denotes that A affects B (*i.e.*, A is a cause to B), where A is a class and B is a property. In addition, B can also be a relationship itself. In this case, A is called moderator, as in the case of 'Experience' *concept*. A *moderator* represents that A affects the direction and/or the intensity of B relationship.

As described, the representation has just ten semantic constructs, which is one of the reasons for which we believe in its relative simplicity, but it also makes us aware about potential limiting factors in its capacity of representing different aspects of theories (and evidence). The ten semantic constructs are separated into five *relationships* and five *concepts* types.

Concept types are named: *archetype*, *contextual aspect*, *cause*, *effect* and *moderator*. The first three types are called *value concepts* as they represent a particular variable value, usually an independent variable. *Archetypes*, the root of each hierarchy, are fixed in four concepts (activity, actor, technology and system) in an attempt to capture the typical situation in SE described as an actor applying a technology to perform activities in a software system. *Contextual aspects* are used to describe the necessary or at least the important conditions under which the stated propositions are expected to occur or were observed. Sjøberg *et al*. (2008) do not give a specific name for this type of concept, but we assign it a name because it delimits the theory scope. *Causes* can be thought as *contextual aspect* subtype as it one addition element present in an observed context, but they are directly responsible for the stated propositions. The last two value types, on the other hand, are called *variable concepts* as they are associated to the aspects that are assumed to have value variations, usually dependent variables. *Effects* are the counterpart of *causes*, and the pair cause-effect are usually propositions that researchers are most interested in. *Moderators*, as the name suggests, are used to define concepts (i.e., moderators) that are understood to influence cause-effect relationships.

Relationship types are denominated: *is a*, *part of*, *property of*, *cause-effect*, and *moderation*. The relationship types can be separated into two categories: *structural relationships* and *influence relationships*. *Structural relationships* (*is a*, *part of*, and *property of*) are used to describe how concepts are 'statically' arranged. For this reason, except for the *property* relationship, which can be used for both value and variable concepts, they are mainly used to organize the *contextual aspects* concepts. Complementing the relationship types, the two *influence relationships* types, *cause-effect* and *moderation*, are the two available alternatives to indicate how concepts can affect each other.

As it can be observed in Figure 3 cause concept is central since it is what defines the hypothesis: 'given the element X (cause) in a given context (value concepts) this is expected (effects and moderators)'. This will be important to establish the aggregation criteria when we define how the representation will be used to that end. Most investigations in SE tries to observe how technologies affects software development actors (persons, teams and organizations), activities (full lifecycles or specific phases), and system (software products, components or other artifacts). Thus, the cause will usually be a type of technology archetype. Nevertheless, it is possible to have any archetype as super type of a cause. For instance, there are investigations concerned with observing how developers' (actor) personality types (cause) affects pair programming (technology) characteristics (effects) (Sfetsos *et al.*, 2008). Still, for being one of the most investigated 'causes' for software development, the technology term is used with different meanings. Sjøberg *et al*. (2008) do not discuss this issue in depth leaving room for different interpretations. In this work, we adopt the definition given by Falbo *et al.* (1998) in their ontology for software development process, which states that software development activities use resources (*i.e.,* tool support) or adopt procedures (*i.e.*, method, technique or guideline). For this reason, we will consider tools, methods, techniques and guidelines as technologies.

Apart from making explicit cause-effect relationships and their moderators, a significant portion of the context in which they are expected to be observed is also described with the representation. The context modeling is performed with the structural relationships where activity, actor, system, and technology are specified (using *is a* relationships) and detailed (using *part of* and *property of* relationships). This is aligned with Thagard and Nowak (1990) who state that these types of relationships are essential to any concept network in science as any scientific revolution consists in, apart from creating and excluding concepts, changing their relationships, particularly, *is a* and *part of* relationships.

To supplement the diagrammatic representation, some textual description must also be provided. It is necessary to textually define each *concept*, describe each *proposition*, and provide the *explanations* of why they can be sustained. These definitions are indexed using the labels 'P' (proposition) and 'E' (explanation) next to the relationships arrows (Figure 3). Some examples of the textual descriptions are given in Table 1. It is important to see that the proposition values are defined in qualitative terms (*e.g.*, Code Refactoring positively affects Maintainability). A complete textual description can be seen in Santos and Travassos (2011).

Table 1 – Some textual descriptions for theory elements from Santos and Travassos (2011)

Concepts	
C1	*Code Refactoring* (development practice the act of modifying software structure without changing its observable behaviour)
C2	*Source Code Structure* (structural properties perceptible in the source code, ex.: readability, algorithm structure)
Propositions	
P5	Code refactoring positively influences code structure
Explanations	
E5	Source code structure improves: • It becomes more homogeneous throughout the entire software project, according to the previous knowledge of the developers. • Its size and complexity is reduced.

As it is recommended to any knowledge representation, a set of activities should be defined in order to researchers be able to model theories using it. Sjøberg *et al*. (2008) present heuristics for theory building using the representation separated into five steps: (i) concepts definition, (ii) propositions identification, (iii) explanations specification, (iv) scope determination, and (v) experimental evaluation. The detailing of these heuristics is not given here, since they are more focused on the notion of theories while in this work we will be more focused in the knowledge representation and research synthesis. This discussion will be focused when we present the proposal of this thesis.

2.5 Epistemological perspectives

There are several views about how scientific knowledge is acquired to and from described by theories. Besides presenting a brief discussion about theories' epistemological perspectives, the idea is to situate Sjøberg *et al*. (2008) proposal among them. This has not been done by Sjøberg *et al*. (2008), but it is examined here as a mean to make some of its limitations and orientations explicit in comparison to other views.

Magnusçl (2012) distinguish among five theory perspectives, two of which are considered the most well-developed by philosophers of science:

- **Statement (or structural):** a theory is considered structural whether it is possible to express it in logical language. Thus, in this view, laws of nature are axioms in a deductive system represented by universal generalizations of the form '*for all x in scope S, ϕ(x) is true*' (Wieringa *et al*., 2011). Furthermore, explanations are theorems in the system, provable from the laws of nature as axioms. Although this view is still being adopted by many scholars, its opponents usually cite the following issues: (i) the representation of theories as linguistic entities, using first order logic, artificially turns the problem of comparing theories into a translation problem; and (ii) logical formulations requires, in most cases, a significant amount of idealized suppositions, which make researchers consider many structural theories distant from real world phenomena, with limited practical applicability. On the other hand, these characteristics, considered by many negative aspects, are exactly the same which make them convenient to design simulation models or arrange controlled studies, as most part of the knowledge necessary to plan these kind of studies is already formalized;
- **Semantic:** this view was proposed as a reaction to the structural perspective. According to Bickle (1993), two were the main motivations for the emergence of the semantic view: (i) scientific theories should not be conceived by the means of language entities (*i.e.*, a set of logical sentences), but rather be described by models; and (ii) the proper tool to formalize scientific theories are not first order logic, but mathematics. As it can be seen in these motivations, theories in the semantic view are defined as a collection of models (Suppe, 2000). Even though emphasis was initially given to mathematical models, currently the semantic perspective accepts other types of models such as the mechanistic

(Glennan, 2005) and the propositional (Thomson-Jones, 2006) – a taxonomy about different types of models can be found in Thomson-Jones (2006). Apart from each type of model specificities, a model in a semantic view is understood as a representation (abstract or physical) of reality. Therefore, differently from the structural perspective where theories are linguistic entities and, thus, have its sentences evaluated as true or false, *models in semantic view are assessed for their degree of similarity to or amount of richness about the modeled phenomenon* (Clarke and Primo, 2007). A question that may be raised at this point is what is the role of theories if models are already assessed comparing them to the real world phenomenon? According to McKelvey (1999), the role of theories in the semantic view is to try to explain the models behavior. Thus, a theory is always associated with and tested through a model, and do not try to explain the real world phenomenon directly. Rather, it describes the behaviors of ‘idealized world’ models.

According to Magnus (2012), the other perspectives are not so articulated as the structural and semantic perspectives. The *cognitive* view is concerned with theories relative to the process of theoretical understanding. A theory is what the agent cognitively grasps, the structure present in the scientist’s mind. Churchland (1992 apud Magnus, 2012), one of the first scholars to think about theories in this way, used the connectionist framework that treats brains as neural networks. In his proposal, he suggests that the codification of theories’ content is embedded in the patterns of neural network middle layers. There are other proposals of theory representation, but all share the notion of theory as the content in the scientists’ minds. The *toolbox* perspective rejects the view of theories as general explanations of world phenomena. Models are the primary unit of analysis, and different theories are put together as tools to construct models (abstract or physical) – Magnus (2012) gives an example of hybrid physical models with parts based on classical and others on quantum or relativistic physics. The *historical* perspective is defined as a distinct period of an existing research tradition. Magnus (2012) cites Kitcher (1984) to describe how researchers think about ‘historical theories’: ‘Classical genetics persists as a single theory with different versions at different times in the sense that different practices are linked by a chain of practices along which there are relatively small modifications in language, in accepted questions, and in the patterns for answering questions’.

In SE, theory epistemological perspectives is discussed to some extent in Wieringa *et al.* (2011). In consonance with Magnus (2012), who put structural and semantic perspectives as the most important and discussed, Wieringa *et al.* (2011) only focus on these two perspectives. Yet, they use a different terminology denominating the structural and semantic perspectives as logical and architectural (or mechanistic), respectively. Indeed, the term logical is commonly used as synonym to structural, since the structural view uses first order logic. However, what they call mechanistic view is not a perspective per se, but rather a specific way to construct models on the semantic conceptualization (Glennan, 2005), as previously mentioned.

Among the different theory epistemological perspectives, Wieringa *et al.* (2011) suggest that the semantic view would be the closest conceptualization to the framework proposed by Sjøberg *et al*. (2008), with direct association to mechanistic models. According to them, this indication is based on the presence of cause-effect relationships and the possibility to represent moderation of these relationships. Together, cause-effect and their moderations are the basic elements to represent most mechanisms among the components of a given system. They allow making explicit that a mechanism (moderation relationship) can neutralize or reduce the effect of another mechanism (cause-effect relationship), which is essential to represent non-universal generalizations (*i.e.*, effects that occur sometimes, but not always). Noting that one of the semantic view criticisms over the structural view is precisely the universal generalizations present in the logical form: '*for all x in scope S, ϕ(x) is true*'.

Even though Wieringa *et al.* (2011) indication of the semantic view to Sjøberg *et al*. (2008) framework seems to be, indeed, correct, it is important to observe that other interpretations would also be acceptable. Taking, for instance, the model taxonomy defined in Thomson-Jones (2006), Sjøberg *et al*. (2008) framework would be categorized among propositional models. In propositional models, as the name suggests, a set of propositions are used to represent systems' characteristics or behaviors. And as described in the previous section, a set of propositions is one of the main characteristics of Sjøberg and his colleagues diagrammatic representation. Still, even if the understanding about which type of model better qualifies Sjøberg *et al*. (2008) framework is an important philosophical exercise, this discussion is not in the scope of this work. Moreover, for the convenience of adopting a classification already indicated in SE (Wieringa *et al.*, 2011) and for being a view that is unifying researchers of different methodological traditions – including quantitative/qualitative and experimental/non-experimental –, we will adopt the conceptualization of mechanistic models, in the semantic view, as the one which best represent the theory framework adopted in this work.

Objectively using these epistemological perspectives and philosophical discussions as a prism to define Sjøberg *et al*. (2008) framework main characteristics, we point the following:

- One of the main Sjøberg *et al*. (2008) framework features is the possibility to represent non-universal generalizations. This is fundamental to form basic, 'local' theories, which are fundamental to SE since general laws in the area are hard to achieve (Hannay *et al.*, 2007);
- As mechanisms are not universal laws, qualifying Sjøberg *et al*. (2008) framework as a mechanistic model in the semantic view would mean that only theories in development level 1 and 2 (see Section 2.3.3) could be represented. However, in situations where the evidence available to support theories' propositions is sufficiently large, it is possible to accept the emergence of theories to level 3. Even if this view is somewhat contrary to the notion of mechanism 'locality', the idea of law as a scientists' consensus about a specific theme, after producing enough evidence, meet an overall understanding of philosophers of science and researchers regarding the differentiation between laws and theories (Reynolds, 1971, Glennan, 1996). This will be precisely our understating on the theories development, since theoretical structures will be formed from research synthesis (*i.e.,* evidence aggregation);

2.6 Discussions about theories in SE

If theory building and theory-based investigations are uncommon in SE (Hannay *et al.*, 2007), even rarer is the discussion focused strictly on the role and the format of theories in SE – though it seems to recently have begun to receive more attention as we will see in this section. Besides Sjøberg *et al*. (2008) framework presented in previous section, we cite four works discussing theories in SE at the meta-level (*i.e.*, regarding some aspect of its necessity, formalization, organization or structuring): Hannay *et al.* (2007), Wieringa *et. al* (2011), OMG (2013)[5], and Stol and Fitzgerald (2015).

The first work is a systematic review of theory use in software engineering experiments (Hannay *et al.*, 2007). Apart from presenting essential theory features, such as discussed in Section 2.3, the authors report some interesting characteristics from 103 papers found in the review, such as the theories' roles in each experiment (e.g., motivation for the study design, experimental testing or new theory generation), their distribution among SE disciplines, and means of representation (e.g., text, logic,

[5] The citation from Object Management Group dates 2013, but SEMAT initiative started at the end of 2009 (www.semat.org).

references or diagrams). After this, some meta-level discussion is presented addressing issues like the usefulness of theories, obstacles for using them, and what constitute a theory in SE. Regarding theories' usefulness, they indicate that theories used in SE experiments provide a conceptual framework for explaining observed phenomena, make investigated research questions more clear and easier to rationalize, and reveal paths for new research since it is common to have parts of theories' underlying mechanisms that have not yet been investigated. Most identified obstacles stems from the perception that it is necessary to accumulate evidence before building theories. However, according to the authors, this is just one of many ways to generate theories. Thus, experiments in SE should begin with theories or, at least, generate a theory so that the theories can be more easily related to experiments. Then, these theories or models can be composed or synthesized together to explain or predict phenomena in different ways that are useful for answering practical questions at hand. This is precisely what the semantic view accommodates and, in the view of Hannay *et al.* (2007), it is the most likely prospect for SE, where smaller units of theory will evolve and address different aspects of a phenomenon.

The proposal from Software Engineering Method and Theory (SEMAT) is not directly focused in theory, but it is in fact a domain-specific language to allow specifying SE methods. Still, the 'theory' in its name stems from a common ground defined to help practitioners to compare methods and make better decisions about their practices. This common ground is implemented through a definition of a *Kernel* whose elements forms a basic vocabulary or a map of SE context (OMG, 2013). The idea is that software development *methods* are composed of *practices*, which are described using *Kernel* elements. The *Kernel* is organized into three discrete areas of concern focusing different aspects of SE: Customer, Solution and Endeavor (software development team). For each one of these areas, the Kernel contains a small number of *Alphas*, which are representations of the essential things to work with, and *Activity Spaces*, which are representations of the essential things to do (OMG, 2013). There are seven *Alphas*, including 'stakeholders', 'software system' and 'team', and fifteen *Activity Spaces*, including 'Understanding Stakeholder Needs', 'Test the System', and 'Coordinate Activity'. With these *Kernel* elements, *practices* can be specified using a combination of *Alphas* and *Activity Spaces*. In addition to the *Alphas* and *Activity Spaces*, *practices'* steps and guidelines are further detailed through the *Alphas'* states and *Activity Spaces'* input, outputs and completion criteria. For instance, in Jacobson *et al.* (2013) a requirements elicitation *practice* is specified using Opportunity, Stakeholders, and Requirements *Alphas*, and using 'Explore Possibilities', 'Understanding the Requirements', and 'Understand Stakeholder Needs' *Activity*

Spaces – all these elements are associated with Costumer and Solution areas. Besides using the *Kernel elements* ‘as is’, practices can be specified by extending the seven *Alphas* and fifteen *Activity Spaces*.

Wieringa *et. al* (2011) describe what constitute *design theories* in SE. The most fundamental characteristic of design theories is they explain effects of artifacts in concrete contexts, that is, as part of an engineering cycle. Four logical tasks are identified as part of any engineering cycle: *problem investigation* of the stakeholders’ needs and measurable criteria for their goals; *treatment design* concerned with solutions designed to operate with stakeholders and artifacts (*e.g.,* devices, software, techniques); *design validation* trying to analyze if theoretical predictions of the treatment design can meet the established goals; and *treatment implementation*, which basically consists of transferring the solution to practice. Based on the description of the engineering cycle, the authors present what should be the structure of *design theories*. In their conceptualization, *design theories* explain why an artifact would contribute to stakeholder goal achievement. The basic idea is that a *design theory* should state what the likely effect E is if an artifact A is used in a context C. It is interesting to notice how the effect is tied up to the use of an artifact in a specific context. This is precisely what is represented in Sjøberg *et al*. (2008) diagrammatic models, where *contextual aspects*, *cause*, and *effect* elements are used to represent $C \wedge A \rightarrow E$ (notation from Wieringa *et. al* (2011), the arrow represent causation, not deduction).

Stol and Fitzgerald (2015) present a different view for theories in SE. The proposed view is based on their diagnosis that, although SE research do not have theories in the way found in other disciplines, most SE technical literature does provide *theory fragments*. To make these fragments explicit, Stol and Fitzgerald (2015) start from the perception that any research study is a result of a combination of three elements or domains: some phenomenon or topic of study, a research method or technique, and a set of concepts or theory. Then, it is presented what they call Research Path Schema, which, as the name suggests, defines different ‘paths’ for research studies. These paths are used to establish how ordered, in terms of importance, the elements should be so that different theoretical focus can be given to each of them. Thus, the argument is that by making explicit how each of these elements are being addressed and what are their order of importance in research studies, then the primary interest domain of the research combined with the supporting domains help researchers identify the theory fragments used or generated. We used the Research Path Schema in this work to help pointing out scientific contributions and its main conceptual and theoretical foundations.

2.7 Conclusion

The notion of theory is pervasive among most scientific disciplines and represent a major building block upon which scientific knowledge can be built. Despite the considerable methodological heterogeneity in SE, and consequent diversity in evidence and scientific knowledge reporting, theory use and generation should not be left aside. In fact, one of the most interesting characteristics of theories is that they can be used in almost any kind of research strategy. Moreover, although different views about what constitutes a theory exist, there is a near consensus regarding its essential features such as the ones discussed in this chapter: utility, basic elements, development levels, and evaluation criteria.

As stated in the beginning of this chapter, we will explore the notion of theories to aggregate evidence in SE precisely for these reasons. The Sjøberg *et al*. (2008) framework was chosen to that end as it has a well-defined (diagrammatic) representation, which is important for aggregation purposes, and is relatively general accommodating almost any type of research examining causality questions. In addition, its semantic view characteristics, particularly of representing non-universal laws, is fundamental in SE where general laws are considered very unlikely, which favor the emergence of short or middle-range theories. Another important characteristic in the context of evidence aggregation is how theories can be structured in the format 'given the element X (cause) in a given context (value concepts) this is expected (effects)' or putting in Wieringa *et al.* (2011) terms Context $\wedge$ Artifact $\rightarrow$ Effect.

In the next chapter, we present the uncertainty framework that will be used in conjunction with the Sjøberg *et al*. (2008) framework. Although both form a conceptual ground for the proposition of the research synthesis method, we still opted to split their presentation in two separate chapters.

3 Mathematical theory of evidence

This chapter presents the uncertainty formalism used for evidence aggregation. Mathematical theory of evidence is a research topic in its own. New applications of the framework are still being discovered and it is still being evolved and extended. Our discussion will be limited to elements necessary for our intended uses, which is basically around the Dempster's combination rule described in the original work of Shafer (1976).

3.1 Introduction

Evidence aggregation process is not only dependent on some kind of representation to determine their combinability, but also on an uncertainty formalism to identify the most important trends considering the analyzed studies. It is important to select an approach that can cope with *uncertainty*, which represents how likely to be true are the findings associated with a piece of evidence, and can handle *ignorance*, which indicates the lack of knowledge about the results that usually come from more unsystematic observations. Particularly important in SE, where there is a considerable heterogeneity of primary studies types, it is critical to use a probability theory independent of occurrence frequencies (e.g., known distributions).

These features are considered one of several methodological advantages of Dempster-Shafer Theory (Ruspini *et al.*, 1992), which is also regarded for its consistency with classical probability theory, its compatibility with Boolean logic and its manageable computational complexity. The Mathematical Theory of Evidence, also known as Dempster-Shafer Theory (D-S) or belief-function framework, is a mathematical formalism used to reason about uncertain events or hypotheses. The main motivation to its proposition was an intent to release the classical probability theory or Bayesian framework from the necessity of assigning an uncertain measure to all hypotheses under consideration (Shafer, 1976). To show the importance of how this can represent a constraint in many situations, the following example is given: three hypotheses A, B and C are possible solutions for a given problem. There is a weak evidence that indicates B as a possible solution (and nothing else). Based on that, using D-S, a low value could be assigned to B, for instance, 0.1 (in a range of 0-1), 0.0 to A and B, and 0.9 of uncommitted support. Representing this using the classical probability theory can be problematic as there is not a direct way to manage the uncommitted support, i.e., ignorance. In these cases, what is usually done under the classical probability theory is to distribute support uniformly trying to represent ignorance (Shafer, 1976). The idea is that when hypotheses have close support values, there is no indication of what is the most probable result, which can represent a kind of ignorance. In the given example, it could be defined as P(A) = 0.3, P(B) = 0.4 and P(C) = 0.3. However, although this can in fact translate the notion of ignorance, this probability distribution could represent a real distribution and not an attempt to represent ignorance. So this kind of ambiguity should be avoided.

In D-S, aggregation is achieved by the *Dempster's Rule of Combination*, which takes two pieces of evidence and produces new evidence representing the consensus of the two original pieces. To that end, D-S defines a *basic probability assignment function* that allows expressing evidence in terms of belief values assigned to different hypotheses and is used as input for combination. The set of the possible hypotheses forms the *frame of discernment*. The result of the D-S combination of two pieces of evidence is also expressed using a basic probability assignment function and, by definition, can be combined with another piece of evidence. To identify the most probable result, D-S defines the *belief function*, which computes the amount of belief that each hypothesis have.

In this chapter, we briefly present D-S theory describing its main constituent parts cited in the previous paragraph. We follow the order in which these concepts are used in the process of aggregation using D-S, which involves defining a *frame of discernment* (Section 3.2), representing each evidence using the *basic probability assignment function* (Section 3.3), combining evidence by the means of *Dempster's Rule of Combination* (Section 3.4), and identifying the most probable hypothesis using the *belief function* (Section 3.5). We also describe the *discount operation* (Section 3.6)*, which* is used to adjust the confidence in evidence depending on how it was obtained. After that, a short example is described (Section 3.7) and concludes de chapter in Section 3.8.

3.2 Frame of discernment

The frame of discernment is a collection of alternatives that evidence can indicate as acceptable to be true, a solution to a problem, possible results of an event, or any other representative state of the real world or theoretical conceptualizations. As D-S uses *set theory* as its mathematical foundation, the frame of discernment is defined as a mutually exclusive and collectively exhaustive set of elements, usually named Θ, with elements {a_1, a_2, a_3, … a_n}.

All operations and definitions of D-S depend on determining a proper frame of discernment. Thus, it is an important step when preparing to use D-S theory as an uncertainty formalism.

3.3 Basic probability assignment function

The basic probability assignment function (bpa) represents the strength of evidence. It generalizes the probability function from the classical probability framework, which assigns a number in the range [0, 1] to every singleton of Θ such that the numbers sum to 1, by extending the assignment domain to the power set of Θ, denoted by 2^{Θ}. By doing this, bpa allows allocating values in the range [0, 1] to every *subset* of Θ. In other words, values can be assigned to all singletons, all subsets of two elements, three elements, and so on, to the entire superset. These values are represented in terms of *m-values*, e.g., m(A) = s, where A is a subset of Θ and s a value in [0,1]. Consistent with the classic probability theory the sum must equals one as well. The m-value for the empty set is zero, i.e., m(Ø) = 0.

In short, bpa is defined as *m*: $2^{\Theta} \rightarrow [0,1]$, where the only restrictions are:

$$\sum_{A \in 2^{\Theta}} m(A) = 1 \text{ and } m(\emptyset) = 0$$

It is important to say that although the 'probability' term on the bpa name suggests m(A) might be a probability, a clear distinction has to be made. First, and most important, is that probability distribution functions are defined over Θ whereas bpa is defined over the power set 2^{Θ}. In addition, bpa does *not* require that (Rakowsky, 2007):

- m(Θ) = 1,
- m(A) ≤ m(B) if A ⊂ B, or
- a relationship between m(A) and its complement m(¬A).

Trying to avoid the misleading about these two different notions, bpa, which was first named by Shafer (1976), received different denotations in technical literature, such as, *basic belief assignment* (Smets and Kennes, 1994), *belief structure* (Denœux, 1999), and *basic belief mass* (Srivastava and Mock, 2002). In this work, we use the original denomination from Shafer (1976). The 'probability' value itself is referred as m-value, mass, belief value or degree of belief.

The quantity m(A) is a measure of the portion of the total belief committed exactly to A. This portion cannot be further subdivided among the subsets of A and does not include portions of belief committed to subsets of A. Here, note that we use the term belief and not probability to refer to the m-values.

A question that could emerge at this point is what represents the allocation of belief to a set that is not a singleton. The answer to this question is not straightforward, as it depends on the problem at hand. The general notion is it represents that there is evidence to believe that one of the hypotheses contained in A is true. Taking a concept from Boolean logic, it could be roughly said it represents an ‘or’ operator of the A set elements. Following this rationale, it is interesting to notice that the quantity m(Θ) is a measure of the portion of the total belief that remains unassigned after commitment of belief to the various subsets of Θ, as m(Θ) represents the belief assigned to the whole frame of discernment. For instance, evidence favoring a single subset A need not say anything about belief in the other subsets. If m(A) = s and m assigns no belief to other subsets of Θ, then m(Θ) = 1 – s. Thus, the remaining belief is assigned to Θ and not to the negation of the hypothesis (¬A), as would be assumed in the Bayesian model.

3.4 Dempster’s rule of combination

Given two bpa’s, both with the same frame of discernment, but related to independent observations, the Dempster’s rule of combination computes a new bpa representing the combined evidence. Interestingly enough, according to Shafer (1976), there is ‘no conclusive *a priori* argument [to the Dempster’s rule of combination]’, but it still ‘does seem to reflect the pooling of evidence’. Analytically, it is regarded appropriated to model the narrowing of the hypothesis set with accumulation of evidence, a process which according to Gordon and Shortliffe (1985) characterizes expert reasoning in general.

The mathematical formulation of the combination rule is shown below:

$$m_3(C) = \frac{\sum\limits_{\substack{i,j \\ A_i \cap B_j = C}} m_1(A_i)m_2(B_j)}{1 - K}, \text{ where } K = \sum_{\substack{i,j \\ A_i \cap B_j = \emptyset}} m_1(A_i)m_2(B_j)$$

The aggregated belief value for each hypothesis C is equal to the sum of the product of the hypotheses belief values whose intersection between all hypotheses A_i and B_j of both evidence is C. When the intersection between two hypotheses is an empty set, we say there is a conflict. Conflicts originate from bpa’s that represent partially contradictory evidence. When this happen, conflict is redistributed to the resulting aggregated hypotheses – that is the function of 1 - K in the denominator.

Different bpa's are represented with indexes next to m, e.g., m_1 or m_2. The bpa representing the aggregation result can use a different index, e.g. m_3, or it can be denoted as $m_1 \oplus m_2$.

One important property of the rule is related to the order of combination. Dempster's rule of combination is commutative (*i.e.*, $m_1 \oplus m_2 = m_2 \oplus m_1$) and associative (*i.e.*, $(m_1 \oplus m_2) \oplus m_3 = m_1 \oplus (m_2 \oplus m_3)$) operation, but not idempotent (*i.e.*, combination of two identical functions is not the same function, i.e., m1 + m1 != m1).

3.5 Belief and plausibility functions

Evidential functions[6] are applied upon a belief mass distribution to evaluate the degree of belief associated with a hypothesis. Arguably, from the two most cited evidential functions (belief and plausibility), the most important is the *belief function*. It is the basic definition that computes how much belief is committed to A or any of its subsets. The idea is to have a measure of the belief that directly supports a given hypothesis A or a more specific (subset) one. In this way, it forms a lower bound of how much is possible to believe in A, i.e., how much belief it is already committed to A or its subsets. Mathematically, a belief function can be written as:

$$Bel(A) = \sum_{B \subseteq A} m(B)$$

The upper bound is defined by the plausibility function. It defines the maximum possible belief that still can be assigned to A. Mathematically, it can be defined as:

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$$

The idea is to compute the maximum share of evidence we could possibly have, if, for all sets that intersect with A, the part that intersects is actually true. Thus, Pl(A) measure the total belief mass that can 'move' into A, whereas Bel(A) measures the total belief mass that is 'confined' to A. In other words, the plausibility is a measure of

[6] It seems that evidential function is not a widely used term in D-S research community, but it can be found in some works, e.g., Rakowsky (2007) and Lowrance *et. al* (2008).

how much is still possible to believe in A, or how much belief is not committed to the other sets of 2^{Θ}. Plausibility function can be simply defined in terms of the belief function: Pl(A) = 1 - Bel(¬A).

A last evidential function, the belief interval function, basically defines the size of the difference between how much belief is already committed to A (i.e., lower bound) and how much can still be committed (i.e., upper bound). It represents the ignorance about A indicating the difference between how much A is believed, i.e., Bel(A), and disbelieved, i.e., Bel(¬A). Ignorance is defined as Pl(A) - Bel(A). Thus, when A is not disbelieved, i.e., Bel(¬A) = 0 or Pl(A) = 1, we have a scenario where all belief not committed to Bel(A) is ignorance. On the other hand, if Bel(A) = Pl(A), it means that Bel(A) + Bel(¬A) = 1, so there is no ignorance about A, but rather a uncertainty about A and ¬A.

As a resume of all evidential functions presented in this section, Figure 4 show how they are interpreted.

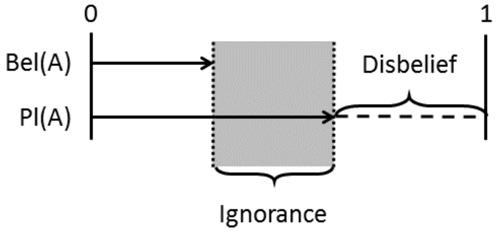


Figure 4 - Evidential functions and their interpretations

Evidential functions' main application is to support determining which hypothesis has most evidence support. There are several rules to determine this (Bloch, 1996). For instance, the maximum of belief or plausibility – usually over simple hypotheses. It is interesting to notice that the definition of a rule to select compound or simple hypothesis is not straightforward. Many aspects affect the selection or definition of the most appropriate rule, including aspects related to the problem at hand. For instance, a basic property of the belief function is that for all A ⊂ B then Bel(A) ≤ Bel(B). Thus, the simple rule of maximum belief must take into account that property, if it is used to select hypotheses sets with different cardinality.

3.6 Discount operation

The discount operation is an instrument to adjust a bpa representing an evidence to reflect the source's credibility. It is commonly used in situations where m-values are obtained from unreliable sources such as sensors, since they are susceptible to many environment conditions. Discounting is performed using a simple multiplicative factor over m-values, expressed as a rate d ∈ [0, 1] where d = 0 represents a completely reliable source and d = 1 a completely unreliable one. Formally, we have $m_{disc\text{-}1}(A_i)$ = [1 – d] x $m_1(A_i)$ for all $A_i \neq \Theta$ and $m_{disc\text{-}1}(\Theta)$ = [1 - d] x $m_1(\Theta)$ + d otherwise.

3.7 An Example

The concepts presented in this chapter can be difficult to understand. This is explicitly stated in Gordon and Shortliffe (1985) where the authors enumerate some problems that according to them hinder a wider adoption of the D-S theory: ‘One problem has been the mathematical notation used in most of the books and papers that discuss it. In addition, the discussions generally lack simple examples could that could add clarity to the theory’s underlying notions.’ In fact, more recently it is not difficult to find papers in which D-S is described using examples, e.g., Parsons (1994) and Srivastava and Mock (2002). Thus, in this section we present a hypothetical example using D-S in SE context. It is important to say that the example description and procedures have no connection with the method for research synthesis proposed in this thesis.

Our example is about a common situation faced by many software organizations. A software team diagnoses their software with severe functional quality issues (i.e., large number of faults). In a project retrospective (Salo and Abrahamsson, 2007), they hypothesize the problem originates from a deficiency in the testing process, as the team is not performing any systematic approach for quality assurance. Given this context, the team enumerates three options for improving their software testing process: (i) manual functional testing (MF), (ii) automated functional testing (AF), and (iii) automated unit testing (AU).

Based on that, the team decides to assign one of its software developers to search the technical literature for indications that could support their decision. The outcome from this research suggested that automation has several benefits over manual testing, considering the team specific context of short development iterations, which would require regression tests at each release. Still unsure about the best choice for their quality issue, the team manager decided to hire a software consultant specialized in software testing. After performing a detailed analysis of the team current practices, the software consultant concludes that functional testing should be adopted, as the team already adopt robust requirement engineering practices, which could support functional testing procedures. In addition, his previous experience based on data from other software projects shows that functional testing tend to deliver the best results in terms of number of defects. This data originates from both manual and automated functional testing, but the consultant personal experiences suggest that teams should start with manual testing to avoid some risks associated with software automation.

We can use D-S theory to combine the two indications if we would want to know the 'combined opinion' from the two sources. The first step to this end is to define a frame of discernment. In this case, we can define the frame of discernment as the set of the three testing approaches, i.e., Θ = {MF, AF, AU}. Then, it is necessary to represent the strength of each opinion using the basic probability assignment function. As we did not set any explicit criteria for defining the belief values in this hypothetical example, we just provide some justifications to determine them. In the developer's case, his findings could be defined as m_1({AF, AU}) = 0.7 and m_1(Θ) = 0.3. The rationale for defining {AF, AU} as a proposition and not the singletons {AF} and {AU} is based on the fact that the findings were associated with testing automation regardless of the test level (unit or functional). The software consultant indication can be modeled defining m_2({MF, AF}) = 0.4, m_2({MF}) = 0.5 and m_2(Θ) = 0.1. The idea in this case is to define belief values to both a compound and singleton propositions so that it could represent the indication favoring functional testing in general, but also expressing the consultant's experiences regarding manual testing for inexperienced teams.

After defining the bpa's, it is possible to use the Dempster's Rule of Combination. Computing the combined bpa $m_1 \oplus m_2$ (or m_3) is a straightforward application of the formula presented in Section 3.4. A schematic detailing of the calculation steps is presented in Table 2. The first line and column cells contain bpa propositions different from zero for m_1 and m_2. The other cells contain the result of the intersection between propositions from m_1 and m_2, in addition to the respective product of their belief values. This organization creates a visual structure of all summation members. The summations are enumerated after the table for each $m_3 > 0$.

Table 2 – Dempster's rule of combination intersection operations

***m*₂** / ***m*₁**	**{MF,AF} (0.4)**	**{MF} (0.5)**	**Θ (0.1)**
{AF,AU} (0.7)	{AF} (0.28)	Ø (0.35)	{AF,AU} (0.07)
Θ (0.3)	{MF,AF} (0.12)	{MF} (0.15)	Θ (0.03)

κ = 0.35 and 1 - κ = 0.65,
$m_1 \oplus m_2$ ({AF}) = 0.28/0.65 = 0.431,
$m_1 \oplus m_2$ ({MF}) = 0.15/0.65 = 0.231,
$m_1 \oplus m_2$ ({AF,AU}) = 0.07/0.65 = 0.108,
$m_1 \oplus m_2$ ({MF,AF}) = 0.12/0.65 = 0.185,
$m_1 \oplus m_2$ (Θ) = 0.03/0.65 = 0.045,
$m_1 \oplus m_2$ is 0 for all other subsets of Θ.

Having defined the combined bpa, we can compute the belief function to determine the hypothesis with the most committed belief:

Bel_3 ({AF}) = m3({AF}) = 0.431,
Bel_3 ({MF}) = m3({MF}) = 0.231,
Bel_3 ({AF,AU}) = m3({AF}) + m3({AU}) = 0.431,
Bel_3 ({MF,AF}) = m3({MF}) + m3({AF}) = 0.662,
Bel_3 (Θ) = 1.

The combined bpa representing the consensus of both opinions suggests that the team should adopt functional testing, preferably automated (based on Bel_3 ({MF,AF}) = 0.662). It is interesting to see how these results are consistent with the given indications about the most appropriate testing approach to use, since one indicated to adopt testing automation whereas the other pointed to the benefits of functional testing. This ‘natural’ result, although not mathematically proved as stated by Shafer (1976), seems to stem from the extension of the probability assignment domain combined with the use of the set theory, particularly the intersection operation. Additionally, its capacity on handling ignorance helps modeling evidence according to the real information that they convey.

3.8 Conclusion

In science, any piece of evidence originates from some kind of observation, either systematic or not. Thus, an important element to research synthesis methodologies is some mechanism to capture scientific evidence uncertainty and ignorance about real world phenomena. In some cases, mostly in qualitative methodologies, this is delegated to the researcher’s analytic skills. For this reason, they are usually criticized for its lack of transparency. In other cases, in the more precise methodologies, usually quantitative, it is based on a rigorous mathematical formalization. And because of that, they usually require several quantitative information that many research strategies do not focus.

In this chapter, we presented the uncertainty framework selected for our evidence aggregation methodology: the Mathematical Theory of Evidence. One of the major drivers for this decision was its balance between a mathematical transparency and dissociation from known quantitative occurrence distributions. Apart from that, it is interesting to mention that D-S theory is applied to several fields such as Medicine, Computer Vision, Accounting, Engineering and Geology. It is usually explored in these areas by the Artificial Intelligence community. The original work from Shafer (1976) has over 13000 citations in Google Scholar, including applications and extension of the framework.

4 Scientific Knowledge Engineering

In this chapter, we characterize Knowledge Engineering as an important groundwork for structuring scientific knowledge. We argue that knowledge-based computational infrastructures can support researchers in organizing and making explicit the main aspects needed to make inferences or extract conclusions from an existing body of knowledge. The current state of the art is presented with 22 knowledge representations and computational infrastructure implementations, summarized with their main relevant properties analyzed and compared. Based on this review and on the theoretical foundations of Knowledge Engineering, a high-level systematic approach for specifying and constructing scientific computational environments is described.

4.1 Introduction

In the beginning of this text, we justified why we believe that a research synthesis method can be an intrinsic part of knowledge translation and why this could be important to evidence-based practice. Although it can be argued that research synthesis methods are not part of the knowledge translation process itself, but rather a preceding step useful for its goals, we have hypothesized that knowledge representation can represent the tangible link between them. In the previous two chapters, we have presented the two conceptual foundations to develop a research synthesis method with a particular focus to the knowledge representation. Thus, in this chapter, we supplement these conceptual foundations with knowledge formalization aspects and its relation to computational infrastructure implementation.

The idea is that knowledge representation not only is important from the point of view of human manipulation and utilization, but also is essential for computational processing, which can be useful to support knowledge translation process (Chapter 1, Section 1.2). Therefore, from the knowledge translation perspective, computational support can help disseminate (e.g., proper knowledge presentation and web platforms) and distillate (i.e., structuring and organizing knowledge to facilitate its analysis and synthesis) evidence. Putting in this way, we are trying to contrast with a previously cited excerpt from Straus *et al.* (2013) that they stated to introduce the importance of knowledge translation: ‘knowledge creation (first generation research), distillation (creation of systematic reviews or second-generation research), and dissemination (appearance in journals) are not usually sufficient on their own to ensure appropriate knowledge use’.

Thus, it can be seem that the benefits of a formal knowledge representation are twofold. One is turning the research synthesis method that uses it part of the knowledge translation process, as the representation is used not only as a tool for the synthesis itself, but also as a ‘language’ to be used by the target audience. The other is to allow the construction of computational infrastructures, which can support evidence dissemination and distillation, and can be particularly important to support the incremental incorporation of new primary studies’ results to the corresponding existing synthesized results by all interested community (academic or professional) members.

In this chapter, we review works that propose or use knowledge representations for scientific knowledge. The goal is to collect and identify the most important aspects involved in formalizing scientific knowledge. We also want to understand what concerns are associated with constructing computational infrastructures to support the organization bodies of scientific knowledge. Based on this analysis, we conceptualize what we dominate Scientific Knowledge Engineering.

4.1.1 Background

The production of scientific knowledge reached such amazing heights in the last decades that better mechanisms to improve the means of its dissemination, interpretation and use are becoming an increasingly relevant issue for the academic community. The concern with this theme has been around for some time (Hars, 2001, Dennis, 2002) and ranges from the management of data generated by scientists in their investigations (Travassos *et al.*, 2008, da Cruz *et al.*, 2009; Maccagnan *et al.*, 2010) to the focus on the searching of formalisms to represent and reason with scientific knowledge (Hunter and Liu, 2010). Common to most initiatives of this kind is an intensive computational manipulation of scientific knowledge, be it in the application of ‘data-driven’ analysis techniques to explore patterns from which new interpretations can be induced (Newman *et al.*, 2003), in attempting to simulate the creative processes employed by researchers in the course of scientific discovery (Džeroski *et al.*, 2007), in the support of researchers activities — for instance, in providing provenance for experiments conducted in scientific workflows (da Cruz *et al.*, 2009) —, or even in assisting the search for scientific studies and results (Bechhofer *et al.*, 2013).

Traditionally, the main means of scientific knowledge dissemination and consumption have been through research articles written in prose. In fact, this format arguably has not undergone significant change since Guttenberg's mechanical printing invention in the 15th century. Clearly, the advent of the Internet and digital media in general represent a major improvement, having brought many benefits. However, even with these advances, research articles are still basically a straightforward digital metaphor of the paper printing model.

In spite of the considerable attention that researchers put to make knowledge embedded in their publications precisely and objectively interpretable, the textual format shows signs of its limitations when we consider the volume of produced research. One of the main limitations is that it requires researchers to carefully hand pick information on papers to make the dissemination and consumption of scientific knowledge possible. Moreover, the information collected still needs to be thoroughly analyzed by identifying what is comparable, searching for similar patterns and comparing differences among parameters, to produce the desired interpretations, abstractions, and generalizations.

As it turns out, the challenge is how to make scientific results a ‘first class citizen’ in the digital era by making it understandable not only by humans but also by computers, supporting researchers in better exploiting the scientific body of knowledge. In this chapter, we suggest that the discipline of Knowledge Engineering (KE) offers a solid foundation upon which this vision can be constructed. One of the main points in proposing this is that Scientific Knowledge Engineering (SKE) can emerge as an area with its own issues and concerns. This somewhat resembles what happened, for instance, in the case of business workflow management systems and scientific workflow management systems (Barga and Gannon, 2007) where many particular issues and concepts were identified (e.g, system architecture (Lin *et al.*, 2009) and provenance (da Cruz *et al.*, 2009)), but also directly translated from the business to the scientific domain.

Thus, the goal of this chapter is to present a conceptual construction of SKE based in a theoretical correspondence with Knowledge Engineering as a relatively more general area and an analysis of the state of the art current proposals and implementations. To cover this objective the following organization was defined. The motivation for SKE and the possible alternatives are presented in Section 4.2. Plus, although not yet recognized as an area, its essence is implicitly present in many recent works used to characterize the state of the art in Section 4.3 literature survey. Based on that survey, Section 4.4 details the fundamental aspects of SKE and presents a step-by-step approach to start new projects. Finally, in Section 4.5 the final considerations are discussed, including work limitations and future work paths. As a final remark, this chapter is mostly based on the article Santos and Travassos (2015).

4.2 Motivations for Scientific Knowledge Engineering and its alternatives

The main idea associated with SKE is the elaboration of formal representations which allow modelling scientific results in computational infrastructures[7] built mainly to execute some kind of scientific inference or extract a conclusion. In SKE, the translation of scientific results into knowledge representations is fundamental to allow the organization of a more precise body of scientific knowledge consisting of facts, rules, approximated representations, and conceptualizations of observed phenomena (Lenat and Feigenbaum, 1991, Studer et al., 1998). It is also what could enable the body of scientific knowledge to be the target of automated or semi-automated computational reasoning.

This definition is basically a straightforward extension of the KE definition to the scientific domain — and more specifically an extension of the paradigm of KE as a modelling process[8] (Wielinga et al., 1992, Ford, 1993, Studer et al., 1998). KE, in this paradigm, can be defined as the set of methods and techniques for knowledge acquisition, modelling, representation and usage (Schreiber, 2000). It essentially consists of two main stages (Dibble and Bostrom, 1987): knowledge acquisition and development of knowledge-based systems. And in almost all methods at least two actors are present: the domain specialist and knowledge engineer.

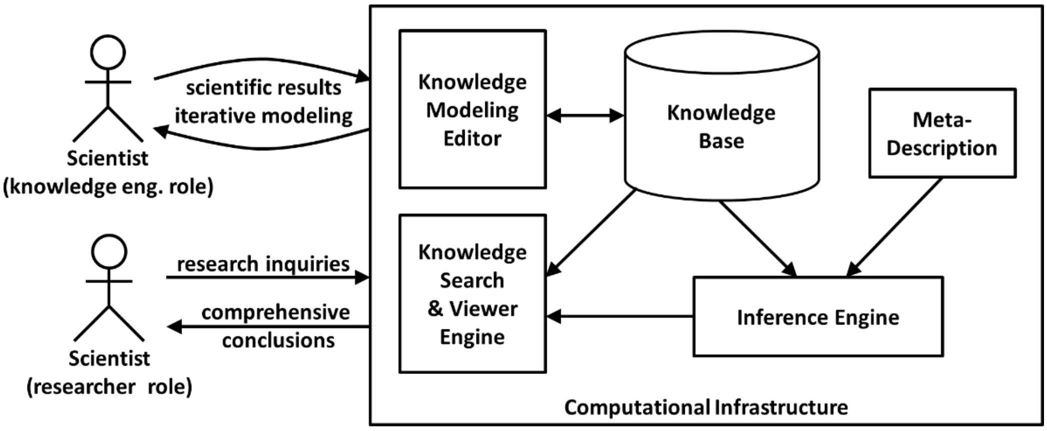
In KE (as a modelling process paradigm), the acquisition stage undergoes significant changes in comparison to the transfer process paradigm. In this paradigm, knowledge acquisition is viewed as a modelling activity where the knowledge-based system is not only filled with knowledge extracted from a specialist by a knowledge engineer, but also designed as an operating model which has a particular behaviour, given a set of specific conditions (i.e., knowledge) (Wielinga et al., 1992). In some cases, in seeking to support the knowledge engineer work or allowing domain

[7] We use the term computational infrastructure instead of knowledge-based system or expert system, as we believe that it better represents the KE application to the scientific domain and draws attention to the fact that it is not its intent to replace the scientist expertise but to augment it. It is also consonant with Hars (2001) which uses the term Scientific Knowledge Infrastructure.

[8] The alternative paradigm is the transfer process paradigm (Studer et al., 1998).

specialists to carry out some knowledge engineer functions, knowledge acquisition tools can be created (Eriksson, 1992). To this end, however, it is necessary to build a generic model (i.e., meta level) that prescribes what domain knowledge is required to enable the knowledge-based system to perform its functions (Wielinga et al., 1992).

In SKE, KE is focused on the design of generic knowledge representation models and on the construction of a computational infrastructure which, besides other facilitation mechanisms, should provide functions similar to knowledge acquisition tools. In this perspective, scientists assume the dual role of domain specialist and knowledge engineer as, in addition to have the technical expertise in a scientific domain area, they are responsible for modelling scientific results into the computational infrastructure that allow the obtaining of intelligent answers from this knowledge. Figure 5 shows a typical organization of SKE computational infrastructures.



Scientist (knowledge eng. role)
scientific results
iterative modeling
Scientist (researcher role)
research inquiries
comprehensive conclusions
Knowledge Modeling Editor
Knowledge Base
Meta-Description
Knowledge Search & Viewer Engine
Inference Engine
Computational Infrastructure

Figure 5 – A typical organization in a SKE computational infrastructure

The designing and building of a computational infrastructure requires the undertaking of a comprehensive knowledge-level analysis of the problem at hand. The analysis involves an exhaustive investigation of the potential formalisms for knowledge representation of the domain explored — in this case, the scientific domain and its disciplines — and the search for the appropriate inference methods. Once the computational infrastructure is built it can be used to instantiate models of the knowledge representation which will form the knowledge base. And from this knowledge base the specified inferences are used to exploit the stored knowledge.

Even though the modelling of scientific results into knowledge bases is central to SKE, it is important to see that the proposition of SKE does not have to be restricted to this aspect. This is because the successful development of knowledge-based systems involve many other factors (Freiling *et al.*, 1985, Fellers, 1987, Dibble and Bostrom, 1987, Rook and Croghan, 1989, Motta *et al.*, 1990, Plant, 1991, Studer *et al.*, 1998, Schreiber, 2000). This extensive technical literature on KE processes and procedures form a solid basis upon which SKE can be systematized to guide one through most of the relevant technical decisions made in specifying and building scientific computational infrastructures.

Thus, having KE as a foundation for SKE can represent a sound way to address the increasingly amount of scientific results with the use of knowledge representation formalisms and automated reasoning techniques. Examples of the answers that can be obtained in this vision, which cannot be directly extracted by the plain digitalization of scientific articles and their availability in digital libraries search engines, include: (i) what is the current state of the art associated with a particular research question? (ii) how is concept X defined and how does it relate to other concepts in a discipline? (iii) in what contexts is a specific technology/procedure/methodology/intervention more utilized? (iv) what are the experimental variables most used to evaluate technology X? (v) was prediction X observed in any study? (vi) is there any contradictory result to study X, what is it, and what are the differences? (vii) which model can generate the best results for problem X and how does it compare to other models?

Arguably, all of these questions can be investigated by searching digital libraries using keywords and doing a manual analysis of the pertinent results found. In fact, it should be said that SKE should not aim at the elimination of the textual publication form. On the contrary, its purpose should be to supplement the textual format, as prose is a rich form of communication that is required for many kinds of analysis and interpretation — as, for instance, is the case of argumentative articles which do not focus on putting forward direct results of scientific investigations (Mons and Velterop, 2009). The central issue, however, is that manual search of technical literature in addition to the interpretation and analysis of huge amounts of information demands great effort to maintain the scientific body of knowledge continuously and consistently evolving.

For instance, the work of Dinakarpandian *et al.* (2006) allows researchers to model research outcomes as assertions following a strict format in the form of Subject-Predicate-Object. This basic ‘triple’ is further detailed in a formal grammar which captures other characteristics related to the context, such as the object quantity qualifier, a place qualifier among other information. Based on this formalization, a knowledge based can be searched using inferences based on the relations among entities. In another example, Santos and Travassos (2013) propose to use a diagrammatic representation for theories as a mean for evidence representation and aggregation. The authors describe how researchers model scientific results with the diagrammatic representation. Based on the structural and semantic comparison of the models, an aggregated model is derived combining the uncertainty associated to the model propositions. These examples show how SKE focuses on knowledge transformation to formal knowledge models to support researchers in making enhanced inferences from the scientific body of knowledge.

Therefore, to better distinguish SKE, the sections below make a brief introduction and compare similar approaches in the use of computational methods to scientific applications. Just as a remark, although data curation has some intersections with SKE and the other approaches presented in this section, it was excluded from the comparison. Data curation is defined as the activity of managing and promoting the use of data from its *point of creation* (Lord *et al.*, 2004) (i.e., raw data (Hunter, 2008)). Thus, in contrast with SKE, the purpose of data curation is to organize any type of data or information produced in research activities (and not only scientific results). Furthermore, SKE is more focused on knowledge representation and the development of computational infrastructures, whereas data curation is geared towards data availability, archiving and preservation (Lord *et al.*, 2004).

We should also point that semantic publishing was not considered amongst the approaches either as, in spite of its similar strategy to make researchers formalize knowledge, it focuses more on publishing aspects such as live DOIs and hyperlinks, interactive figures, semantic mark-up of textual terms with links to further information, a re-orderable reference list, citations in context (using a supporting claims tool tip), and tag trees (Shotton, 2009). In addition, initiatives to support research reproducibility such as databases (e.g., http://db-reproducibility.seas.harvard.edu/), although its usage of computational support to organize experiments' datasets, are more focused on transparency and auditability than on understanding and interpreting results.

4.2.1 Data intensive approaches ("big data")

One of the alternatives that has been quite explored to support scientific investigations is what can be named data intensive approaches or what, in many cases, is also known as "big data". Many disciplines apply data intensive approaches to do science, including High Energy Physics, Earth and Environmental Sciences, Bioinformatics, Astronomy, and Astrophysics (Fiore and Aloisio, 2011). In data intensive approaches, useful data is commonly and already find in digital format, usually originating from sensors, satellites, or scientific data repositories (e.g., gene and protein archives) (Hey and Trefethen, 2003). From these sources, data is computationally processed to support scientific investigation using, for instance, data mining techniques (Fayyad and Stolorz, 1997) or simulations in scientific workflow management systems (Deelman *et al.*, 2009).

The differences between data intensive approaches and SKE can be summarized by the characterization of the moment and way in which computational resources are employed (**Figure 6**). Data intensive approaches can be characterized by the use of relatively more raw data and usually aims at support identifying patterns in a huge mass of data. On the other hand, SKE focuses on using more elaborated data (i.e., scientific results generated as a final step of investigations) and modelling individual results to be collectively processed later with automated reasoning.

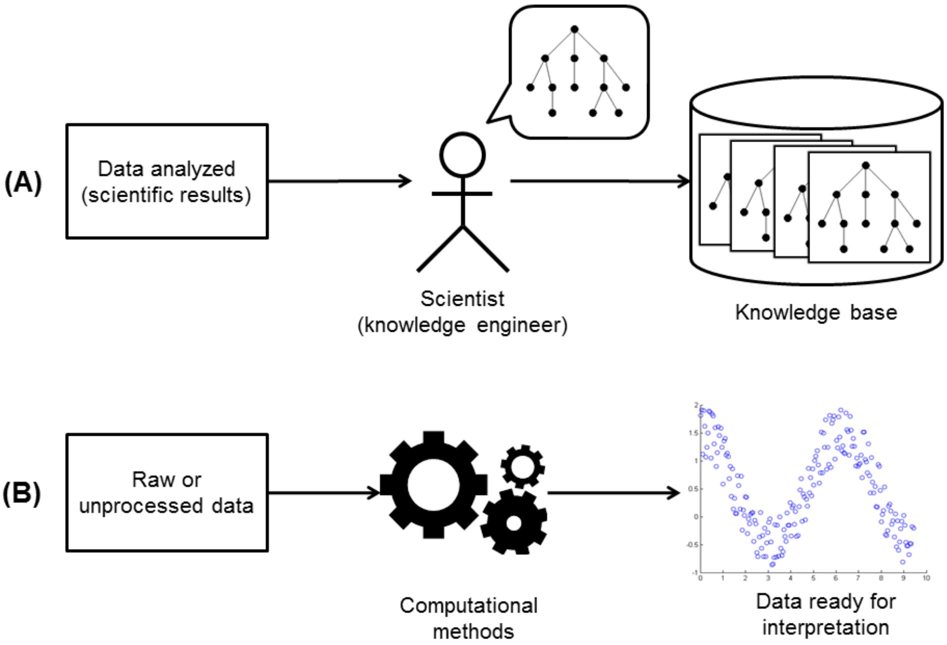

(A)
Data analyzed (scientific results)
Scientist (knowledge engineer)
Knowledge base
(B)
Raw or unprocessed data
Computational methods
2
1.5
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
8
9
10
Data ready for interpretation

Figure 6 – Comparison between (a) scientific knowledge engineering and (b) data intensive approaches

Furthermore, also contrasting to SKE, the researchers' intervention in data intensive approaches is more restricted. This is because, as previously said, the data is most often collected from automated sources such as sensors. And even when produced by researchers, it normally results from a direct product of research activities — what Hunter (2008) named 'born-digital research output' such as data streams, images, complex arrays, and maps, amongst others. Computational techniques and algorithms are applied to this data to subject it to the examination of the researchers, who in turn can support the answering to the research questions investigated. It is also worth pointing out that apart from providing huge amounts of data to researchers' examination, there are some applications of data intensive approaches which aim at answering research questions directly and solely from the data, e.g., Callahan *et al.* (2011).

Based on these characteristics, it is possible to observe that the two approaches are not exclusive alternatives, but rather complementary. One of the most immediate examples of this complementary nature can be seen when the output of data intensive approaches is used to support scientific inquiries and the results of these investigations are then modelled into a knowledge base as defined in SKE. In other words, the basic idea is that the output of one approach can be used as an input by the other. It is also possible to conceive the combination in the reverse direction. Since knowledge bases can accumulate huge amounts of data, it is possible that techniques and algorithms primarily used in data intensive approaches can be applied to explore these collections of data.

Another example of how these approaches can be combined occurs in some application of text mining to scientific articles (Rzhetsky *et al.*, 2004; Kiritchenko *et al.*, 2010). In these situations, there is a focus on first extracting a set of relevant information items from the text to a knowledge representation and then use it to perform some inference. Illustrating this, in Kiritchenko et al. (2010) 21 experiment's characteristics are retrieved and organized by a four-level taxonomy. And in a similar way, Rzhetsky et al. (2004) uses an ontology to determine what information has to be retrieved from the text. The same ontology is later used on a knowledge base serving as basis to perform inferences.

These examples show an interesting complementary nature between the perspectives in the computational manipulation of scientific knowledge and also indicates the possibility of taking techniques and algorithms from one to another. Specifically, the example involving text mining techniques may raise the question of whether SKE is really necessary, as even more elaborated scientific results and conclusions can be automatically retrieved from research papers. However, although the automatic retrieval of scientific results sounds appealing, it is important to emphasize that there are limitations to this approach. To some researchers, such as van Valkenhoef *et al.* (2013), the precision achieved in mining scientific texts is still insufficient to use this in systems supporting strategic decisions. Besides, Dinakarpandian *et al.* (2006) adds that even when a high precision level is achieved, one usually reaches that level in a limited context and often without considering the retrieval recall.

Generally speaking, this issue was named by some researchers (Mons, 2005; Cohen and Hersh, 2005) as the 'buried knowledge' problem. The point made by them is that it seems unreasonable to 'bury' knowledge in research papers and then try to use text mining techniques to (partially) extract it back. Thus, the point is why this is necessary when a proper formalization of the scientific results can be made in the first place? This excerpt from Bairoch (2009) summarizes the situation: '*it is quite depressive to think that we are spending millions in grants for people to perform experiments, produce new knowledge, hide this knowledge in an often badly written text and then spend some more millions trying to second guess what the authors really did and found*'.

4.2.2 Computational Discovery of Scientific Knowledge

The computational discovery of scientific knowledge was one of the first initiatives intending to combine computational methods and Science. The area emerged from the perception that Science is a problem-solving activity and that problem solving can be cast as search through a space of possible solutions (Džeroski *et al.*, 2007). In heuristic problem-solving, a person uses mental operations to transform knowledge from one state to another grounded on basic rules, to select the appropriate operators, choose from the candidate states, and decide when an acceptable solution is found. According to Simon (1977), scientific discovery could be described in a similar way. Scientific theories would be seen as knowledge states, and in response to new observations scientists would use mental operations to transform these states in new scientific theories or refinements and adaptations. Based on this proposition, the first systems implementing the computational discovery of scientific knowledge began to be constructed in the 70's (Džeroski *et al.*, 2007).

Contrasting to SKE where there is a focus on the translation of scientific results into knowledge representations, the computational discovery of scientific knowledge focuses on devising automated reasoning mechanisms that can emulate the whole creative mental processes involved in the generation of such scientific results (Figure 7). These processes include hypothesis formulation, designing experiments, and the analysis of study results (deductions, inductions, and abductions). Another difference that is somewhat a consequence of the previous one, is that it seems that the computational discovery of scientific knowledge is more restricted to domains where mature scientific methodologies are established and have a tradition in using (mathematical) models as it emulates the researchers' discovery process. On the other hand, by keeping the human factor in scientific creative reasoning, SKE is less influenced to do that. Nevertheless, as we will discuss in Section 4.3, scientific methodologies also affect how scientific results are described and organized. Thus, they have some influence on how knowledge representations can be engineered, in a manner similar to that where the computational discovery of scientific knowledge takes advantage of the systematization of mature disciplines.

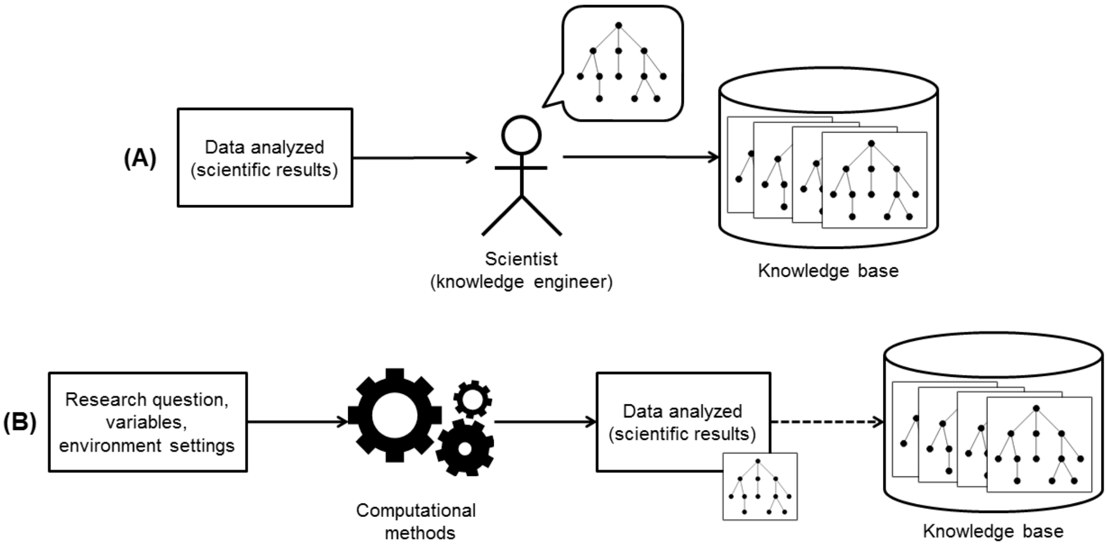

(A)
Data analyzed (scientific results)
Scientist (knowledge engineer)
Knowledge base
(B)
Research question, variables, environment settings
Computational methods
Data analyzed (scientific results)
Knowledge base

Figure 7 – Comparison between (a) scientific knowledge engineering and (b) computational discovery of scientific knowledge

Just as an example of this kind of automated discovery, one of the first developed scientific discovery systems tried to induce quantitative laws from a set of experimental variables (Langley *et al.*, 1983). The BACON program is provided with a set of independent and dependent variables which it uses to carry out simple experiments drawing from simulated data, and which it uses to organize results into a taxonomic hierarchy. Once BACON gathers data for a given node in its hierarchy, it searches for constant values of dependent terms (augmenting the node's description with that constancy) or relations between independent and dependent terms (defining new terms as products or ratios of existing terms) and continues the search. Then, the system propagates constant values to higher levels in its hierarchy, where it treats them as dependent values in its search for higher-level numeric laws. BACON uses a diverse set of heuristics which can be divided into data-driven approaches (e.g., numerical relations between independent and dependent terms) used to direct the discovery process from regularities in data and theoretical-driven strategies (e.g., physical symmetry and conservation properties) used to summarize earlier findings in simple ways. In Langley (1987) many of the discoveries made with BACON are discussed in detail, such as Ohm's Law of electrical circuits, Archimedes' Law of displacement and the Law of Gravitation.

Implicitly present in this example is the division of the scientific behaviour into two parts (Džeroski et al., 2007): scientific *knowledge structures* and scientific *activities*. Scientific knowledge structures are the means used by researchers to build and maintain scientific knowledge, which according to Džeroski et al. (2007) can be classified into three main types: taxonomies, laws and theories. Scientific activities, on the other hand, are those that are conducted to build and apply scientific knowledge. They include inductive activities such as the formation and revision of taxonomies, laws and theories. In addition, they also include deductive activities such as the formulation of predictions and explanations. Predictions take a law and contextual factors to infer what can be observed as a result of an specific intervention. Typically, researchers derive a model from the law, considering the environments' specificities, and deduct a prediction from the model. Explanations usually connect a theory to a law (or a law to a prediction) describing how and why, and what specific phenomenon was observed. Finally, scientific activities also involve abductions which are associated to reasoning explanation, but involve some kind of supposition or analogy.

Some common aspects of SKE and computational discovery of scientific knowledge becomes more apparent when we split scientific knowledge structures and activities. For instance, to emulate scientific activities computational discovery of scientific knowledge needs formal knowledge structures to generate new scientific findings using these formalizations. The same kind of formal representation is also needed by SKE, even though with the different purpose of representing the final results and conclusions of scientific inquiries. Figure 7B highlights this aspect showing how scientific results in computational discoveries are already formalized with knowledge representations and thus could eventually be taken directly into knowledge bases (the dashed line denotes this). Therefore, this shows the possibility that lessons learned by using knowledge representations in computational discovery can be brought to SKE. It is not the scope of this paper to discuss this possibility in depth, but nevertheless many examples of models used to represent scientific knowledge structures in computational discovery can be obtained in Shrager (1990), Valdés-Pérez (1996) and Džeroski et al. (2007).

4.3 The State of the Art on Scientific Knowledge Engineering: a Literature Review

4.3.1 Method

The initial idea for the literature survey was to conduct a systematic mapping study (Biolchini *et al.*, 2005). Mapping reviews are particularly useful to bring an overview of a research area, identifying the existing works, main topics, and quantities. Amongst the most important characteristics of mapping studies cited by Kitchenham and Charters (2007) and Budgen *et al.* (2008) are: (i) the mapping of studies conducted in an area, (ii) the identification of research gaps and clusters in a set of studies aimed at identifying topics that can be the target of systematic reviews, (iii) the possibility of answering multiple research questions with a broader nature, (iv) the focus of an also broader data extraction usually done by summarizing procedures and some kind of classification, and (v) results that seek to direct future works in the area.

However, in the absence of a well-defined terminology for the area the path chosen was to conduct a non-systematic mapping study. Because of that decision, some steps of systematic mapping studies were not followed or not properly registered. For instance, the search string was not defined *a priori* based on the research questions. Just as a sample of how difficult it was to find the relevant search terms, only one paper mentioned the term 'knowledge engineering' and some articles did not even contains the term 'knowledge'.

To overcome this problem and be able to search the articles, we have used the snowballing strategy (also known as cross-checking citations). The basic idea of this technique is to identify a set of initial articles using relevant terms to the research theme and then identifying further papers looking at those referenced by (backwards direction) and those referring to (forward direction) this initial set. The terms used to collect the initial set of papers were a combination of 'knowledge engineering', 'knowledge representation' and 'ontology' and 'evidence' and 'science'[9]. The snowballing was exhaustively executed, that is, backwards and forwards until there was no paper that suited the review goal. The Scopus digital library was used for search — using article, abstract and keyword fields. Supplementing Scopus, the Google Scholar search engine was also used, but only for snowballing.

[9] To have an impression of the precision at hand with a more structured search string, we compiled one *a posteriori*, i.e., after we identified the papers in this section. The string captures two dimensions, namely 'knowledge engineering' and 'scientific knowledge'. Retrieving all the papers presented in this section, the search string ended up with the following terms: ('scientific knowledge' OR 'science knowledge' OR 'science information' OR 'scientific information' OR 'pathway' OR 'scientific paper' OR 'scientific publication' OR 'scientific assertion' OR 'scientific discourse' OR 'supplementary nature scholarly discourse' OR 'scientific statements' OR 'scholarly communication' OR 'mechanism knowledge' OR 'evidence based' OR 'scholarly argumentation' OR 'scientific argumentation' OR 'scientific contributions' OR 'scientific claim' OR 'evidence representation' OR 'research proposal' OR 'scientific theory') AND ('knowledge engineering' OR 'knowledge management' OR 'expert system' OR 'knowledge-based system' OR 'computational infrastructure' OR 'ontology' OR 'inference' OR 'knowledge representation' OR 'semantic web' OR 'rdf' OR 'argument system' OR 'discourse representation' OR 'research system' OR 'belief functions' OR 'decision support system'). With these terms, more than 13,000 papers were returned which clearly shows an absence of consensus in terminology, but also represents the diversity of applications of SKE.

Despite the fact that the literature review was not systematic, some aspects of mapping studies were pursued. The first one was the definition of an explicit set of inclusion and exclusion criteria for the papers. The inclusion criteria were papers that focused on the computational representation of scientific results and that put researchers with an active role on the modelling/'translating' of these results into knowledge representations, using a computational infrastructure. Both papers with new proposals of computational infrastructures or knowledge representations and papers that only discussed the theme were included. We should add that some variation in the scope of the papers was accepted. For instance, some papers focused on specific aspects of scientific results (e.g., how to represent the hypotheses associated with findings) whereas others were concerned with the scientific results as a whole (i.e., as it usually published in technical literature). The exclusion criterion was papers that addressed other approaches of computational manipulation of scientific knowledge such as the ones detailed on the previous section. Papers which exclusively introduce ways of managing and organizing scientific data produced *along* research activities to support its analysis or provenance were also excluded. No restrictions were made considering the quality of the papers.

The second aspect was the definition of a classification scheme for the papers found. Following systematic mapping orientations, the scheme was refined in an incremental way as new papers were included. The classification sought to identify the essential properties that could help understand and characterize SKE. A more generic classification, independent of the research area, was also used to provide an idea of what domains have new SKE proposals or discussions and their distribution over time. This more generic form of classification is commonly used in systematic mapping (Petersen *et al.*, 2008).

Often, the classification scheme is presented as broad research questions. In the case of the objectives of this paper, the research questions were defined as:

RQ1 In which areas, period and publication types has SKE been presented?

RQ2 Are there any specific particularities of the (scientific) knowledge which are the object of SKE? Which ones?

RQ3 What are the main SKE aspects in terms of techniques, technologies and activities employed in the construction and utilization of computational infrastructures?

These questions are discussed in the following sections.

4.3.2 Publication areas, period and types (RQ1)

Due to the short time frame, with the first papers published only in 2006, and to a limited number of papers found (22 articles), it is not possible to say that the interest on SKE is increasing over the years (Figure 8), but it can at least suggest that SKE is emerging as a research area. From the 22 articles, three (Groth *et al.*, 2010, Clare *et al.*, 2011, Kuhn *et al.*, 2013) related to the same proposal (nano-publications), but contrary to other cases where only the most recent or important paper was selected, we chose to keep all of them in the review as they introduce alternative extensions and visions to the initial nano-publications proposal and also are from different authors. The literature review included papers until May 2014 so it is possible to see in Figure 8 that last year was compromised in terms of numbers of papers. Most papers are published in journals, although there are many in conferences and workshops as well, which suggests that SKE works are achieving a high level of quality/importance and also that there are several preliminary proposals being discussed.

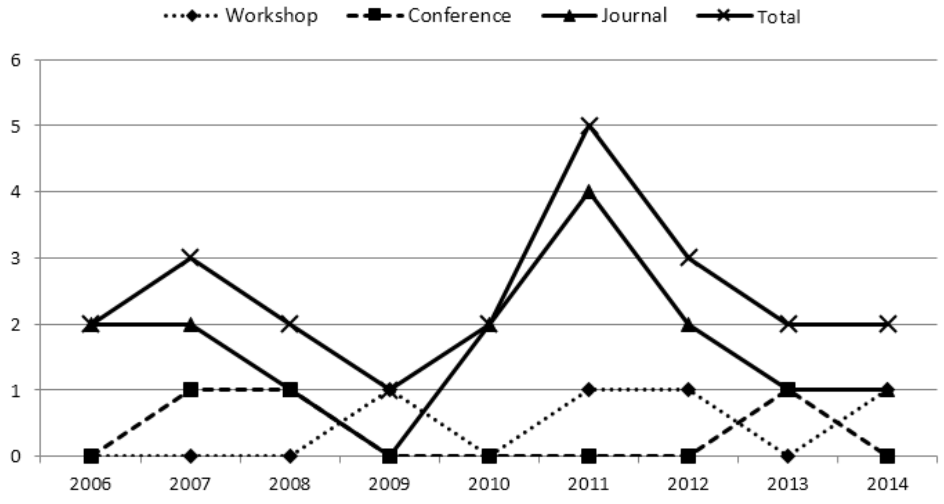


Figure 8 – Paper publication types and distribution over time

Considering the paper distribution in the scientific domains (Figure 9), a significant part of the proposals (41%) have, according to their authors, general applicability. Another significant part (also 41%) is distributed among some biological and health sciences (Biology, Biomedicine, and Medicine). It should be noted here that even among the general applicability proposals, almost all of them have examples from the Health Sciences. The exception is the work of Pike and Gahegan (2007) which is proposed as independent from domain but shows applications to Geology. The remaining 18% includes three more domains: Software Engineering, Sustainability Sciences, and Geology.

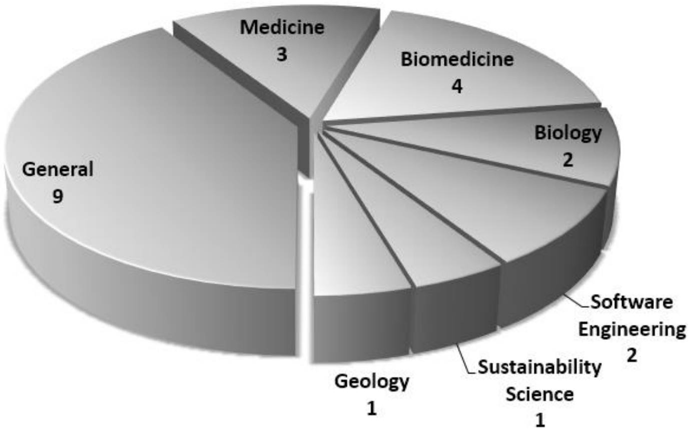


Figure 9 – Paper distribution over domains

4.3.3 Scientific Knowledge Properties Addressed (RQ2)

It seems that the fact that most of the papers found have their origins in, or at least have examples from Biology, Biomedicine and Medicine, is associated with the long tradition of these areas of using systematic research methodologies. These domains have high formalization and standardization levels in their scientific practices, with a strong focus on controlled quantitative studies and meta-analysis.

In Medicine, the work of Russ *et al.* (2011) is a clear example of how it is possible to take advantage of these properties. In their knowledge representation, the authors decouple the experimental design model from the domain specific reasoning model to allow, respectively, both observational assertions (based on specific data from carefully-planned experiments) and interpretation-based assertions (based on a higher-level understanding of the phenomena under study) to be modelled. Other works found in the review that also benefit from the scientific practices adopted in Medicine are Hunter and Williams (2012) and van Valkenhoef *et al.* (2013). In these two papers many aspects of the evidence-based practice are incorporated into the computational infrastructures which as a result support researchers in the aggregation of evidence and decision-making, based on such aggregations. Also related to the scientific knowledge properties, both works explicitly discern which evidence types define the scope of knowledge supported by the infrastructures. In Hunter and Williams (2012), for instance, the scope is limited to evidence resulting from randomized controlled trials, cohort studies and meta-analysis, comparing two experimental treatments.

The papers that stand closer to Biology, including Biomedicine, draw from systematic scientific practices as well, but also exploit the notion of mechanisms, which are widely used in the area (Craver and Darden, 2005). In general terms, using the definition from epistemology, mechanisms consist of a coordinated sequence of causal interactions between the parts of a system organized in such a way that the mechanism operation is what produces, or is the source of, the phenomenon to which the mechanism is indicated as an explanation (Bunge, 2004). In Boyce *et al.* (2007) the mechanisms related to the way of how drug-drug interactions occur are represented by rules which, in conjunction with a set of evidence (e.g., *in vitro* studies) on the drugs, allow the determination of an interaction and the belief level associated to the real existence of it. And in Croft *et al.* (2011), biologists model human pathways and reactions using primary studies, vocabulary bases and other resources.

Just as in Medicine, the evidence-based practice paradigm is also being leveraged in Software Engineering as a reference to design knowledge representations. This is shown in the works of Santos and Travassos (2013) and Ekaputra *et al.* (2014). The main difference between the two proposals lies in the representations' level of detail of the scientific elements. In Ekaputra *et al.* (2014), the knowledge model is tied to controlled trials concepts such as measurements, factors, treatment and metrics, in a similar way to Hunter and Williams (2012). In addition, the model also describes Software Engineering sub-domain concepts and is designed in such a way that it must be constantly extended and revised in order to accommodate new scientific results. In the case of Santos and Travassos (2013), the proposal is not limited to specific evidence types as it models higher level elements of scientific theories such as concepts, relationships and confidence level. As a result, it only requires that the primary studies results describe causal relationships between the investigated software technology (method, technique and tool) and the context software systems and developers.

Also focusing on the notion of scientific theories is the work in the domain of Geology (Sharma *et al.*, 2010). Similarly to Santos and Travassos (2013) the authors focus on representing the context information of each result — such as rocks and temperature in Geology, and software developer experience and programming language in Software Engineering. By incorporating this contextual description into theories it can distinguish which model (i.e., theory) best explains an observed situation. An important difference between Sharma *et al.* (2010) and Santos and Travassos (2013), is that in the Geology proposal the context is part of the uncertainty which is being uncovered by theories so that, for instance, it is necessary to give a probability to the fact that a certain deposit A has a host rock B. In the Software Engineering proposal, on the other hand, the context is expected to be unequivocally described and, thus, it is not necessary to assign probability to the fact that a software team has a software tester, but only to the confidence that a software tester can improve the software quality attribute (i.e., the causal relationship). Lastly, additional work using scientific theory as the starting point for the knowledge model is Brodaric *et al.* (2008). In this case, given its intended general applicability, it concentrates in representing higher level elements linking theories to its parts (e.g., equations and predictions), source publications, facts, data, and other theories.

There is another kind of proposal in which knowledge models try to capture the practice of researchers in reporting their findings. In Kraines and Guo (2011), the authors describe an ontology covering three major conceptualizations of sustainability science: (i) situations, which is basically an activity-event model where activities (e.g., transportation) can have physical objects associated (e.g., fuel cells) and starting and ending events (e.g., a point in time); (ii) scenarios, which define types to describe scenarios for achieving sustainability, such as problem type, goal and alternative scenarios; and (iii) analysis, which describes the analysis methods and tools used to study the situations and phenomena described. From the biomedical domain, Bölling *et al.* (2014) define a structured representation for scientific evidence. The model concentrates on the aspects routinely considered by a researcher when analyzing evidence of a given scientific finding. That is, the representation of the (i) experimental methods and settings used to obtain the results, (ii) reasoning and assumptions used to infer the result at hand, and (iii) information sources and authors through which the finding was reported and propagated. The generic model from Pike and Gahegan (2007) is designed, according to the authors, as a bottom-up representation. Contrasting to top-down representations in which the knowledge structure is predefined in terms of, for instance, ontologies, bottom-up representations try to draw from the researchers' collaboration. The bottom-up proposed representation has six types of knowledge resources: concepts, people, files, tools, places, and tasks. These types are used and connected to each other to describe what the authors define as a situation.

From the remaining works, there is no significant discernment regarding the scientific knowledge properties — which is what could be expected as they were indeed conceived for general application. It is possible to identify two main strategies used in the proposals to achieve a higher level of generality. One is marked by the group of works (Dinakarpandian *et al.* , 2006, Groth *et al.*, 2010, Marcondes, 2011, Clare *et al.*, 2011, de Waard and Schneider, 2012, Kuhn *et al.*, 2013) which models scientific results as assertions using triples of the type <antecedent><relationship><consequent> (also described as concept-relationship-concept and subject-predicate-object). On the other hand, the other group (Mancini and Buckingham Shum, 2006, Groza *et al.*, 2007, Ciccarese *et al.*, 2008, de Waard *et al.*, 2009) is more focused on modelling the argumentation structures used in scientific papers, from the organization of their sections to the arrangement of arguments for and against a specific theme.

4.3.4 Relevant factors identified in Scientific Knowledge Engineering initiatives (RQ3)

Differentiation between container and content

The idea of abstraction levels in knowledge-based systems is intuitively well understood in terms of how detailed knowledge is represented. One interesting way to characterize different level of abstraction used in knowledge representations in scientific domain is the distinction made by Bechhofer *et al.* (2013) which distinguishes between the container and content levels. On the container level, the goal is to identify the existing semantics between text blocks and the discourse embedded into them. Text blocks vary from small extracts to whole sections. On a content level, in contrast, knowledge is more granular and tends to knowledge items in the shape of assertions, positions, and arguments. The meaning of content in this context is used to denote scientific knowledge itself, upon which representation and reasoning formalisms are applied to derive new or uncover existing scientific results.

Closer to the container end we can find the proposals classified as scientific discourse models (Mancini and Buckingham Shum, 2006, Groza *et al.*, 2007, Ciccarese *et al.*, 2008). For instance, Ciccarese *et al.* (2008) define four types of discourse elements: (i) Discourse Element, which is a more granular narrative object, representing a mapping of digital resources to statements in natural language (e.g. sentences, paragraphs); (ii) Research Statement, representing a particular discourse element having a claim or hypothesis nature; (iii) Research Question, associated with the topic under investigation; and (iv) Structured Comment, which acts as a structure representation for a comment in a digital resource. In addition, as opposed to other representations, Mancini and Buckingham Shum (2006) do not model the coarse-grained rhetorical or linear structure of the publications, but rather concentrate strictly on organizing the coherence among text segments using relations such as causal, general, problem related, similarity, taxonomic, and support/challenges.

On the other hand, most discussions focus on content knowledge structures (Dinakarpandian *et al.* , 2006, Boyce *et al.*, 2007, Pike and Gahegan, 2007, Brodaric *et al.*, 2008, Groth *et al.*, 2010, Sharma *et al.*, 2010, Clare *et al.*, 2011, Croft *et al.*, 2011, Kraines and Guo, 2011, Marcondes, 2011, Russ *et al.*, 2011, Hunter and Williams, 2012, van Valkenhoef *et al.*, 2013, Santos and Travassos, 2013, Kuhn *et al.*, 2013, Bölling *et al.*, 2014, Ekaputra *et al.*, 2014). The difference between container and content made in this subsection will be implicitly recurrent in next subsections where their distinctions and similarities regarding technologies, reasoning capabilities and modelling process will be more apparent. It is also worth noticing that this differentiation does not represent a dichotomy. Some proposals, arguably most notably Bölling *et al.* (2014), combine elements from both abstraction levels modelling scientific concepts and assertions, and also tracing them to their argumentative structure within and across publications.

Technologies, techniques and formats used for knowledge representation

The most common format used in SKE papers is what could be generalized as triple-based. In fact, the triplet structure has been proven to accommodate several types of knowledge in different domains and it is one of the building blocks of most ontology formats, including the open Semantic Web standards for RDF and OWL, being also found in other representations such as Conceptual Graphs. The wide use of this format is aligned with Hunter and Liu (2010) who state that 'any domain in Science has ontological knowledge that could be usefully encoded and used in the form of a description logic[10]'. Among the triple-based works, RDF is the most common technology used for knowledge representation (Groza *et al*., 2007, Brodaric *et al*., 2008, Ciccarese *et al*., 2008, de Waard *et al.*, 2009, Groth *et al.*, 2010, Sharma *et al.*, 2010, Kraines and Guo, 2011, Marcondes, 2011, Clare *et al.*, 2011, de Waard and Schneider, 2012, Kuhn *et al.*, 2013, Bölling *et al.*, 2014). The vocabulary developed in these works is quite diversified, including the definition of assertive relationships (e.g., cause-effect in Groth *et al.* (2010)) and a taxonomy for the characterization of assertions (e.g., hypothetical or dubitative (de Waard and Schneider, 2012)). The antecedent and consequent parts of triples also vary considerably from simple terms and concepts (e.g., Groth *et al.* (2010), Marcondes (2011)) to small sentences and expressions (e.g., called AIDA — Atomic, Independent, Declarative and Absolute sentences in Kuhn *et al.* (2013)). Papers from the discourse model background which use ontologies (Groza et al., 2007, Ciccarese et al., 2008) try to represent the main discourse elements found in scientific articles such as relationships between

[10] Description logic is a subset of the language of classical logic. Ontologies are commonly used in conjunction with description logics, which allow logical reasoning based on monadic and binary predicates to represent relations such as sub-concepts, union and intersections.

paragraphs and sections, and the common organization found in scientific result reporting (e.g., research questions, motivations, and study procedures).

There are also works (Dinakarpandian *et al.*, 2006, Mancini and Buckingham Shum, 2006, Pike and Gahegan, 2007, Santos and Travassos, 2013) which, though not using ontology technologies, use the triple-based format as well. Part of these works uses specific technologies for knowledge representation. In the case of Dinakarpandian *et al.* (2006) a Backus-Naur grammar is used to represent complex scientific assertions. And in Santos and Travassos (2013) the Unified Modelling Language (UML) is used to model theories concerned with evidence generated in primary studies. The other papers, based on theoretical foundations, use their own set of tools built to formalize the ontological knowledge — for instance, Mancini and Buckingham Shum (2006) use a view centered on the definition of relationships based on the work of Sanders *et al.* (1993) Cognitive Coherence Relations.

The remaining works proposed representations more specialized to the type or characteristics of scientific knowledge to which they were devised. As a result, given the different requirements, a more diverse set of technologies was used. As regards the aggregation of quantitative data originated with controlled experiments, we can cite Hunter and Williams (2012) and van Valkenhoef *et al.* (2013). To that end, Hunter and Williams (2012) structure extracted study information (e.g., outcome indicator, a binary relationship between the treatments (<,>,=), and the statistical significance) and used that information to formalize these facts in a collection of arguments in favour and against experimental treatments using a directed binary graph. Then, the aggregation is achieved by using the Dung (1995) theory of argumentation. The basic idea is to identify the arguments (i.e., evidence) which are free of conflicts or that can refute all counter-arguments. Van Valkenhoef *et al.* (2013) designed a conceptual model (object-oriented) to represent the main results produced by primary studies in Medicine (e.g., measurements and treatment). And, with the data structured within the conceptual model, the aggregation is accomplished by using conventional statistical methods. The object-oriented paradigm was also used to model results from controlled experiments in Ekaputra *et al.* (2014), but, besides the different domain, in the case Software Engineering, Ekaputra *et al.* (2014), is in a more preliminary stage and currently focusing more on search capabilities.

As mentioned in the previous section, Russ *et al.* (2011) split the scientific knowledge into observational and interpretation-based assertions. The authors developed a graphical representation model called KEfED (Knowledge Engineering from Experimental Design) which allows the design of an experimental protocol using a workflow with graphical elements such as activities, parameters (independent variables), measurements (dependent variables), among others. The terms and concepts found in the protocol can be defined in external ontologies. These protocols are then instantiated from the data of primary studies which used the protocol design. With the stored observation data, interpretations can be inferred from a set of studies. The interpretations are modelled using first order logic on the data. In the paper example, logical rules (e.g., *part of* and *overlap*) are defined to map cerebral regions.

Also using first order logic, Boyce *et al.* (2007) model drug-drug interactions using a set of `IF-THEN` rules. The work presents four rules representing the interactions — for instance, one of the rules could be described as '*a precipitant inhibits the metabolic clearance of an object drug*'. Based on the rules, a justification-based maintenance system is used to evaluate how the rule consequent elements are predicted, given certain justifications (i.e., antecedents — the IF portion — and other clauses). Evidence from primary studies are accumulated into the knowledge base as properties used in the justifications. And given the user-defined criteria for belief in an evidence based on the study type (e.g., randomized controlled trial or cohort-study) the system can enable or retract the justifications which in turn affect the consequent elements (i.e., predictions).

The last work in the literature review (Croft *et al.*, 2011) aims at capturing knowledge on pathways. In a simplified manner, pathways are descriptions of the molecular interaction network in cells. The proposed system uses a frame-based knowledge representation (Forbus and DeKleer, 1993) to model these interactions. The frame model consists of classes (i.e., frames) which describe different concepts (e.g., reactions, physical entities and their relationships and events). Instances of theses classes are created to capture knowledge from complex experimental data. All of the classes and their properties can be manipulated using a graphical representation, named pathway browser, which uses the System Biology Graphical Notation (Novère *et al.*, 2009). It should be pointed here that there are many pathway databases available (Bauer-Mehren *et al.*, 2009; Khatri *et al.*, 2012). Still, Reactome was chosen because of its explicit concern for extracting pathways and reactions from biological experiments and literature — which is done by a small group of researchers and curators who spend months studying a pathway, such that they would be familiar with almost every publication on that pathway (Bauer-Mehren *et al.*, 2009). In addition, Reactome is regarded as one of the most complete and best curated pathway databases (Bauer-Mehren *et al.*, 2009) and is positioned among the last generation pathway tools from the three generations identified by Khatri *et al.* (2012). Thus, from a feature-wise perspective, Reactome can be considered representative for the goals of this review.

Considering all the papers, it is possible to observe that the range of technologies and procedures is still somewhat restricted if we think of the considerable diversity of methods and techniques available in the Artificial Intelligence technical literature. Interestingly, at the same time, it seems that the ‘simplicity’ of some representations, especially those based on RDF, is what contributes to the extent of generality in terms of scientific domains and disciplines it can accommodate. Nevertheless, this simplicity comes with a trade-off of inference power. For instance, the more domain-specific drug-drug interaction proposal is capable of more precise inferences in terms of answering the possible existence of an unanticipated drug-drug interaction. Thus, SKE will still have to find the appropriate balance of its representations to achieve an adequate representation scope while keeping inferences useful.

The knowledge base and the possible answers

The types of searches and results that can be submitted to or obtained from semantic-based models are one of the main benefits expected from SKE and are mostly associated with inference possibilities. A significant part of the papers (41%) have not presented any discussion or specific explanation about the search facilities. Some papers (de Waard *et al.*, 2009, Hunter and Williams, 2012, Santos and Travassos, 2013) mentioned that this aspect was not being addressed at that time or, as in the case of Groza *et al.* (2007), that this aspect was still under development as the proposal was in a preliminary stage. Other papers (Ciccarese *et al.*, 2008, Groth *et al.*, 2010, Marcondes, 2011, Kuhn *et al.*, 2013) have not discussed the search aspect or simply mentioned in a general way that the representation could be used to support users (i.e., researchers) in searching the knowledge base.

Amongst the papers that gave specific attention to the search aspect there are some that even reserved whole sections to discuss the theme. In this group, it is possible to identify two levels of sophistication: based on the data model and based on the semantic inference. The searches based on the data model only explore the syntactic structure into which the data is represented usually by using ‘fill in the blanks’ filters. This type of filter is often available in the ‘advanced search’ of the ‘conventional’ information systems and consists of `<term, field>` combined by logical operators (`NOT, OR, AND`) where `field` is associated to some data model attribute. Even if not representing the ideal case for SKE, the computational infrastructures based on this type of search (Clare *et al.*, 2011, Croft *et al.*, 2011, van Valkenhoef *et al.*, 2013, Ekaputra *et al.*, 2014) tend to have more precision than those based on full text search such as the ones used in digital libraries.

On the other hand, the searches based on semantic inference, as the name implies, can extract logical consequences (i.e., answers to the queries) from a set of facts (i.e., the information persisted in the knowledge base) and a set of specified rules. Thus, the main benefit is the possibility of discovering new 'hidden' facts in the data and support scientists to make sense of scientific results. In two works (Boyce et al., 2007, Russ et al., 2011) query results are a direct consequence of scientific evidence deposited in the knowledge base (e.g, questions in the form: 'based on the following evidence could a given drug interaction happen?'). All other works (Dinakarpandian *et al.*, 2006, Mancini and Buckingham Shum, 2006, Pike and Gahegan, 2007, Sharma *et al.*, 2010, Kraines and Guo, 2011, de Waard and Schneider, 2012, Bölling *et al.*, 2014) describe in detail how the description logics of their triple-based representations can be used to answer queries. For instance, Dinakarpandian *et al.* (2006) thoroughly describe how the relationship types such as generalization/specialization can be used to improve query results by returning assertions that uses terms more generic/specific than those used in the search for a clinical issue or a specific Biology question.

Another interesting application of triple-based representations is found in Sharma *et al.* (2010), which combines a taxonomic hierarchy of rock types and properties with probability over the taxonomy relationships. Ontologies are used to formalize the taxonomic hierarchies describing instances in the geological domain such as particular locations and models of published known geological configurations using probabilities. The search is, then, a probabilistic matching of instance-to-models or model-to-instances — using Bayesian Networks given the directed acyclic graphs (i.e., the taxonomic hierarchies). Finding the most likely models for the instance can be used to determine what is the mineral or landslide most likely to be found at the particular site described by the instance. Similarly, it can be useful to compare one model to multiple instances, to find the location that is most likely to contain given minerals.

As it could be seen in this section, although the possible queries and answers that computational infrastructures support are a central aspect of SKE, many papers did not give them the proper attention. It seems that in most cases the authors suppose that knowledge representation already defines what the potential inferences are and how the knowledge base can be queried. We believe, however, that designing adequate facilities for searching and describing how scientists can benefit from it is essential for the establishment of SKE.

Aggregation of scientific results

This represents a major motivation for SKE, as the aggregation or synthesis of scientific results is an important instrument to cope with the huge amount of new findings that come to the fore. The notion of synthesis can be defined as any method or procedure aimed at integrating and interpreting investigation results for the purpose of creating generalizations or answering specific research questions (Cooper *et al.*, 2009). One way to understand the outcomes from (research) synthesis is via the distinction between integrative and interpretative methods established by Noblit and Hare (1988).

Generally speaking, an integrative synthesis involves the summarizing and pooling of data, incorporating the results into each other. As a result, integrative syntheses can reveal what a set of scientific results 'says' as a whole — usually a qualitative description or a quantitative indicator on the size and direction of a correlation or cause-effect relationship. An interpretative synthesis, on the other hand, has as its main goal the describing or developing of concepts in such way that it can be organized in a theoretical model, taxonomy, narrative arguments or any other form that allows a connection between the synthesized results to be figured out. Hence, interpretative syntheses achieve the synthesis through the grouping or classification of scientific results into a higher level (conceptual) model (Dixon-Woods *et al.*, 2005).

Table 3 presents the papers classified into integrative or interpretative according to their main synthesis characteristics. Among the integrative papers, the aggregation approaches include conventional statistical methods and first-order logic. It is interesting to see that the use of uncertainty formalisms such as fuzzy set, rough sets and possibility theory is still missing in the proposals. There are researchers such as Kuhn *et al.* (2013) who state that the uncertainty concern should be separate from the representation itself suggesting the integration with the work of de Waard and Schneider (2012). Yet there are examples where the opposite strategy is used (e.g., Santos and Travassos (2013) which uses Dempster-Shafer Theory (Shafer, 1976)) within the representation. The interpretative-focused proposals sought to offer a means to formalize conceptual semantic relationships mainly with the use of triple-based representations. In addition, several works included some kind of description logics to allow scientists to explore patterns in the formed conceptual structures. Papers classified with both integrative and interpretive characteristics show concerns with how the conceptual models are defined while, at the same time, define how the uncertainty in the model relations is pooled. The papers listed in Table 3 are those that explicitly discussed the aggregation possibilities.

Table 3 – Aggregation strategies supported by proposals

Integrative	Boyce *et al.* (2007), Russ *et al.* (2011), Hunter and Williams (2012), van Valkenhoef *et al.* (2013)
Integrative/ Interpretative	Sharma *et al.* (2010), Santos and Travassos (2013)
Interpretative	Dinakarpandian *et al.* (2006), Mancini and Buckingham Shum (2006), Pike and Gahegan (2007)

Graphical representations

The representation of scientific results in a visual format can be an effective resource in attempting to find the means to facilitate the understanding of scientific knowledge. There are debates in the Philosophy of Science specifically focused on the issues of what are the best formats to represent scientific theories and their impacts on understanding, particularly in graphical representations. In Vorms (2011), the concern with this issue can be identified in the sentence 'two representing devices can contain the very same information (concerning the same thing) though conveying it in different way to us'.

The importance of graphical representations was reflected in the fact that almost half (45%) of knowledge representation proposals have some discussion about visualization. It is interesting to add that this percentage could be significantly increased (to 91%) if the proposals based on RDF or ontologies were included, as there are many tools to visually manipulate these representations. Still, to maintain analysis consistency, we have chosen to consider only the papers that developed or explicitly cited forms of graphical representations.

Among the considered proposals (Table 4), the most often used representation is the graph-based, including representations such as semantic networks, conceptual graphs and cognitive maps. The second category includes computational infrastructures offering facilities for workflow representation, which can indeed be considered as graphs as well, but are distinguished here due to their different focus, with the former geared to structural concerns (concepts and its relationships) and the latter to behaviour (sequence of activities or states). And the last type of representation used in the proposals is UML based, which is a particular kind of graph as well.

Table 4 – Graphical representation types used on papers

Graph-based	Dinakarpandian *et al.* (2006), Pike and Gahegan (2007), Sharma *et al.* (2010), Marcondes (2011), Hunter and Williams (2012), van Valkenhoef *et al.* (2013), Ekaputra *et al.* (2014)
Workflow representations	Croft *et al.* (2011), Russ *et al.* (2011)
UML-based	Santos and Travassos (2013)

Scientific knowledge modeling process

We begin this section with a quote from Markman (1999) who accurately describes the importance of the modelling process: '*It makes no sense to talk about representations in the absence of processes. The combination of the first three components (a represented world, a representing world and a set of representing rules) merely creates a potential for representation. Only when there is also a process that uses the representation does the system actually represent, and the capabilities of a system are defined only when there is both a representation and a process*'.

A modelling process is what can make explicit the link of how a researcher can start from the scientific results and end up with this knowledge modelled into a representation. This process consists basically of a set of procedures showing what knowledge items should be identified in scientific papers or studies, and how these items should be handled and organized for modelling. However, despite the importance of a modelling process, most of the papers showed more concern in detailing the knowledge representation than in describing how it could be used for modelling scientific knowledge. The general perception that could be extracted from the papers is that the narrower the scope and the better structured is scientific knowledge, the more direct becomes the definition of the procedures that should be followed, to translate knowledge into a representation. The underlying idea of this correlation can be explained by the fact that a more restricted scope and a more structured knowledge domain makes the mapping between scientific knowledge and the knowledge representation more clear in terms of how the notational syntax and semantics, rules and symbols should be used to represent the different scientific results elements. For instance, disciplines that follow the evidence-based practice have a significant systematization level as to how the area knowledge is produced, structured, evaluated, and used in practice. As a consequence, the organization of this more structured kind of knowledge into computational infrastructures can be facilitated.

We tried to map whether the definition of a modelling process is somehow related to the scope of scientific disciplines the knowledge representation can support and to their relative organization. In Figure 10, it is possible to observe that, although the structuring of scientific knowledge does not seem to be largely leveraged in SKE, most of the papers that focused on the modelling process were those which had a relatively reduced scope. It is worth pointing out the difficulty in this classification, as it is not a trivial task to find the limits of how well structured a knowledge domain is, relativize a scope as narrow or wide, or detect the boundaries of a well-defined process or not. Thus, despite the intrinsic subjectivity in the classification, it was guided by the following criteria: (i) scope: general or restricted to a scientific domain or sub-domain; (ii) structure of knowledge: non-structured or structured (domain knowledge with mechanisms and theoretical formalizations widely accepted by the academic community and/or the extensive application of well-defined research methodologies); (iii) process: defined (the paper mentions some procedure related to the use of the proposed knowledge representation) or *ad hoc* (knowledge representation utilization is associated only to the understanding of its syntactical and semantic description).

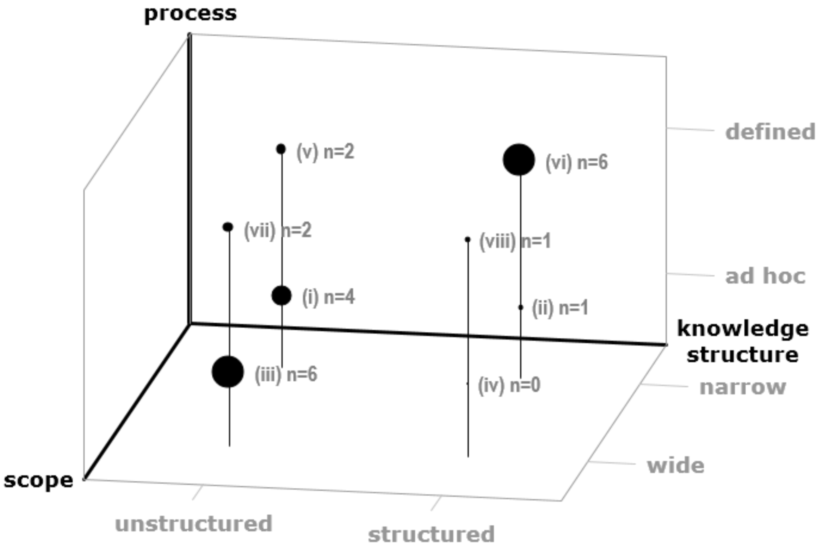


Figure 10 – Paper classification in three dimensions: the definition of modelling process for representing scientific knowledge and its mapping to knowledge structure and scope of scientific disciplines. N is the number of papers in each of the classification eight points[11].

There is an even number of proposals (50%) which have some kind of description associated with what steps are necessary for modelling scientific knowledge using the representation. Most representations (64%) have not been proposed based on how scientific knowledge is structured, be it associated with scientific methodologies

[11] References: (i) Dinakarpandian *et al.* (2006), Ciccarese *et al.* (2008), de Waard *et al.* (2009), Kraines and Guo (2011) – (ii) Sharma *et al.* (2010) – (iii) Groza *et al.* (2007), Pike and Gahegan (2007), Brodaric *et al.* (2008), Groth *et al.* (2010), Clare *et al.* (2011), Marcondes (2011), Bölling *et al.* (2014), Ekaputra *et al.* (2014) – (v) de Waard and Schneider (2012), Santos and Travassos (2013) – (vi) Boyce *et al.* (2007), Croft *et al.* (2011), Hunter and Williams (2012), van Valkenhoef *et al.* (2013) – (vii) Mancini and Buckingham-Shum (2006), Kuhn *et al.* (2013) – (viii) Russ *et al.* (2011).

concepts or elements related to theoretical formalizations (e.g., conceptual frameworks or mathematical models). Regarding scope, about 59% of the papers, as previously mentioned (Figure 9), have a narrow scope of application.

To better illustrate this classification the following examples are cited:

- Narrow scope and structured knowledge (ii and vi): the first example is the modelling of drug-drug interactions evidence in the work of Boyce *et al.* (2007). In this work, its restricted scope and well-known drug interaction mechanisms by the academic community of the area makes it virtually immediate to identify what knowledge should be modelled. The authors define four rules for mapping drug-drug interactions. Then, evidence is collected from papers indicating, for instance, drug enzyme inductions or inhibitions. In combining these aspects, it is possible to have the necessary elements to the use (i.e., the process) of the knowledge representation. The researchers that use this representation have a direct correspondence of what information should be collected and of what it represents for modelling purposes — in this case, the modelling of drug-drug interactions. The paper of van Valkenhoef *et al.* (2013) is also a good case of how the limited scope and structured domain knowledge can be the source of important design decisions of computational infrastructures. The work is based on Evidence-Based Medicine where the rise of randomized controlled trials and meta-analysis to the 'gold standard' (Sackett *et al.*, 1996) has played a dominant role in the 'simplification' of what can be considered evidence, as it establishes a homogenized view of how knowledge in the domain should be made available and interpreted (Booth, 2011);
- Narrow scope and unstructured knowledge (i and v): with that characteristics can be cited the proposal of Santos and Travassos (2013). In their work, the authors explicitly mention the heterogeneity in the type of data (quantitative and qualitative) and the research methods used in the area. As a consequence, the modelling process aspects which receive most attention focus on the way that concepts should be identified and organized, and how cause-effect relationships should be valued using a *Likert* scale based on qualitative or quantitative reports;
- Wide scope and structured knowledge (iv and viii): in this category, the work of Russ *et al.* (2011) has similarities with van Valkenhoef *et al.* (2013) as regards its focus on controlled experiments, although it is proposed as domain-independent. The representation consists of a workflow notation and inference facilities for experiments parameters (inputs) and measurements (outputs). The systematization of knowledge generated by controlled studies associated with

the way the authors suggest that observation and interpretations in a scientific investigation life cycle should be distinguished and modelled according to the representation is what was considered as modelling process in this case;

- Wide scope and unstructured knowledge (iii and vii): Mancini and Buckingham-Shum (2006) illustrate how even in a situation where there are no references to the domain properties or to methodological aspects, it still possible to have indications of how the modelling process should be conducted. In the paper, the modelling process is manifested in the idea of 'sense-making'. By using this notion, modelling is part of making sense as it is seen as a means to share an understanding with others through the use of a symbol system. The authors commit a significant portion of the text to describe, under the Semiotics (Peirce *et al.*, 1931–1935) and Cognitive Coherence Relationships (Sanders *et al.*, 1993) perspectives, how concepts and relationships should be thought of and modelled.

4.4 Scientific knowledge engineering concerns and procedures

The design and construction of a computational infrastructure for managing and performing inference with scientific results is rather complex activity. In general terms, it seems there is no baseline for what should be compared amongst SKE proposals or a systematized organization of the main characteristics that have to be addressed in such infrastructures. This makes the comparison or correlation of the proposals difficult as each one focuses on different aspects of its construction and justifies its usefulness using different theoretical frameworks (e.g., computational, epistemological, and cognitive).

Due to the constant evolution of scientific domains, scientific thinking and its methodological and conceptual structures, this paper adopts KE as a modelling process paradigm. The rationale for this decision is because the modelling paradigm carries in its concept: the iterative refinement of knowledge representation and the computational infrastructure which manipulates this representation.

One way to enumerate SKE concerns and procedures from a KE standpoint is to put the discussion on two levels of abstraction, which we will call the management level and the operational level. The management level has the issues related to the process, roles and relevant decisions which should be considered in the design and construction of the knowledge model and computational infrastructures — a discussion on why the management level is important can be found in Freiling *et al.* (1985). The operational level has the application and use of design techniques to define the model and knowledge base, inference methods, and procedures for knowledge acquisition. Due to the preliminary nature of this work, and also given its conceptual delineation purpose, this work concentrates on the management level. The idea of first addressing this level is to provide means to ‘instantiate’ new SKE projects and leave the operational level to be addressed on a case-by-case basis. The next couple of paragraphs briefly discuss the operational level before returning to the management level.

On the operational level, there are several modelling approaches[12] for the design of the knowledge representation and knowledge bases available in KE as a modelling process paradigm. Just as an example, the two main modelling approaches according to Studer *et al.* (1998) are the role-limiting methods (Mcdermott, 1988) and generic tasks (Bylander and Chandrasekaran, 1987). Both are based on the concept of problem-solving methods. Problem-solving methods can be characterized by (i) the specification of what inference actions have to be performed to resolve a specific task, (ii) the sequence and conditions in which these actions should be triggered and (iii) the role domain knowledge plays in each inference action (Studer *et al.*, 1998). In the context of SKE, knowledge engineering aspects such as determining what the tasks would be, defining ways to elicit the order of inference actions or establishing the role of domain knowledge (i.e., scientific results) in each action, still have to be better mapped to the scientific domain — be it by using existing approaches or in devising a new one specifically for SKE. Indeed, the search for the systematization of the modelling process was one of the factors that allowed the development of problem-solving method reusable libraries (Studer et al., 1998) and favored the rise of languages for the specification of knowledge-based systems (e.g., CML — Conceptual modelling Language Schreiber (2000)). These libraries offer basic combinations of knowledge structures and inference strategies ready to solve some kinds of problems. In an analogous way, it is possible that similar movements come up in SKE while these aspects are being addressed in SKE. An initial step in this direction can be found in Hunter and Liu (2010) where the different characteristics of representation and reasoning formalisms are related to different properties of scientific knowledge domains.

[12] It should be said again that in SKE modelling activities occur at two moments. One is the modelling (or ‘translation’) of scientific results into knowledge representations. The other is the modelling (or design) of the knowledge representation itself as an operational model with a particular behaviour, given a set of specific conditions.

The lack of a better organization of the operational level for SKE could be seen in the papers reviewed, as described in the previous section. Most works did not justify the decisions that led to the choice of a representation formalism, inference methods or design trade-offs. It seems that, since these issues are not detailed in the papers, most of the decisions were made intuitively or based on experience. In fact, this often happens when new domains are explored in KE. In these situations, the knowledge engineer has few opportunities to leverage from existing approaches to design the knowledge representation or reuse parts of ready-made models (Motta *et al.*, 1990). This, in turn, hampers the analysis of the nature of knowledge by the knowledge engineer who, otherwise could be helped by the incremental structuring of existing knowledge representation models in successive abstractions.

All these issues are themes for research. But as mentioned earlier, they will be left to future SKE initiatives. Returning our attention to the management level, the following section presents a management level process based on several works which proposed some kind of process organization for KE projects (Freiling et al., 1985; Fellers, 1987; Dibble and Bostrom, 1987; Rook and Croghan, 1989; Motta *et al.*, 1990; Plant, 1991; Studer *et al.*, 1998; Schreiber, 2000). It should be noted that, on the management level, the focus will be more on what should be done than on how it should be done. In addition, process steps are just briefly described, in trying to pinpoint particular aspects of SKE — for a detailed description of different processes for KE the aforementioned references can be used.

4.4.1 Scientific Knowledge Engineering projects: a step-by-step approach

It is possible to identify three main phases in SKE projects (Figure 11): scope determination, specification and definition of the knowledge representation and the process associated to its manipulation, and the construction of the computational infrastructure. Divided into these stages, the proposed process aims at manifesting the relevance of the following aspects: (i) adequate handling of the complexity involved in the characterization of scientific knowledge to be modelled, dedicating a stage only to scope definition, (ii) detailed analysis of the way (i.e., process) the researchers have to translate scientific results into the proposed knowledge representation, (iii) prioritization of the knowledge representation over the definition of possible inferences that could be obtained from it, and (iv) continuous process iteration aimed at the refinement of the designed knowledge representation and inference methods, in addition to enabling the continuous revision scope of knowledge accommodated by the representation.

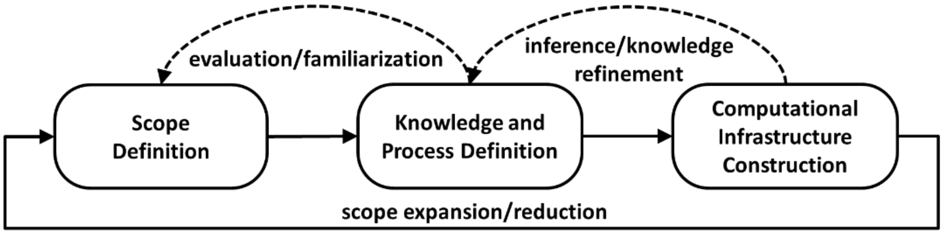

evaluation/familiarization
inference/knowledge refinement
Scope Definition
Knowledge and Process Definition
Computational Infrastructure Construction
scope expansion/reduction

Figure 11 – Main Scientific Knowledge Engineering phases

The roles involved in these stages are those of the domain specialist and the knowledge engineer, which were mentioned previously and are commonly found in the technical literature (Fellers, 1987, Studer *et al.*, 1998), and the not always mentioned software engineer role (Freiling *et al.*, 1985, Rook and Croghan, 1989). The tasks performed by the domain specialist and software engineer are relatively straightforward, being the former the one who has practical experience in the domain area used to define the knowledge base and the latter the one responsible for the computational infrastructure construction.

The knowledge engineer, on the other hand, accumulates more than one function in SKE. The first is associated to the design of the knowledge and reasoning formalisms (stages 1, 2, and 3) and the second relates to the modelling of scientific results using the designed knowledge representation (Figure 6A). Although these functions are usually performed by different persons, we decided to keep them under the role of knowledge engineer because of the intense focus on knowledge modelling, although with different purposes. The first role, which can be called the meta-level as it focuses on the design of the knowledge representation and inference methods, has the attributes commonly found in the KE literature and is preferably done by people with a Computer Science and Artificial Intelligence background. The second role entails additional attributes associated to modeling the scientific results into the knowledge representation, which can be performed by researchers.

Scope definition phase

The first stage in the process has a more exploratory nature and aims at defining the scope and the complexity of the problem (i.e., domain) at hand. The objective of this stage is to find ways to cope with the scientific knowledge heterogeneity by trying to break it down into tractable parts. In the context of SKE this means that it would be quite unusual whether a knowledge representation for a wide scope could be created in only one cycle. Thus, the process has to be iterative to allow the knowledge representation to be reworked and improved, considering a wider or more specific scope.

This initial stage for scope definition should include the following steps:

- Familiarization (step 1)

  This first step can be conceived as analogous to the interview techniques used by a knowledge engineer. The difference, however, is that in SKE knowledge is already explicit and scientific papers can be used as the source of specialist knowledge — which does not exclude the possibility to use domain specialists (i.e., researchers) as source of knowledge as well. In using these sources the objective is then to familiarize one with the type of knowledge to be represented, trying to capture the specificities of the scientific results of the defined domain scope. Hence, this step can be initiated from a representative set of papers, which allows one to grasp how the results of scientific investigations are described and what their main features are. As the universe of scientific papers can be very large, what is 'representative' depends on each case. One way to find this representative set of papers is to restrict the scope, using dimensions such as domain or knowledge area (e.g., specific as drug-drug interaction or more generally as Software Engineering), types of research questions (e.g., studies which asks exploratory, rate or cause-effect questions (Easterbrook *et al.*, 2008)) or research methodologies (e.g., case studies or controlled experiments).

- Knowledge initial organization (step 2)

  The purpose of this step is to try to identify some kind of regularity in the way scientific results within a defined scope are presented. This regularity can be observed in different levels. It can be seen in the way that the scientific discourse is expressed (e.g., reporting guidelines), what is the culture underlying how the scientific results are justified (i.e., if there is more focus on explore semantic and conceptual relationships among the observed entities or on investigate cause-effect variables of phenomena (Dixon-Woods *et al.*, 2005)) or what are the main models and mechanisms used to structure the knowledge in the area (e.g., protein structures in Biomedicine).

  There are many possible alternatives to accomplish this initial knowledge organization. One attempt can be the classification of the recurring types of concepts and relationships within the defined scope (as, for instance, de Waard and Schneider (2012), van Valkenhoef *et al.* (2013)). Another alternative is the building of a knowledge acquisition grammar to express the facts and rules of scientific results (Freiling *et al.*, 1985) (as done in Dinakarpandian *et al.* (2006)). Raw text can be also used. In this case, extraction forms commonly used in systematic literature reviews can help organizing knowledge as it works as a 'questionnaire' to be 'answered' by each one of the papers collected in this scope definition stage.

Knowledge and process definition stage

The second stage's main outcomes are the formalization of the knowledge representation and description of how it should be operated. Knowledge definition should seek to determine how the common objects, relationships, observations, and events used in the defined scope are represented. At this point, any concerns with the definition of inference strategies should be avoided, as the representation itself helps thinking about the interpretations and heuristics necessary to achieve the semantic connections (e.g., between an observation and a conclusion). Clearly, the reverse direction — with the definition of inferences before the representation — can also be followed. However, it will potentially be more laborious, as knowledge initially extracted from the papers in the previous stage often consists of references to elements of the domain knowledge and not to the possibilities or interpretations that can be obtained from their arrangement in a single knowledge base.

The stage defining the knowledge representation is essential, as it is a chance to detect and align possible incongruences in the way specialists communicate their knowledge and how this knowledge should be represented to allow its computational manipulation (Fellers, 1987). For this reason, especially in SKE where the distance between the representation model and the form with which the knowledge is conveyed can be large, it is important that the representation is followed by the definition of the process which should be used to extract the scientific knowledge from the papers.

The following steps are suggested in defining the knowledge and process:

- Domain conceptualization (step 3)

  In this step, the knowledge engineer tries to establish a global structure from the data collected in the previous stage, with the purpose of producing an abstract model of the problem in terms of taxonomic hierarchies, tables, flow diagrams, cognitive maps, object-oriented class models, or any other tool for conceptual modelling. At this point, it should be possible to determine whether the knowledge representation is going to be more oriented to the scientific discourse (i.e., container) or to scientific assertions (i.e., content). It can also be interesting to consider some kind of partitioning of knowledge into modules forming the global structures as, for instance, the module associated to knowledge on the research methods adopted and the module related to the domain itself. This decoupling can help eliciting connection points amongst them and on how they can be used together in the computational infrastructure.

This step is considered crucial in KE technical literature, as it is the moment where the focus is concentrated in the abstract level of the problem, leaving the representation implementation issues aside. Furthermore, it offers the possibility to explore the domain area and eventually conduct some kind of validation of the concepts and relationships structures present in the defined scope. This progressive specification starting from the conceptual level up to the implementation level appears with different names in the technical literature such as primary representations and domain specifications in Plant (1991), shared and external representations in Fellers (1987) and micro and macro-knowledge in Rook and Croghan (1989). As a result of this incremental specification and given the relative lower commitment compared to the following steps, this point can be an appropriate moment to consider if a new iteration is necessary for better familiarization with the domain or for scope refinement.

- Knowledge representation definition (step 4)

  The definition of the knowledge representation should take knowledge organized in the previous steps, especially the domain conceptualization, and convert it into a specific representation scheme (e.g., frames, object-oriented, production rules, formal specifications, or logic programming). It is in this concrete specification that representation model instances can be created and have their construction restrictions validated. A specific representation scheme is also the most preponderant factor for what are going to be the alternatives for inference, so it should be chosen wisely. It should be pointed, at this stage, that even though the previous steps could be specified using informal tools (e.g., raw text and tables), the higher the rigor used in the previous steps, the more direct the representation definition using a specific representation scheme will be. For instance, the use of a knowledge acquisition grammar can help the specifying of production rules that in turn tend to facilitate its implementation in a specific logic programming language (e.g., Prolog) or approach (e.g., justification-based truth maintenance system (Forbus and DeKleer, 1993)).

  At this point, to evaluate these different decisions involved in the knowledge definition step, the possibility of its evaluation using real scientific results extracted from research papers could be considered. In fact, most of the papers found in the literature review used real data to present proof of concept of their proposals. However, few considered this in the preliminary stages based only on the definition of the knowledge representation such as in Santos and Travassos (2013). Again, the opportunity for iteration can be used if one needs to adjust some aspect of the domain conceptualization or even improve the familiarization with the domain, before constructing the computational infrastructure.

- Knowledge utilization process definition (step 5)

  This step aims at detailing how scientific results should be interpreted and manipulated to allow their modelling using the knowledge representation defined in the previous step. We suggest that a well defined process for modelling of scientific results using the knowledge representation should, at least, consider the following aspects:

  - Knowledge extraction: the process should precisely indicate what information should be extracted from the papers. If there are specific sections that usually contain the information needed, this can be suggested as well — reporting guidelines commonly discussed in many scientific domains can be used for this definition. Thus, the absence of the expected information can indicate the impossibility of adding a specific result to the knowledge base, as it will not be possible to instantiate the knowledge representation for this result. Apart from papers, scientific findings can be directly translated to the knowledge representation at hand. In this case, the process must indicate what information is necessary to provide (e.g., description of cause-effect relationships, definition of experimental variables, or identification of relevant contextual aspects).
  - Model instantiation: it is necessary that researchers interpret the results and derive their meaning before being able to represent its results appropriately. The process should assist the research in how to identify the constructs, hypothesis, analogies, models commonly used in the domain and how they can be mapped to defined knowledge representation. For instance, the indication of how to start from raw text and identify the concepts using text codification as mentioned before. Another example is the suggestion of the order in which the information should be used (e.g. first defining the concepts and then trying to identify the relationships). It is also interesting to remember, as discussed in previous section, that the lack of some kind of latent structure in the scientific results requires a higher guidance level in the model instantiation. In the reverse direction, the presence of some kind of structure in the results tends to improve the

directness between the representation and knowledge. But, on the other hand, any incongruence in this mapping makes the representation difficult to use (Chua *et al.*, 2012).

- Representation completion: the iterative nature of a modelling process requires some indication of when its completion should be considered. Contrary to other processes of knowledge acquisition where the process is intrinsically open, given that knowledge is elicited from human specialists, in the case of SKE the elicitation is relatively less open — it is restricted to the results of one study at time. As a result of that, it is possible to suggest some criteria based on the knowledge representation properties or on the modelled knowledge (i.e., scientific results). Some examples of criteria include theoretical saturation (Lewis-Beck et al., 2004), commonly used in text codification and qualitative analysis in general, and criteria involving some representation coverage aspect (e.g., were the semantic capabilities of the representation fully explored?). One way or another, the basic idea underlying these criteria is to allow the researcher to get a perception as to whether the representation instance from a study result 'makes sense' (Chua *et al.*, 2012).

Computational infrastructure construction stage

The construction of the computational infrastructure stage was divided into two steps: the definition of inference strategies and the infrastructure implementation itself. The definition of inference strategies was placed as part of the infrastructure construction due to the suggestion of Freiling *et al.* (1985) who mention that the outcome documentation of the inference definition step should be a 'running inference engine'. The computational infrastructure implementation step, on the other hand, corresponds to the ordinary Software Engineering activities.

- Inference strategy specification (step 6)

  Inference strategies should be developed based on the prior understanding of the underlying processes used by researchers in the production and use of knowledge (adapted from Dibble and Bostrom (1987)). In the case of SKE, the interest lies in examining how scientists make sense of the available scientific knowledge, be it in the establishment of new hypotheses, in the strategies used to answer research questions or simply in exploring the state of the art.

  The specification of inference strategies also depends on the knowledge characteristics and the domain defined as scope and in the semantic richness of the knowledge representation defined in the previous step. Both have a direct effect on the repertoire of possible inferences that can be specified. There are many alternative strategies for automated reasoning with scientific knowledge, including the ones identified in the literature review and also examined by Hunter and Liu (2010), amongst which we can mention deduction and abduction queries used to deduce or abduct (assume) new knowledge from the knowledge base; integrity verification to identify possible contradictions/consensus in the available knowledge; and knowledge aggregation, where the combination is obtained from the reduction of conflicts and redundancies based on the semantic properties of the representation.

- Computational infrastructure implementation (step 7)

  The software engineering activities involved in this step are specification, design, construction, test, and maintenance. Most of the system specification, at least the core related to representation and reasoning mechanisms, is already partially developed in the previous steps. Still, it is necessary to specify several other elements of the computational infrastructure that wrap this core, such as information registering, navigational aspects, validation, system-user interaction among others. Furthermore, one also needs to address non-functional aspects such as security and usability.

  The design activities consist of laying out the architectural organization of the system and structuring the high and low level design elements. Particularly as regards the architectural aspects, the decoupling of the knowledge representation from the inference engine should be considered, following the same idea of the SKE process considering these two aspects in different steps. Yet, while the idea of having two separated steps was to put the specification in the 'right order', on the implementation level, the purpose is to improve the extensibility of the infrastructure by having these two elements in different modules.

  The construction and testing activities strictly follow software engineering procedures even though verification and validation (Wallace and Fujii, 1989) procedures are made more difficult due to the wicked nature of the SKE problems. And, at last, infrastructure maintenance activities can originate from corrective or perfective necessities, or be initiated as a result of new iterations of the SKE process as shown in Figure 10.

4.5 Conclusion

In this chapter, Scientific Knowledge Engineering was conceptualized as a research area concerned with the design and construction of infrastructures for the computational manipulation of scientific knowledge. Based on the Knowledge Engineering theoretical foundations the current technical literature describing scientific knowledge infrastructures was revised. A separation between KE and SKE can be seen as artificial at first, but the literature review could identify many of its salient characteristics and concerns: container and content knowledge representations, technologies and techniques used for knowledge formalization, possible answers supported, the possibility of scientific result aggregation, graphical representations used, and the modelling process used by researchers to represent their inquiry results. In total, 22 papers were found over 8 years, showing an interesting development of the area. The rise of SKE can be related to the massive increase in the amount of scientific production in the last decades and the constant expansion of computational methods and artificial intelligence over different domains. It is interesting to note that the number of 22 papers could be significantly higher considering the fact that the literature review was not systematic (and a new trial with a more structured search string yielded over 13,000 results).

Apart from using KE as a foundation for delineating SKE as a research area, another contribution of this work was the distinction and delimitation of different approaches used for the computational manipulation of scientific knowledge, namely data-intensive approaches and computational discovery of scientific knowledge in comparison with SKE. The papers analyzed in the literature review also supplemented this delineation. The review showed that current SKE proposals are in general too focused on the representational aspects of their formalizations without giving proper justification as to why they were chosen or what the design decisions were made for the representation model. We understand that in the current preliminary stage of SKE, proposals are more concerned with the feasibility of the knowledge infrastructures than creating prototypes and conducting evaluations to test technologies and techniques from KE. Nevertheless, the maturity of the area will only be achieved with these kinds of justifications. Aiming at pushing SKE in this direction, we listed a series of main concerns related to the design and construction of scientific knowledge infrastructures — again, based on KE developments.

At last, we should mention our emphasis given to the methodological and procedural aspects of SKE rather than giving attention to technical decisions and implementation aspects of KE. We think that this was necessary as the goal of this chapter was to present SKE and also because of the need to first indicate ‘why’ and ‘what’ to do, and then examine ‘how’ it can be done. In fact, the literature review has already touched on several technical aspects, but the subject can be investigated in much more depth in new SKE initiatives. Nevertheless, specific details about implementation aspects of computational knowledge infrastructures will be detailed when we present the infrastructure created to support the research synthesis method proposed in this thesis.

5 The Structured Synthesis Method

In this chapter, we present the main contribution of this thesis: the Structured Synthesis Method.

5.1 Introduction

Previous chapters form the conceptual domain upon which we develop a research synthesis method for SE. All of them discussed some aspect of knowledge representation from the epistemological, mathematical, and computational perspectives. Combining these perspectives, in this chapter we propose a research synthesis method, which we denominate Structured Synthesis Method (SSM), aiming at supporting the establishment of a model for evidence-based practice in SE based on a unified evidence representation. Taking advantage from the representational flexibility of theories and the aggregation function of D-S theory, SSM has elements from integrative and interpretive synthesis methods, although it is more focused in giving an integrated summarization of related evidence. Computational aspects will be addressed later complementing the method rather than part of it.

The term research synthesis is usually assigned to any method that is applied to make an *integrated* summarization or provide new *interpretive* explanations from a collection of studies investigating a specific topic or research question (Cruzes and Dybå, 2011b). Common to virtually any research synthesis method is how knowledge extracted from studies suffers intense transformation. This transformation usually aims at translating individual results to a representation allowing them to be analyzed in the same perspective. This is the case, for instance, in meta-analysis (James, 2000, Ciolkowski, 2009) where effect sizes are used to estimate and compare the order of magnitude of the quantitative outcomes, in thematic synthesis (Cruzes and Dybå, 2011a) which uses text codes and cognitive maps, and in meta-ethnography (Silva *et al.*, 2013) where studies' concepts are translated into each other study.

As previously mentioned, the nature of evidence has been aim of intense discussion in the technical literature of different areas (Upshur, 2000, Upshur *et al.*, 2001, Cohen *et al.*, 2004, Rycroft-Malone *et al*., 2004, Scott-Findlay and Pollock, 2004, Tarlier, 2005, Guyatt *et al.*, 2008, Walters *et al.*, 2009, O'Grady, 2012). Therefore, it is difficult to establish a clear scope of what is and what is not possible to represent and aggregate with the procedure proposed in this chapter. We believe this is a hard problem to state objectively. Trying to make at least an indication about this issue, we define its scope based on the different kinds of research questions. Easterbrook *et al.* (2008) define several kinds of research questions categorized into exploratory, base-rate (frequency or patterns of occurrence), relationship, or causality. Given the causal and moderation relationships present in the theoretical structures, we establish a logical link to conclude that it is capable of modeling evidence indicating some kind of causality.

Furthermore, regarding the different epistemological dispositions (e.g., positivist and constructivist) upon which evidence is obtained, SSM also uses the causality notion to link different types of studies' (qualitative and quantitative) results using the arguments found in the work of Mahoney and Goertz (2006). According to them, both quantitative and qualitative studies are usually concerned with uncovering or observing cause-effect relationships. However, qualitative research "explains individual cases; using the causes-of-effects approach" while quantitative research "estimates average effects of independent variables; using the effects-of-causes approach". In this way, in our understanding, we can conciliate different epistemological perspectives associated with research methodologies used in SE primary studies and, at the same time, be aligned with the evidentialism theory of knowledge as defined in the beginning of this Thesis.

This chapter is organized as follows. Before introducing SSM, we describe in Section 5.2 its conceptual ground from a research synthesis perspective. Then, we split the presentation of SSM in two sections. We first present in Section 5.3 the core part of the method defining how theoretical structures and D-S theory are used for aggregation. After that, in Section 5.4, we detail the SSM steps. And Section 5.5 concludes this chapter.

5.2 SSM process conceptual ground

We propose SSM process guidance based on existing research synthesis methods. Therefore, the focus of this section will not be on comparing existing methods, but rather identify their main characteristics, which were directly used in, supplemented, or used to justify SSM process definitions. As a direct consequence of this, it is important to say that we try to avoid as must as possible to 'reinvent' existing methods' orientations. On the contrary, we will try to incorporate their definitions as aforementioned. Using this strategy, we aim at focusing on our goal of investigating representation and engineering of scientific knowledge to support knowledge translation in SE through the proposition of the research synthesis method presented in this chapter.

The distinction between integrative and interpretive methods was originally defined by Noblit and Hare (1988) and precisely determinates the objectives of a research synthesis. An integrative synthesis involves, in general terms, the combination and aggregation of data incorporating studies' findings into one another. Integrative methods, such as meta-analysis, requires a basic comparability between the studied phenomena so that data can be collectively analyzed (Dixon-Woods *et al.*, 2005). The main result of an integrative synthesis is often the identification of some kind of pattern in the data, usually indicating the size and direction of cause-effect relationships. An interpretative synthesis, on the other hand, aims at developing and describing concepts in such way that they can be organized into a theoretical structure, as arguments in a narrative line or any other form that allow figuring out a connection between the analyzed studies (Dixon-Woods *et al.*, 2005). In this way, interpretive syntheses are commonly used to bring a new perspective about the phenomena previously considered independently or to refine the understanding about a phenomenon.

There are several works whose primary focus is reviewing research synthesis methods. Two of them (Dixon-Woods *et al.*, 2005, Cruzes and Dybå, 2011b) were used for screening existing methods. Considering just these two literature reviews about research synthesis methods, we could identify 15 methods (from which 9 are cited in both). From these 15 methods, we identified 4 which could support the definition of SSM: thematic synthesis (interpretative) (Cruzes and Dybå, 2011a), meta-ethnography (interpretative) (Silva *et al.*, 2013), case survey (integrative) (Yin and Heald, 1975), and qualitative comparative analysis (integrative) (Yamasaki and Rihoux, 2009). Besides these 4 methods, we have selected one additional method: theory building with statistical meta-analysis (integrative) (Yang, 2002). We have not defined any explicit criteria for selecting these methods. The selection was based on the subjective interpretations about what could be applicable to *activities*[13] necessary to manage and use the knowledge representation and the aggregation procedure. Several other methods could also be taken as basis for the SSM process definition, but were left out because most of their methodological features are covered in the 5 cited methods. It is the case, for instance, of the content analysis (Evans and FitzGerald, 2002) and meta-summary (Sandelowski *et al.*, 2007) methods.

Following next, we give a brief description of each of the 5 cited methods:

- **Thematic synthesis:** involves the identification of recurrent themes in technical literature, summarizing results from different studies related to each theme (Dixon-Woods *et al.*, 2005). In the process defined in (Cruzes and Dybå,

[13] The difference between representation and activities was discussed in Chapter 1.

2011a), five steps are indicated: data extraction, data coding, translation of codes into themes, creation of a higher-order model and trustworthiness assessment. One of the key principles driving this process is the increasing abstraction levels, starting from the text, passing through the identification of themes and concluding with a higher-order model. It is also interesting the use of a graphical representation for the model, usually a cognitive map (Cruzes and Dybå, 2011a);

- **Meta-ethnography:** as a synthesis method, it seems that the most distinctive feature of meta-ethnography is the translation (reciprocal or refutational) procedure, which supports the researcher in identifying and inducing concepts and relations by considering findings in one study that are like the findings of other studies (Silva *et al.*, 2013). This translation can be done literally, concept by concept, or idiomatically, preserving the meaning of the text. In any case, the objective is to produce new interpretations using analogies to consolidate different studies' results or by developing concepts and relations that can capture the findings together;
- **Case survey method:** Conceptually simpler than the other methods, the case survey method consists essentially of a closed questionnaire, which is applied to each (case) study (Yin and Heald, 1975). Since the questions are closed-ended, the summarization comes from the analysis of the answers distributions. In Yin and Heald (1975) and Larsson (1993) three methodological concerns are emphasized: (i) a reliability of the synthesis as a parameter to assess its replicability, (ii) the differentiation between weak and strong responses to estimate the confidence on each answer and on the synthesis as a whole and (iii) an explicit definition of criteria for excluding studies from the synthesis.
- **Qualitative comparative analysis:** this method assumes that a given outcome can be an effect of different combinations of circumstances so that the notion of causality may be understood in terms of the sufficient and necessary conditions distinction (Dixon-Woods *et al.*, 2005). These conditions are usually associated with studies' dependent and independent variables, which are tabulated using dichotomous values – in the case of crisp set qualitative comparative analysis – to indicate the presence or absence of a cause, effect or contextual aspect. The summarization is then achieved by using first-order logic to determine which descriptive inferences the data supports, constituting the so-called explanatory model or, in other words, what are the sufficient and necessary conditions for the explored cause-effect relations. Although it does not use a graphical representation, qualitative comparative analysis is the most similar method to

SSM whether we consider how it explicitly represents both cause-effect relations and contextual aspects. Thus, the selection of these conditions, as well as the identification of concepts and relations in SSM, is fundamental. There are many approaches described in the technical literature to accomplish this selection (six mentioned in Yamasaki and Rihoux (2009)), including the two cited here: the comprehensive approach, which takes existing theories to determine the conditions that will be explored in the primary studies, and the inductive approach, in which the conditions are selected based on knowledge acquired from primary studies, in similar way to other methods such as meta-ethnography;

- **Theory building and statistical meta-analysis:** as result of combination of two research strategies, its underlying idea is that theories can be developed as a result of a statistical meta-analysis study. Hence, rather than detailing how these two approaches can be combined and used together as described by Yang (2002), the method is cited to indicate the practicability of using theory building techniques to support research synthesis. One of the Yang (2002) main definitions to use theories as a tool for meta-analytical synthesis, is the distinction between theory abstract constructs and observable indicators from the empirical level. Based on this idea, Yang (2002) defines three research domains: theoretical, empirical, and measurement. In the theoretical level, abstract concepts and entities are defined, which are used to define a theory. Both empirical and measurement domains have definitions to operationalize variables. Empirical domain contains all known existing definitions regarding how a construct can be operationalized based on the definition of the theoretical domain whereas measurement domain contains representative samples of observations from the empirical domain for the same construct. Based on this distinction, the explicit link between theoretical and empirical levels can be formed, and meta-analytical statistic instruments are used to verify if the theoretical relationships are statistically significant according to the variables used to measure them.

5.3 Method core: Structured Aggregation

The Structured Aggregation is the essential element of SSM. It is basically formed by the evidence knowledge representation adapted from the diagrammatic representation proposed by Sjøberg *et al.* (2008) for scientific theories in SE and a formalization for uncertainty using the D-S theory. In the aggregation, the graphical representation is used as a mean to make knowledge explicit and to support researchers to reason about evidence information content during aggregation whereas D-S theory is used to describe uncertainty about evidence and to provide mathematical instrument for aggregation. The name Structured Aggregation stems from the knowledge representation used for which we preferred to use the term theoretical structure. As it could be noticed, the graphical representation plays a central role in the aggregation.

5.3.1 Evidence diagrammatic representation and uncertainty formalization

A complete reformulation of the original syntactic (visual) notation, which is based on UML and used in Santos and Travassos (2013), was implemented in order to try to improve the cognitive effectiveness of how easily and accurately the diagram can be processed (Moody, 2009). Following prescriptions from (Moody, 2009), major changes include the absence of explicit cause-effect and moderation connections to reduce the diagram complexity and a better use of retinal variables (e.g., visual shapes) to improve the perceptual discriminability.

Only syntactic constructs were affected in this reformulation, all ten semantic constructs were not modified, although three ones were added. The new syntactic definitions are described as follows (Figure 12). *Value concepts* (which are *archetypes*, *contextual aspects* and *causes*) are represented as rectangles. *Causes* are differentiated from the other *value concepts* by text decoration, using a bold font and a label ‘C1’ (for ‘cause 1’ – at most two causes can be present in a diagram[14] when they are being compared, in this case C1 and C2). *Variable concepts*, which are *effects* and *moderators*, are represented as ellipses and parallelograms, respectively. *Is a* and *part of* relationships are kept the same, using UML notation, and *property of* relationships are now represented with dashed lines. *Cause-effect* relationships are not connected by lines anymore as they can be interpreted by the visual shapes: if there is an *effect* and a *cause*, then there exist a *cause-effect* relationship between them. *Moderation* relationships are also interpreted implicitly between effects and moderators, but text decorators are added to map which moderators affects what effects. A label ‘M_1’ besides a moderator identifies it whereas when the same label is found besides an

[14] In the original work of Sjøberg *et al.* (2008), only one cause can be represented. We extended their representation in other to represent comparative evidence. In multi-causality scenarios, it is necessary to create one model for each cause.

effect indicates that it is affected by that moderator. A second moderator is identified as ‘M_2’ and so on. There is no restriction for the number of moderators.

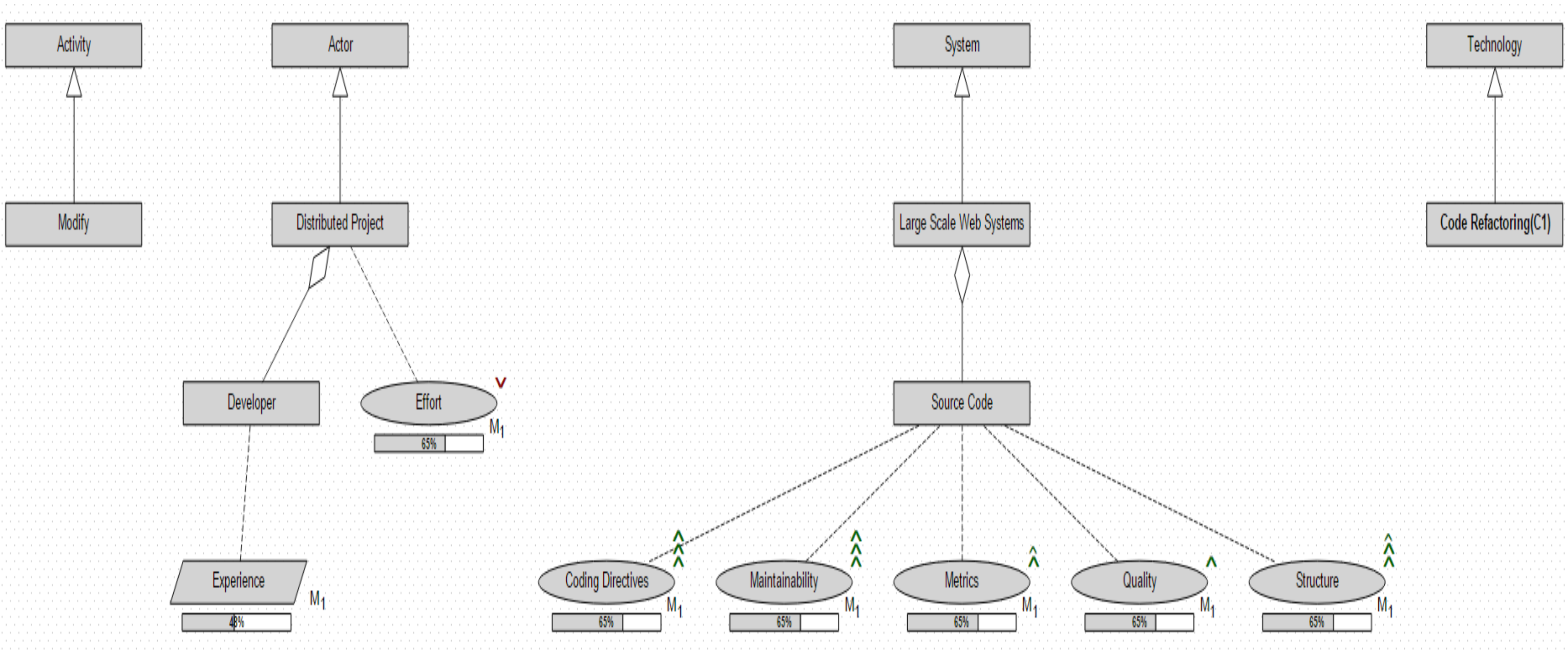


Figure 12 – New concrete diagrammatic syntax for theoretical structures (original concrete syntax for the same evidence is shown in Figure 2). All semantic constructs from Sjøberg *et al.* (2008) are still present, although some implicit.

As the representation from Sjøberg *et al.* (2008) was not originally proposed for evidence representation, three new semantic constructs were added in order to accommodate more information elements from evidence. One new semantic construct introduced is the possibility of having two causes compared. As previously mentioned, the only syntactic modification in this regard is the addition of a text decoration. When two causes are present, effects are interpreted comparatively with one cause to another.

The other two added semantic constructs are associated with the representation of the effect intensity. The original proposal do not have this information represented, although it usually is determined in the text description (Table 1). As we intend to represent both qualitative and quantitative evidence, it has been defined a semi-quantitative seven-point *Likert* scale from a negative to positive effect: strongly negative (SN), negative (NE), weakly negative (WN), indifferent (IF), weakly positive (WP), positive (PO), and strongly positive (SP). In the diagram syntax, this is represented with symbols indicating one of these values. For instance, the symbol '︽' indicates a strongly positive effect, as is the case of Maintainability in Figure 12, and '⌣' indicates a weakly negative effect, as is the case of Effort in the same diagram. A second aspect regarding the effect intensity is the uncertainty associated with it. The uncertainty is represented with a 'bar' under each effect with a percentage value according to the confidence about the observed results. Moderators also have uncertainty associated, but they do not have intensity defined. Only their direction is defined using the scale: inversely proportional (IP), indifferent (IF), and directly proportional (DP).

Clearly, just diagrammatic elements would not be sufficient for uncertainty formalization as they are just used to present the information. Thus, to formalize uncertainty related to evidence results we use D-S theory. The frame of discernment is defined based on the scales established for the effect intensity and moderation direction: Θ_e = {SN, NE, WN, IF, WP, PO, SP} and Θ_m = {IP, IF, DP}. Then, basic probability assignment function is used to assign belief values to one of the frame of discernment elements. In other words, this means that we are making explicit the uncertainty associated with this kind of relationships. Therefore, based on these definitions, it is possible to define, for instance, that code refactoring strongly positively affects maintainability with 0.65 belief: $m_{1\text{-maintainability}}(\{SP\}) = 0.65$. One could ask at this point why there is no uncertainty associated with structural relationships (*is a*, *part of* and *property of*). The reason for this is that structural relationships are used to describe contextual attributes which are usually controlled or determined by the investigated environment and, thus, there is not much room for ambiguity in its determination.

The value 0.65 used in this case was just a hypothetical example. A systematic procedure for estimating confidence values will be presented later in this chapter. In fact, in Santos and Travassos (2011) both qualitative and quantitative data were used in the analysis. Thus, just the differentiation between the data type used to explain each effect could be exploited to estimate the belief values. Moreover, when a belief is assigned to a proposition with more than one element (e.g., $m_{1\text{-metrics}}$({WP,PO}) = 0.65), it should be interpreted as range, widening the uncertain about the real value (*i.e.*, weakly positive or positive).

Another important aspect about the scale is the role of the 'indifferent' value. Indifferent means that a potential effect is not affected by the cause being considered. Thus, if for instance it is stated that $\text{Bel}_{\text{-quality}}$({IF}) = 0.9, then given the high belief value associated with *indifferent value* for the *quality effect* it can be concluded that this relationship, and the related effect, can be removed from the theoretical structure. Analogously, the same line of argument can be thought to include new effects to a theoretical structure. Therefore, if a new variable concept (effect or moderator) is identified in an evidence and the theoretical structure does not have it (i.e., there is no belief committed to any of the scale values), then the new relationship is just added to the evidence model. The decision about when a new theoretical structure must be created or not due to new evidence is discussed in more detail in next section and supplemented with the SSM process description in Section 5.4.

5.3.2 Evidence aggregation procedure

The main goal of the structured aggregation is to combine evidence results in order to identify patterns or trends in the data as a whole. In the discussion that follows, we suppress details about the evidence modeling process using theoretical structures and assume that it has been correctly done in the examples presented. Our focus at this point is, thus, to show the mechanics involved in the aggregation using the representation and the uncertainty formalism. Still, using words from Higgins and Shirley (2000), it is important to say that 'regardless of the method used to create the theory, the amount of evidence accrued to support it promotes confidence in its use by practitioners and scientists'.

In Figure 13, the aggregation procedure is depicted in its main steps and decisions. The basic idea is to identify whether evidence can be mapped to an existing theoretical structure, according to the concepts describing the context, and aggregate its causal and moderation relationships using D-S theory. If an existing theoretical structure can not be mapped to the evidence being considered then a new theoretical structure is created to represent that single evidence. When it happens, the resulting theoretical structure will have a low empirical support level.

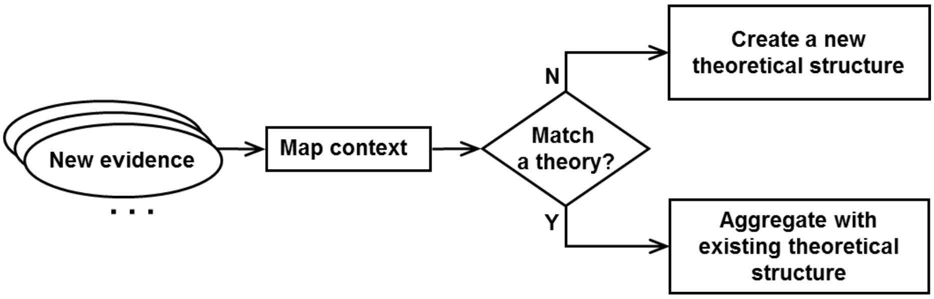

New evidence
. . .
Map context
Match a theory?
N
Y
Create a new theoretical structure
Aggregate with existing theoretical structure

Figure 13 – Simplified view of the structure aggregation steps

To aggregate evidence it is necessary to define what makes a theoretical structure match another, i.e., what makes them compatible. We define two theoretical structures as compatible when their *value concepts* match. To that end, it is necessary first to verify if the investigated object represented by a *cause* concept is the same in both models. If the *cause* is the same, then the other value concepts are checked. The other *value concepts* are *archetypes* and *contextual aspects*. As *archetypes* are the same for all theoretical structures, this means that it is only necessary to compare the *contextual aspects* concepts.

Requiring that evidence have been observed under the same conditions is arguably a methodological consensus in most research synthesis methods to ensure a minimal comparability baseline for analyses involving more than one piece of evidence. For instance, if an evidence results from an investigation in a large scale web system, this should be identified by a value concept describing this aspect in the corresponding theoretical structure. As a result, in order to any other evidence be compatible with it, it is necessary that the second evidence results from a large scale web system observation as well. It is possible to see that this 'match operation' between two theoretical structures is highly dependent from well-defined concepts, so that two pieces of evidence observed in similar contexts can be associated to the same theoretical structure. Although regularly present in SE, taxonomies are crucial in this regard. Nevertheless, taxonomies in SE is not a specific issue in SSM, but rather in any research synthesis method used in SE.

Pragmatically speaking, instead of trying to map an evidence to an existing theoretical structure it is usually more practical to create a theoretical structure for each evidence being aggregated and then identify their compatibility (Figure 14). After evaluating if an evidence is compatible with an existing theoretical structure or if two theoretical structures representing two evidence are compatible, it is time to perform the aggregation itself.

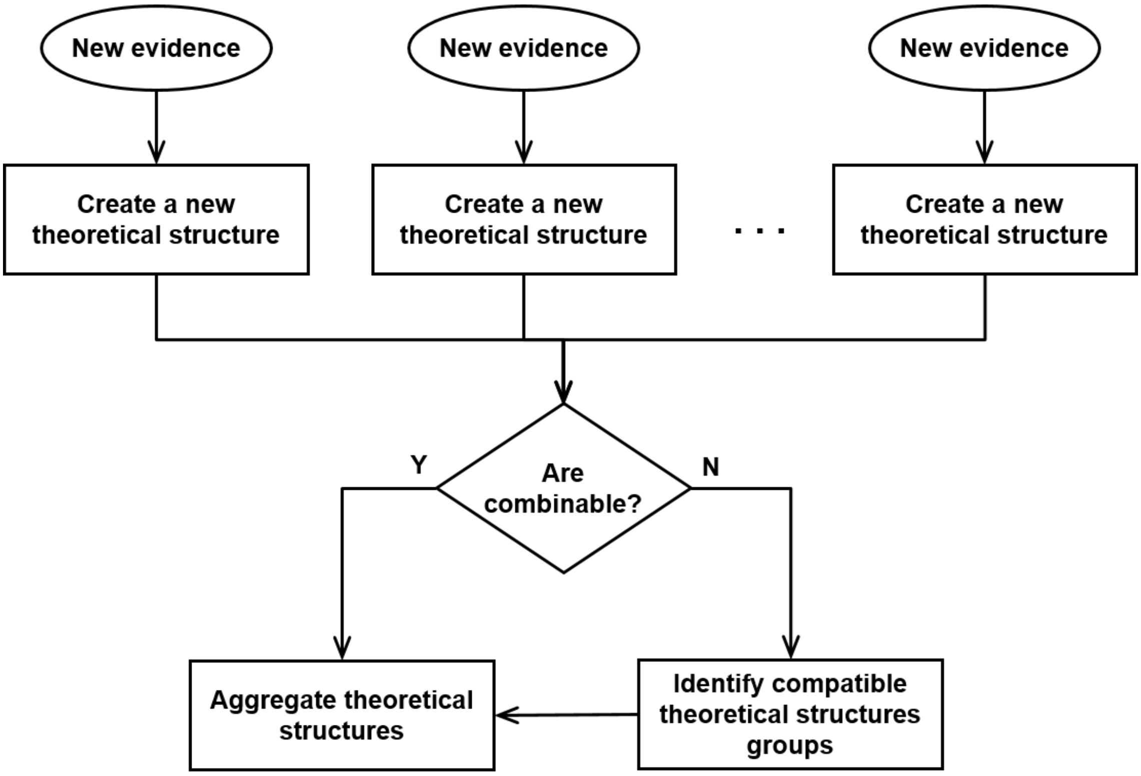

New evidence
New evidence
New evidence
Create a new theoretical structure
Create a new theoretical structure
. . .
Create a new theoretical structure
Are combinable?
Y
N
Aggregate theoretical structures
Identify compatible theoretical structures groups

Figure 14 – Alternative view of the structure aggregation steps (compared to Figure 13)

The aggregation is performed over the *causal* and *moderation* relationships. The resulting aggregated theoretical structure is formed by the shared *value concepts*, in addition to the *variable concepts* that will have their *causal* and *moderation* relationships aggregated. It is important to remember at this point that if only one of the evidence being aggregated has an *effect* or *moderator* it will be added to the resulting model.

Since the scale used in *causal* and *moderation* relationships determine the *frame of discernment*, and given that evidence effect intensity and moderation direction are defined using the *bpa function*, the aggregation is a result of a straightforward application of D-S theory rule of combination. Figure 15 shows the list of all *bpas* for two hypothetical evidence related to source code refactoring represented by the Figure 12 model. As all *bpas* for the first (m_1) and second (m_2) have the same belief value, Figure 15 puts *effects* and *moderators* together. But, in fact, it should be read as $m_{1\text{-experience}}$({IP}) = 0.65, $m_{2\text{-experience}}$({IP}) = 0.4, $m_{1\text{-effort}}$({WN}) = 0.65, and so on.

$$m_1\left(\begin{array}{l}\text{Experience}(\{\text{IP}\})\\ \text{Effort}(\{\text{WN}\})\\ \text{Structure }(\{\text{PO,SP}\})\\ \text{Coding directives}(\{\text{SP}\})\\ \text{Quality}(\{\text{WP}\})\\ \text{Metrics}(\{\text{WP,PO}\})\\ \text{Maintainability}(\{\text{SP}\})\end{array}\right)=0.65+m_2\left(\begin{array}{l}\text{Experience}(\{\text{IP}\})\\ \text{Effort}(\{\text{NE,WN}\})\\ \text{Structure }(\{\text{WP,PO}\})\\ \text{Coding directives}(\{\text{PO}\})\\ \text{Quality}(\{\text{IF,WP}\})\\ \text{Metrics}(\{\text{SP}\})\\ \text{Maintainability}(\{\text{PO}\})\end{array}\right)=0.4$$

Figure 15 – Aggregation of two evidence related to the same theoretical structure

Table 5 shows the combination of two *bpas* from the two hypothetical evidence described above associated with the effect 'structure'. The aggregation between two evidence completes when the rule of combination is applied to all *effects* and *moderators* considered.

Table 5 - Combination of two bpa functions ('structure' effect)

$m_{2\text{-structure}}$ / $m_{1\text{-structure}}$	**{WP,PO} (0.4)**	**Θ (0.6)**
{PO,SP} (0.65)	{PO} (0.26)	{PO,SP} (0.39)
Θ (0.35)	{WP,PO} (0.14)	Θ (0.21)

The resulting combined *bpa* m_3 or $m_1 \oplus m_2$ is given by:

$m_1 \oplus m_2$ ({PO}) = 0.26,
$m_1 \oplus m_2$ ({PO,SP}) = 0.39,
$m_1 \oplus m_2$ ({WP,PO}) = 0.14,
$m_1 \oplus m_2$ (Θ) = 0.21,
$m_1 \oplus m_2$ is 0 for all other sets of Θ.

From the resulting *bpa function* it is possible to compute the *belief function*:

$Bel_{3\text{-structure}}$({PO}) = 0.26,
$Bel_{3\text{- structure}}$({PO,SP}) = 0.39 + 0.26 = 0.65,
$Bel_{3\text{- structure}}$({WP,PO}) = 0.14 + 0.26 = 0.40,
$Bel_{3\text{- structure}}$(Θ) = 1.

As the *belief function* indicates the belief committed to each of the *frame of discernment* elements, which in turn is defined by the *Likert* scale for *effect* intensity and *moderation* direction, the resulting values for *causal* and *moderation* relationships of the aggregated theoretical structure is obtained from it. There is no general rule in D-S theory defining how the most probable hypothesis from the *frame of discernment* is selected from the *belief function* outcome. This depends on the problem being modeled (Bloch, 1996), especially when there are semantics associated with the compound hypotheses (i.e., a subset of Θ with more than one element) as, for instance, is the case of $m_{1\text{-metrics}}$({WP,PO}) = 0.65.

In the problem addressed here, a subset with two hypothesis represents a situation where there is not sufficient precise information to define the relationship value and, as an alternative, it is chosen to define an interval (*e.g.,* between weakly positive and positive). A compound hypothesis, is chosen whenever the contained hypotheses do not contribute with more than or equal to 75% of its belief. Thus, for instance, for a compound hypothesis with size of two elements, there are two contained hypotheses with size of one (*i.e.*, singleton hypotheses), which, following the defined rule, none should add more than 75% to the final belief of the super hypothesis. Otherwise, that contained hypothesis is chosen instead. In the given example above, this happened with the 'structure' effect. The subset {PO, SP} is the compound hypothesis with the highest belief value and its singleton hypotheses do not contribute with more than 75% – {PO} adds 40% of total belief (0.26/0.65 = 0.4) and {SP} adds 0% (0/0.65 = 0). The general idea is that the most specific hypothesis is chosen whenever it has a major impact to the superset hypothesis belief, otherwise we understand it is better to use a less specific superset hypothesis as a result (Figure 16).

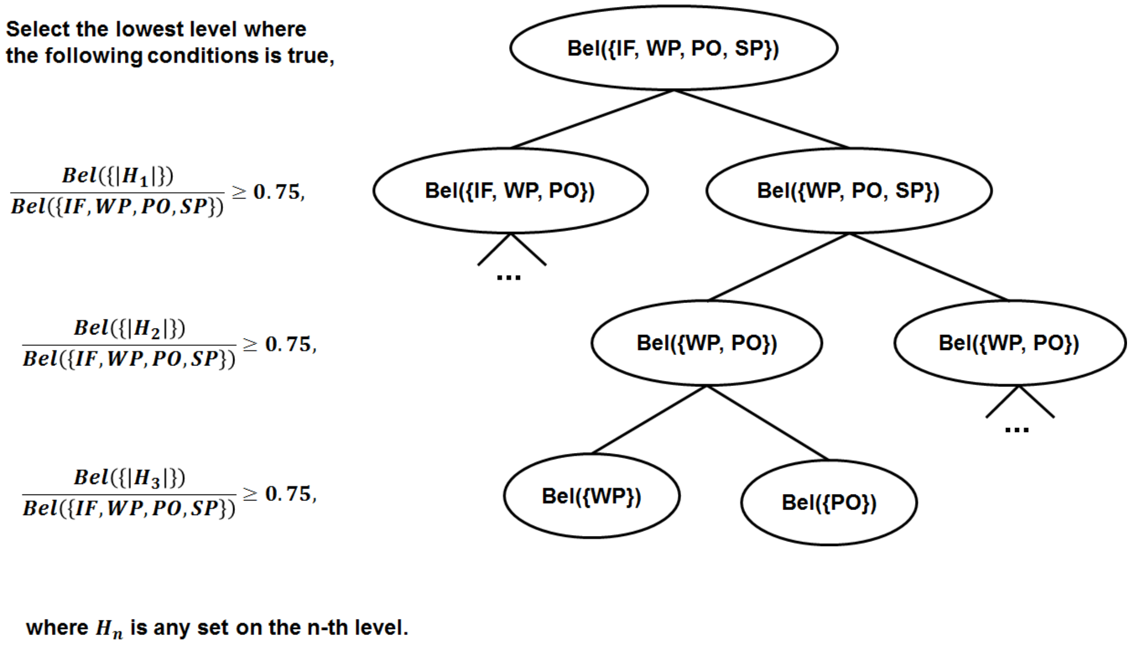

Select the lowest level where
the following conditions is true,
$\frac{Bel(\{|H_1|\})}{Bel(\{IF, WP, PO, SP\})} \geq 0.75,$
$\frac{Bel(\{|H_2|\})}{Bel(\{IF, WP, PO, SP\})} \geq 0.75,$
$\frac{Bel(\{|H_3|\})}{Bel(\{IF, WP, PO, SP\})} \geq 0.75,$
where H_n is any set on the n-th level.
Bel({IF, WP, PO, SP})
Bel({IF, WP, PO})
Bel({WP, PO, SP})
...
Bel({WP, PO})
Bel({WP, PO})
...
Bel({WP})
Bel({PO})

Figure 16 - Hypothesis selection rule procedure based on hypothesis specificity and the respective belief[15]

[15] The compound hypothesis with size of four was chose just as an example, since it does not contains hypotheses (effect intensities) with opposite direction. A compound hypothesis with size of five elements, e.g., {WN, IF, WP, PO, SP}, necessarily contains hypotheses with opposite direction. In this case, {WN} and {WP}. Still, the same procedure applies.

Using this criterion, we can determine the final values for the *causal* and moderation relationships of the aggregated theoretical structure:

{IP} for experience, since $Bel_{3\text{-experience}}$({IP}) = 0.79,
{WN} for effort, since $Bel_{3\text{-effort}}$({WN}) = 0.65,
{PO, FP} for structure, since $Bel_{3\text{-structure}}$({PO,SP}) = 0.65,
{SP} for cod. directives, since $Bel_{3\text{-cod.directives}}$({SP}) = 0.53,
{WP} for quality, since $Bel_{3\text{-quality}}$({WP}) = 0.65,
{WP, PO} for metrics, since $Bel_{3\text{-metrics}}$({WP,PO}) = 0.72,
{SP} for maintainability, since $Bel_{3\text{-maintainability}}$({SP}) = 0.53.

These aggregation results represent two aspects of relationships. One is the probable value considering the defined scale and the other is the confidence about it. The confidence level or, in D-S theory terms, the belief values derived from the combination should be interpreted along with other parameters of the aggregation. In general, it can be said that it reflects two aspects of the aggregated evidence: (i) the level of agreement between them and (ii) their strength. Thus, when a high belief value is obtained from an aggregation it could be a result of, for instance, a large number of weak evidence as well as a small number of strong evidence. On the other hand, a low belief value can be, for instance, an outcome of a small number of compatible weak or conflicting strong evidence. In any way, what is an interesting in applying D-S theory to evidence aggregation is that results not only indicate a trend (*e.g.*, positive or negative), but also provide a numeric parameter to interpret how reliable are the findings.

If a third evidence is considered for aggregation, it is important to highlight that the whole *bpa function* associated with the aggregated pair is taken for the next aggregation. This means that all *m-values* resulting from the previous aggregated pair is taken for the next aggregation, not only the belief value and the hypothesis indicated by the *belief function*. Continuing from the previous example, let's say that the third evidence has been estimated with a 0.9 probability committed to strongly positive for source code structure – $m_{4\text{-structure}}$({SP}) = 0.9. In Table 6, we show the rule of combination application for the third evidence. Taking these results and computing the *belief function* the new value for 'structure' effect is {SP}, since $Bel_{4\text{-structure}}$({SP}) = 0.84.

Table 6 – Combination of a third evidence for effect 'structure'

$m_{4\text{-structure}}$ $m_{3\text{-structure}}$	**{SP} (0.9)**	**Θ (0.1)**
{PO} (0.26)	Ø (0.23)	{PO} (0.03)
{PO,SP} (0.39)	{SP} (0.35)	{PO,SP} (0.04)
{WP,PO} (0.14)	Ø (0.13)	{WP,PO} (0.01)
Θ (0.21)	{SP} (0.19)	Θ (0.02)

The values for the combined *bpa function* m_5 or $m_3 \oplus m_4$ are:

κ = 0.36 and 1 - κ = 0.64,
$m_3 \oplus m_4$ ({PO}) = 0.03/0.64 = 0.05,
$m_3 \oplus m_4$ ({SP}) = (0.35+0.19)/0.64 = 0.84,
$m_3 \oplus m_4$ ({PO,SP}) = 0.04/0.64 = 0.06,
$m_3 \oplus m_4$ ({WP,PO}) = 0.01/0.64 = 0.02,
$m_3 \oplus m_4$ (Θ) = 0.02/0.64 = 0.03,
$m_3 \oplus m_4$ is 0 for all other sets of Θ.

It can be seen that the evidence aggregation procedure allows the body of knowledge to be progressively expanded according to new investigation results published. The order in which evidence are combined do not affect the final result (Shafer, 1976). Therefore, in long term, the aggregation procedure can be performed incrementally. However, it must be observed that decisions about evidence compatibility will affect further aggregations. Consequently, the researcher should always pay attention to the possibility of considering to re-evaluate previous aggregation decisions or even start the aggregation from scratch.

One major issue that can be faced when aggregating evidence is the occurrence of conflicting evidence, as was the case of the third evidence aggregated which resulted in a 0.36 conflict level. Although D-S theory has its own mechanism to address this situation, we enumerate three actions that can be taken based on Ciolkowski (2009):

- **Ignore:** this is the simplest case and was adopted in the above example. Conflicted is redistributed among the frame of discernment hypotheses. This is the standard way to handle conflicts in D-S theory (Shafer, 1976);
- **Explain:** the researcher tries to analytically describe why the conflict was found. This is usually performed by identifying a moderator variable which is capable of explaining a significant part of the variability among evidence results (Ciolkowski, 2009). We suggest to use this approach only when there is a conflict superior to 0.5[16] considering all evidence combinations, or when the

[16] Conflict resolution is an active area in D-S theory (Haenni, 2005). There is not a consensual threshold for conflict value that can be used to determine when a combination is invalid or is not reliable. This seems to be defined in case-to-case basis considering the problem at hand. In our case, 0.5 is used because it indicates there is more belief in ‘conflict’ than in the hypotheses.

mean conflict[17] is higher than $1/n$ where n is the number of evidence being aggregated. Other conflict levels can be just ignored;

- **Incorporate:** as an alternative to explaining, the goal in this case is to incorporate the variability into the aggregated results by reducing its precision. It is interesting to see that this procedure does not exist in D-S theory, but it is suggested here to the cases where there is not a direction conflict (e.g., between {NE} and {WP}), but only a strength difference (e.g., between {PO} and {SP}). We indicate this as an alternative approach to *explain* the differences. Incorporating conflict is performed in the following manner. If there is a conflict between two singleton hypotheses, for instance, {PO} and {SP}, the conflict is incorporated into the compound hypothesis containing these elements (in this case, {PO, SP}). Example: m_1({PO}) = 0.9 and m_2({SP}) = 0.8. Conflict: 0.72. The resulting aggregation become m_3({PO, SP}) = 0.72 instead of redistributing conflict to m_3(A)/0.72 to all A where m_3(A) > 0. In addition, singleton hypotheses (in the same direction) can be up to two units away in the *Likert* scale ({WP} and {SP}). In this case, the same procedure can be performed, but the conflict is incorporated to the {WP, PO, SP} hypotheses. As compound hypotheses can also conflict, the same rule also applies. For instance, when {WP, PO} conflicts with {SP} we can incorporate the conflict into the {WP, PO, SP} hypothesis.

Aggregation of comparative theoretical structures

To conclude the explanation about the evidence aggregation procedure, we need to describe additional concerns involved with comparative evidence. The only important difference between descriptive and comparative theoretical structures is the way that *causal* relationships are described. In comparative theoretical structures, *causal* relationships are defined relatively to the two causes (*e.g.*, technologies) observed in an evidence. Given this difference, it is necessary to define an analog scale describing the comparison. A seven-point *Likert* scale with the following values is defined: strongly inferior (SI), inferior (IN), weakly inferior (WI), indifferent (IF), weakly superior (WS), superior (SU), and strongly superior (SS). In addition, we redefine the *frame of discernment*: Θ = {SI, IN, WI, IF, WS, SU, SS}.

[17] The mean conflict is given by $\frac{\sum_{c=1}^{n-1} K_c}{n-1}$, where n-1 is the number of evidence combinations and K_c is the conflict value for a given combination.

In fact, besides the effects' scale distinction, all the described procedures for aggregation are still applicable in both cases. However, there are situations where it is necessary to aggregate both descriptive and comparative evidence. Descriptive evidence is characterized by its focus on describing possible benefits and drawbacks of a single cause whereas comparative evidence try to do that relatively to another cause under the same category (*e.g.*, two inspection techniques). Despite this difference, we understand that aggregation is still possible since we can find the notion of causality in both kinds of evidence. Following the research questions line of argument to justify this claim, Easterbrook *et al.* (2008) distinguish between 'causality' and 'causality-comparative' questions.

We define two additional procedures to aggregate mixed kinds of evidence: (i) determining a comparative theoretical structure based on the comparison of two descriptive theoretical structures and, the reverse operation, (ii) dismembering a comparative theoretical structure into two descriptive ones. In both cases, the understanding of compatible theoretical structure is maintained: it is only possible to compare theoretical structures having the same *value* and *variable concepts*, and dismembering a comparative theoretical structure produce two compatible descriptive theoretical structures with the same *value* and *variable concepts*.

The comparison of two descriptive theoretical structures is performed in the following manner. Using the defined *Likert* scale as an approximation for an interval scale, the maximum distance between the seven-point scale extremes (i.e., {SN} and {SP}) is 6 and the minimum is 0 when the compared values are the same. Based on this, we define a conversion rule between the descriptive and comparative scales:

- **{SI} or {SS}** when the difference is equal or larger than 3 units (e.g., from {IF} and {SP} = 3 up to {SN} and {SP} = 6);
- **{IN} or {SU}** when the difference is equal to 2 units;
- **{WI} or {WS}** when the difference is equal to 1 unit;
- **{IF}** when there is no difference.

As a convention, we say 'superior' when the first compared cause is better than the second and 'inferior' otherwise. For instance, if there is one descriptive evidence for two inspection techniques with $m_{1\text{-t1-\#defects}}(\{PO\}) = 0.3$ and $m_{1\text{-t2-\#defects}}(\{IF\}) = 0.9$ then, as the difference between {PO} and {IF} is equal to 2 units, the conversion would result in $m_{1\text{-t1/t2-\#defects}}$({SU}) = 0.3[18]. Another relevant aspect about the conversion rule is the definition of the belief value. As the comparative scale is defined from two evidence we take the minimum belief value between the pair. Thus, in the above example we have min(0.3, 0.9) = 0.3.

The dismembering procedure, on the other hand, should only be considered when raw descriptive data is not available, but comparative data such as numeric difference or qualitative description about differences. Otherwise, even if the study reports a comparative evidence, whenever the raw descriptive data is available it should be used to model descriptive theoretical structures. Therefore, the dismembering procedure only provides an approximate way to estimate the 'individual' effects for each cause considered in the comparison. To that end, we have defined an 'inverse' conversion rule based on the comparison procedure. Dismembering precision depends on the effects' difference intensity and direction (positive or negative).

For instance, in the best case, when compared causes are both non-negative and the first is strongly superior ({SS}) to the second, then the only possibility to meet these considerations when dismembering is that the first cause is strongly positive ({SP}) and the second indifferent ({IF}). This is because, by definition, strongly superior must have 3 units of difference and given that causes do not have distinct direction we conclude that one is {SP} and the other is {IF}. Following this reasoning, the worst case for dismembering is a situation where both causes are non-negative (or non-positive), but do not have difference ({IF}). Given these two considerations, there are four possible equally acceptable answers for dismembering, since both descriptive causes could assume values {IF}, {WP}, {PO}, and {SP}. In Table 7, we enumerate all possible combinations[19] for dismembering comparative theoretical structures.

[18] Clarification about the notation used: 1-t2-#defects should be read as 'evidence 1 related to the technique t2 for the effect #defects and 1-t1/t2-#defects as 'evidence 1 comparing t1 and t2 in relation to #defects'.

[19] Non-negative and non-positive cases can be converted to negative and positive cases by just removing {IF} from dismembered effects values. In this case, as the maximum difference between two descriptive values is 2 units (e.g., between {WP} and {SP}), the comparative values can not assume {SI} or {SS}.

Table 7 – Conversion rules from comparative to descriptive theoretical structures

	Comparison between cause 1 and 2	Comparative causes	Dismembered value for cause 1	Dismembered value for cause 2
Non-negative effects	1 > 2	{SS}	{SP}	{IF}
		{SU}	{PO,SP}	{IF,WP}
		{WS}	{WP,PO,SP}	{IF,WP,PO}
	1 = 2	{IF}	{IF,WP,PO,SP}	{IF,WP,PO,SP}
	1 < 2	{WI}	{IF,WP,PO}	{WP,PO,SP}
		{IN}	{IF,WP}	{PO,SP}
		{SI}	{IF}	{SP}
Non-positive effects	1 > 2	{SS}	{IF}	{SN}
		{SU}	{WN,IF}	{SN,NE}
		{WS}	{NE,WN,IF}	{SN,NE,WN}
	1 = 2	{IF}	{SN,NE,WN,IF}	{SN,NE,WN,IF}
	1 < 2	{WI}	{SN,NE,WN}	{NE,WN,IF}
		{IN}	{SN,NE}	{WN,IF}
		{SI}	{SN}	{IF}
One effect non-negative and the other non-positive	1 > 2	{SS}	{IF,WP,PO,SP}	{SN,NE,WN,IF}
		{SU}	{IF,WP,PO}	{NE,WN,IF}
		{WS}	{IF,WP}	{WN,IF}
	1 = 2	{IF}	{IF}	{IF}
	1 < 2	{WI}	{WN,IF}	{IF,WP}
		{IN}	{NE,WN,IF}	{IF,WP,PO}
		{SI}	{SN,NE,WN,IF}	{IF,WP,PO,SP}
Note: when the comparative value is an interval, we assume the worst case in the table (more imprecise)		{WS, SU} (non-negative effects)	{LP,PO,FP}	{IF,WP,PO}

As shown in Table 7, we only have defined conversion rules when effects have different directions. Conversion rules in other situations, although possible, would not have any validity or would be too imprecise. Thus, dismembering is only possible when there is some indication about the effects direction. This occurs, for instance, when raw data about the comparison is not available, but there are charts such as boxplots or dispersion showing the direction. Or, in qualitative cases, where authors report that both causes had positive (or negative) effects, but one was superior to another.

5.4 SSM

SSM is a research synthesis method that addresses concerns associated with *knowledge translation*. Therefore, as a process, SSM combines elements required to that goal[20]: (i) same orientations from systematic literature review, (ii) definitions and procedures for evidence aggregation and (iii) guidelines and heuristics for knowledge modeling using the diagrammatic representation for evidence. It is also important to remind that these aspects are based on the hypothesis that a shared and unified view about evidence representation can contribute to the process of knowledge translation within the evidence-based practice in SE (Figure 1).

We define five phases for SSM based on Ciolkowski (2009):

i. **Planning and definition:** the study objectives are defined, including the research question and the inclusion/exclusion criteria formalized. In some situations, a theoretical structure can be modeled to serve as basis to identify what must be present in selected papers in a similar manner as extraction forms used in systematic reviews.

ii. **Selection:** primary studies are collected by following a systematic procedure considering the defined criteria. To help managing and organizing papers, basic information about the studies can be extracted, such as bibliographic data and important aspects as research goal or study type.

iii. **Quality assessment:** the quality of primary studies is evaluated using quality assessment checklists proposed for SE (Shull *et al.*, 2011, Kitchenham *et al.*, 2012). The quality assessment is used as an input to estimate the confidence (*i.e.*, belief values) on the study results' causal and moderation relationships.

iv. **Extraction and translation:** evidence are extracted from studies and translated to theoretical structures. Knowledge extraction is performed in terms of concepts and relationships identification, following the restrictions of the adopted diagrammatic model (Section 5.3.1).

v. **Aggregation and analysis:** based on the extracted theoretical structures compatible evidence is aggregated by pooling their effects and moderators (Section 5.3.2). Then, the results are analyzed together.

Before detailing the SSM process, we try to indicate where the existing research synthesis methods were used in SSM definition. In

[20] We mention this drawing on a parallel with the three key elements which form Knowledge Translation as enumerated in Chapter 1: 'the outcomes of a systematic literature review; interpretations of what these mean in particular contexts; and appropriate forms for communicating them'.

Table 8, we map the research synthesis methods reviewed in the previous section to the SSM process phases, explaining what parts of the methods were incorporated into SSM process. It is interesting to observe that SSM process definition was not linear since consecutive refinements were made throughout its development, but it has essentially consisted in adapting the five phases from Ciolkowski (2009) by incorporating features from existing research methods aligned with the goals defined for SSM. The final results of this rationale are detailed in the following subsections after

Table 8.

Table 8 – Existing research synthesis methods' aspects inspired for or incorporated into SSM

<table>
<tr><th>Method
SSM Phase</th><th>Thematic Synthesis</th><th>Meta-Ethnography</th><th>Case survey</th><th>Qualitative Comparative Analysis</th><th>Theory building and meta-analysis</th></tr>
<tr><td><b>Planning and definition</b></td><td colspan="5" rowspan="2">• In general terms, these research synthesis methods do not have any particular concern about research definition and studies selection, with few minor considerations:
○ Discussions involving meta-ethnography and qualitative comparative analysis point to the importance of prioritizing fewer studies than large generalizations when retaining studies' context is important.
○ Case survey highlights the importance of having explicit exclusion criteria for studies.
○ Theory building and meta-analysis indicate steps used in systematic literature review – which is also used in studies using the other methods, although not mentioned as a general guideline.</td></tr>
<tr><td><b>Selection</b></td></tr>
<tr><td><b>Quality assessment</b></td><td>—</td><td>—</td><td>• Level of agreement between researchers is used as a parameter for results reliability</td><td>—</td><td>• '(…) meta-analysis attempts to record various aspects of research methodologies for the existing studies to identify their relationship to study findings'</td></tr>
</table>

Method SSM Phase	Thematic Synthesis	Meta-Ethnography	Case survey	Qualitative Comparative Analysis	Theory building and meta-analysis
Extraction and translation	• Increasing level of abstraction (text coding) is useful for concepts and relationships identification	• The method suggests to use tables to enumerate most part of the data used for synthesis (context, concepts, and relationships) and the synthesis itself, which contributes to the method transparency and improves the evidence comparability	• Its notion of confidence associated with survey answers, which represent evidence findings, is accommodated by D-S theory	• Logical description of (absent and present) conditions associated with dependent and independent variables is similar to the theoretical structures' value and variable concepts along with their possible relationships • Data tabulation is suggested here to be able to describe all combinations of conditions and is similar to the role of theoretical structures in improving evidence comparability	• When modeling evidence with theoretical structures, it is important to be aware about the differentiation between constructs, concepts, and variables since it can support researchers identify important information in papers. Still, from the perspective of the representation itself, i.e., after modeling evidence, this differentiation is secondary for understanding the evidence model
Aggregation and analysis	• Usage of diagrammatic representation indicate its applicability as a tool for research synthesis	• Translation procedure is useful to homogenize (translate) evidence models into one another	—	• The comprehensive approach can be adapted to use an existing theoretical structure to decide which evidence and their respective theoretical structures can be combined • The inductive approach is similar to the approach described in Section 5.3.2, where all theoretical structures are analyzed and refined together	—

5.4.1 Step 1: planning and definition

SSM study planning and definition virtually share most of the concerns with any secondary study. It must define the research questions, search strings, primary study sources, and inclusion and exclusion criteria. Considering the theoretical structure knowledge elements (*i.e.,* concept and relationship types), we formulate a basic format for research questions using SSM:

1. *What are the expected effects when Technology T is applied in context C (Activity, Actor, and System)?*[21]
    a. *How Technologies T1 and T2 compare when found in context C (Activity, Actor, and System)?*

This format supports research questions as specific as the topic under investigation requires. If it is a more specific question, context can be specified with more details by specifying concrete concepts. It is even possible to focus on a specific causal relationship: 'what is the impact of Technology T over the feature F of the System S?' Otherwise, context can be specified with more abstract concepts or even by leaving out some of the archetypes (Activity, Actor, and System).

Regardless of its specificity, we understand that research questions using this format should not be focused in particular cause-effect pairs, but rather make synthesis scope wide enough so that multiple variables and their inter-relationships can be analyzed according to what is found in the technical literature. This is important so the researcher can exploit the theoretical structure expressivity in terms of multiple effects and contextual description. Nevertheless, if the researcher choose to investigate what is there in the technical literature for a specific set of possible effects, then we suggest to use a theoretical structure as a research question instead. The idea is that a theoretical structure is modeled based on the researcher background or an initial set of papers, and it is used to as the inclusion and exclusion criteria. Besides, it can also define the 'compatibility criterion' for evidence aggregation in Step 5.

Based on the research question, a search string must be defined to find articles that could be used to answer it after synthesizing all evidence. In systematic reviews, the PICO (Population, Intervention, Comparison, and Outcome) format is commonly

[21] Any other combination of causal element and context can be arranged in the research question. The technology as causal element in this case is just for illustrative purposes, although it is also expected to be the most frequent format.

applied (Pai *et al.*, 2004). Analogously, as the aggregation is performed using theoretical structures, we proposed the ATAS (Actor, Technology, Activity, and System) format when defining search strings in SSM. The order of archetypes reflect the theoretical structure purpose of describing 'an actor applying a technology to perform activities in a software system'. Alternatively, it is possible not define a search string whenever snowballing or other approach to search for primary studies is used.

Studies selection should follow a set of inclusion and exclusion criteria defined in advance. Systematic review has several guidelines in this regard (Biolchini *et al.*, 2005). One particular concern in a research synthesis is to only consider empirical studies for inclusion. Expert opinion, theoretical/analytical, or just speculative papers are usually not aligned with the purposes of this kind of secondary study. Any other kind of primary study can be included in SSM studies, even the ones which are associated with a weaker form of evidence such as experience reports and case studies. SSM aggregation procedure is able to represent evidence with different strength levels using a belief value corresponding to that strength which, then, can be combined using D-S theory to analyze what the evidence, together, indicate.

Another specific aspect for planning and defining SSM studies is a particular attention to the use of scales. As cause-effect relationships use a semi-quantitative *Likert* scale, it is recommended to define how quantitative variables will be converted to that scale. This improve the synthesis transparency and, consequently, its reliability. If known conversion rules are available in technical literature, its usage is recommended – for instance, Foreman *et. al* (1997) define quantitative ranges to qualify different levels of ciclomatic complexity. Otherwise, researchers can define conversion rules based on their own experience. As it may be concluded, it is not possible to know all variables that can be found in technical literature beforehand. Therefore, the conversion rules can be revised during the process. In addition, conversion rules based on absolute values (e.g., number of defects) should be avoided as they are highly dependent of a specific context (10 defects in a large system can be considered a low number, whereas it probably won't be in a smaller system). Thus, instead of using absolute values, it is preferable to use normalized values (e.g., defect density per thousand lines of source code) as they are easier to have a consensual understanding about what represent a high or a low value. Qualitative values, on the other hand, do not need explicit conversion rules as they are much more imprecise and dependent of adverbs and adjectives used by authors.

5.4.2 Step 2: selection

Primary studies selection also follows general systematic review guidelines. In this step, papers are searched and selected based on title, abstract or on their full text. When selecting technical articles, researchers must follow the criteria for inclusion and exclusion, justifying their choice if necessary. More than one researcher should be involved in this process, since interpretation of selection criteria can be subjective. All these elements, together, will improve the chances of identifying primary studies aligned with the research synthesis goals and questions.

For papers included, selection step can also be used to register basic bibliographic information which are useful to keep the link between theoretical structures and their source. It is also important to identify and record the research method used, as it is useful for the next step of SSM.

5.4.3 Step 3: quality assessment

The quality assessment step is crucial in SSM since it is used as an input to estimate the confidence level on the causal and moderation relationships identified in the respective study. SSM uses both the *quality* of primary studies as well as the *strength* of evidence originated from them (Dybå and Dingsøyr, 2008a). Quality is related to the extension to which the primary study planning, execution and analysis is effective in avoiding systematic errors or bias. Strength, on the other hand, is usually associated with a hierarchy of primary study types which assumes that more controlled studies has stronger results than less systematic studies.

Based on these two concepts, SSM defines belief values in the following manner. Using evidence classification from GRADE (Atkins *et al.*, 2004), four study types is used to split the 0-1 belief value range into four subranges: unsystematic observations [0.00, 0.25]; observational studies [0.25, 0.50]; *quasi*-experiments [0.50, 0.75]; and randomized controlled trials [0.75, 1.00]. Then, quality assessment is performed based either on Shull *et al.* (2006) or Kitchenham *et al.* (2012) scoring schemas which is converted into this 0.25 subrange (Table 9). A last adjustment to belief values can be performed using the D-S theory discount operation. The idea of the discount operation is to adjust the mass distribution (i.e., the belief values assigned to the hypotheses) to reflect the source's credibility – a full discount (discount=1) represents a completely unreliable source. This can be used in any situation in which a criteria for discounting can be defined. For instance, it is possible to define a criteria for discounting results that were subjected to statistical hypothesis testing. In this cases, discount value can be defined as the *p-value*. Thus, if *p-value* is low, let us say 0.03, then the discount could be linearly as low as 0.03.

Table 9 – Reference belief value levels based on evidence type categories

Strength	**Evidence strength description (Atkins *et al.*, 2004)**	**Study type**	**Quality assessment checklist**	**Belief value ranges**
High	'Further research is very unlikely to change our confidence in the estimate of effect'	Randomized trial	Kitchenham *et al.*, (2012)	[0.75, 1.0]
Moderate	'Further research is likely to have an important impact on our confidence in the estimate of effect and may change the estimate'	*Quasi*-experiment	Kitchenham *et al.*, (2012)	[0.50, 0.75]
Low	'Further research is very likely to have an important impact on our confidence in the estimate of effect and is likely to change the estimate'	Observational study	Kitchenham *et al.*, (2012)	[0.25, 0.50]
Very low	'Any estimate of effect is very uncertain'	Non-systematic study	Shull *et al.* (2006).	[0.0, 0.25]

Although moderated strength is defined in Atkins *et al.* (2004), *quasi*-experiments are not considered among evidence types. In Atkins *et al.* (2004), additional criteria are defined to increase or decrease evidence grade, so they can assume the moderate value. SSM adds *quasi*-experiments to the repertoire of evidence types, since it is an important study design used in SE. In addition, SSM defines strict separation between belief values ranges. We understand that this strict separation can cause some distortions, as some researchers may consider that some high quality observational study can produce stronger evidence than low quality *quasi*-experiment, but we found that the criteria defined in Atkins *et al.* (2004) for moving an evidence type to a higher or lower grade requires an maturity level in designing studies that has not been yet achieved in or defined for SE. For instance, one of the criteria to increase grade is 'evidence of dose response gradient', which is an approach seldom used or even possible in SE. To decrease grade, one criterion is the presence of 'imprecise or sparse data'. According to Atkins *et al.* (2004) data are sparse 'if the results include just a few events or observations' or 'if the confidence intervals are sufficiently wide that an estimate is consistent with either important harms or important benefits'. Given characteristics of SE phenomena, such criterion can be found in most SE quantitative studies – we cite two meta-analysis with several primary studies in this situation

Hannay *et al.* (2009) and Rafique and Misc (2013). Hence, this criterion would not represent a real difference between studies in our area as most would have their grade decreased if, for instance, this criterion is taken into account. Thus, we understand that experimental SE community should develop its own set of criteria in that regard.

Following this line of argument, instead of adopting the criteria defined in Atkins *et al.* (2004), we use two checklists as scoring schemas. One for experimental studies (ranging from observational studies to randomized trials) and another one for unsystematic studies. The scoring schema for the former was taken from the proposal of Kitchenham *et al.*, (2010) with one additional question about context description in primary studies identified in Dybå and Dingsøyr (2008b), totaling 10 questions (see Appendix A.1). It covers many common aspects related to any experimental research methodology such as research goals, study instruments and validity threads. As quality assessment contributes at most with 0.25, each of the 10 questions is rated with 0.025 – all have the same weight. In addition, checklist questions have some considerations called 'things to consider' in Dybå and Dingsøyr (2008b), which depending on how they are evaluated can make the question rate vary between 0.000 and 0.025. Appendix A.1 gives more details about how rates are computed.

The other checklist, proposed by Shull *et al.* (2006), is indicated to evidence originated in investigations where no research methodology is applied. It is formed by four questions with different weights. As in the other checklist, the 0.25 rate is distributed among questions, but in this case respecting the different weights. Although there is much less aspects to evaluate the quality of non-systematic investigations, and hence only four questions are considered, the checklist offers a minimum support to have some perception about how well an unsystematic observation was planned and performed. The Appendix A.2 contains the checklist with its questions and the respective weights for each question.

A possible risk involved in using these kind of checklists is the reliability of the responses. To that end, it is important to consider to participation of more than one researcher when assessing the quality of the primary studies. In fact, in Kitchenham *et al.* (2012) it was observed a high level of agreement with only two researchers – given that they are free to discuss about divergent perspectives and can reach a consensus.

5.4.4 Step 4: extraction and translation

Information extraction should be thought from the beginning on the needs of its translation and modeling to theoretical structures. The general methodological basis guiding this process can be thought out as text coding which is used to break down the narrative argumentation present in scientific papers into a set of text snippets and codes. In summary, coding process involves labeling of data segments starting from raw text to a more condensed form put together according to the research goals (Auerbach and Silverstein, 2003). Text coding allows data organization and manipulation by grouping and comparing data sharing similar characteristics (Charmaz, 2006). It is usually freely and incrementally performed in small steps (Auerbach and Silverstein, 2003): raw text, relevant text, repeating ideas, themes, theoretical constructs and research interests.

The translation and extraction main goal is to identify concepts and relationships to model a theoretical structure. However, contrasting from the classical form of coding, SSM does not follow the idea of open extraction and labeling of text snippets to reveal main themes or lines of argument at end. On the contrary, it aims its attention at some specific aspects, particularly in identifying the contextual factors (*i.e.*, theoretical structures value concepts) that are important to describe the identified effects (*i.e.*, theoretical structures variable concepts), and how these elements are related (*i.e.*, theoretical structures relationships). This is illustrated in Figure 17.

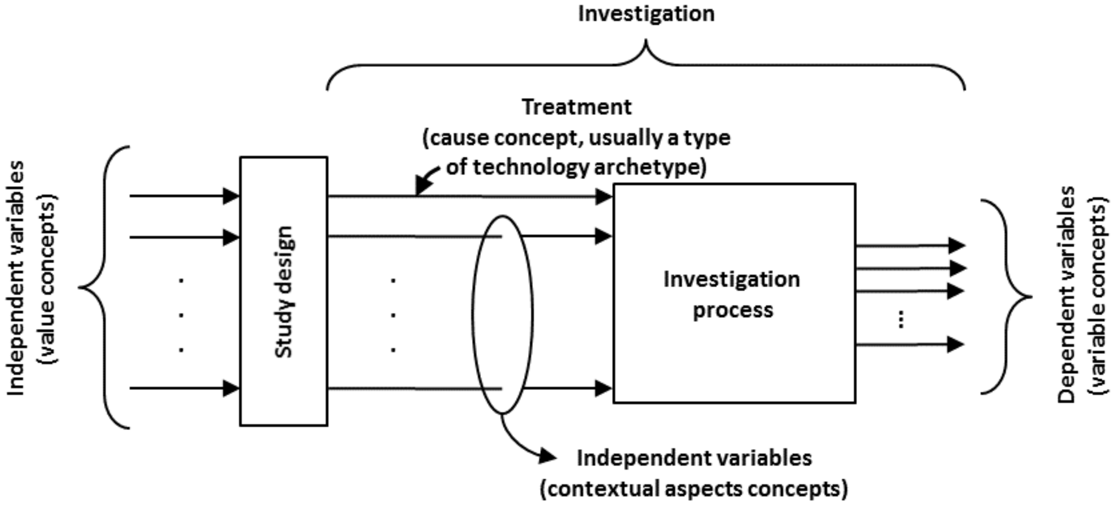


Figure 17 – Common variables in investigations and theoretical structure concept types

Thus, the coding process in SSM does not necessarily need to go through a continuous and iterative process of small steps as it is usually indicated, but it can be focused on the knowledge elements of theoretical structures (Figure 18). Three of them are important in SSM: (i) conceptual codes which identify key concepts related to the object of study or essential dimensions of these concepts, (ii) contextual codes which identify concepts associated with the study context including methodological aspects, and (iii) relational codes which identify and describe the links between these concepts.

These aspects should not be addressed individually, one per turn, but they must be considered in the coding process.

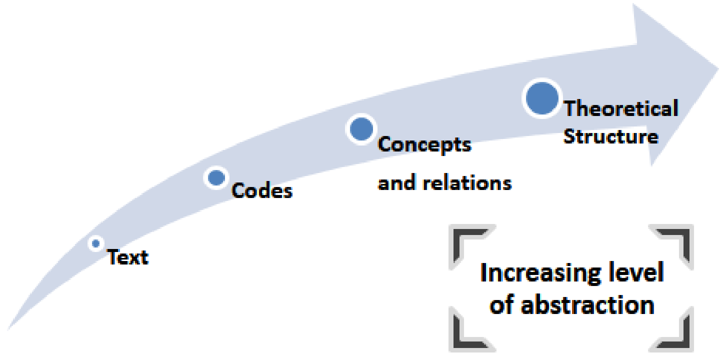


Figure 18 – SSM typical abstraction levels (adapted from Cruzes and Dybå (2011a))

Another important component of the coding process is the translation procedure (Britten *et al*., 2002). In SSM, as the goal is to aggregate evidence by combining the respective compatible theoretical structures, the translation procedure can support the identification of concepts that can translate studies' variables or notions, which at first glance are not comparable, but when translated to the proper concept they become comparable. One example in software context would be translating Understandability and Learnability by a more generic concept, for instance, Usability[22]. Clearly, these kind of generalizations are not free from threads and should be considered in case by case basis according to the researchers' interpretation. Still, these generalizations are not the only possible outcome from translations. In fact, the translation procedure enactment by itself can result in the creation of new concepts that distinguish one study from another or justify their differences. For instance, in the case of two papers reporting studies involving the same type of system, but in one study the system was developed specifically for the study, this could be used to differentiate them. One way to do that is including in the theoretical structures representing each evidence a value concept characterizing the size or the complexity of the system. It is interesting to notice that the translation procedure will usually be used in consecutive iterations between steps 4 and 5, supporting both modeling of primary studies with theoretical structures (step 4) and analysis of compatible evidence for aggregation (step 5).

[22] Understandability, Learnability and Usability terms were taken from the software quality terminology presented in Kitchenham and Pfleeger (1996). The possibility of generalizing Understandability and Learnability by Usability was also taken from there.

During the coding process, researchers must also be aware about the different forms that the concepts manifest. One notion that can help in this regard is the distinction of theoretical, empirical, and measurement research domains, which were already mentioned in this text (Section 2.3.2) and are described in Yang (2002). This distinction can be particularly interesting for the translation procedure, since generalizations, specialization, equivalence and differentiation of concepts are likely to manifest in the intersections between and links of these domains. Moreover, it may improve the transparency of SSM since decisions about these conceptual 'operations' are more objectively characterized. Nevertheless, it is important to highlight, as previously mentioned, that a theoretical structure can contain concepts from different domains, since these different domains are not essential to understand the evidence model.

Apart from strategies for identifying and defining concepts when modeling a theoretical structure for an evidence, we must complement them with strategies to identify and define relationships as well. Three approaches are based on the strategies suggested in qualitative comparative analysis (Yamasaki and Rihoux, 2009): comprehensive, statistical significance, and inductive. In comprehensive approach the researcher defines a set of relationships that are expected to be found in papers before extraction, so that the extraction step becomes a matter of checking if the evidence contain information about them. The 'comprehensive' name is also associated with the fact that SSM does not require causal and moderation relationships to be present in all aggregated evidence – it is enough just one evidence with a specific cause-effect or moderator in order to it be present in the aggregated theoretical structure with its respective intensity and belief value. In the statistical significance approach, the idea is to include only causal and moderations relationships in the theoretical structures representing evidence whether there is a statistical test associated with the observed effect or moderator that reached statistical significance. The inductive approach suggests that relationships extracted from one study can influence extraction in subsequent studies. Induction is associated with the research learning process about the primary studies and the object of study related to the research question. Thus, a study that describes concepts, variables and their interaction more clearly increase the awareness about them, affecting how the researcher will read and extract the remaining papers, or even make he/she revisits the previously extracted ones. Again, this shows the iterative nature of this step in SSM.

The last procedure involved in step 4 regards how the effect direction/intensity and moderation direction should be determined when extracting information from primary studies. Direction is relatively straightforward to interpret from either quantitative or qualitative papers and both for causal and moderation relationships. One important aspect, however, is that positive or negative effect direction is not necessarily associated with an increase or decrease of effect variable, but rather represent if an improvement or worsening has been observed in that variable. Therefore, if, for instance, source code complexity is an effect associated with ciclomatic complexity, a decrease in its value represents an improvement of source code complexity (*i.e.*, it became less complex).

Regarding intensity, on the other hand, the main issue is how values found in primary studies can be translated to the seven-point *Likert* scale. Qualitative values, as previously mentioned, do not need explicit conversion rules as they are usually subject to ambiguity given the usage of adverbs and adjectives by authors. However, for quantitative values the conversion must follow explicit conversion rules respecting data availability. Three typical situations are described here:

- **Precise data:** all needed data to assess the effect size are presented in enough details. This includes raw tabular data, charts with readable scale, mean and standard deviation values, among other formats. In these cases, it is indicated to define conversion rules to precise values of the seven-point *Likert* scale {SI, IN, WI, IF, WS, SU, SS}. In D-S theory terminology, this means a conversion to singleton hypothesis (*e.g.*, {WP});
- **Imprecise data:** incomplete or imprecise data is reported in the primary study. This usually happens in charts with unreadable scale or indirect values regarding effect size such as median, intervals and sample size – it should be noticed that box plots, commonly used to report quantitative data, includes both median and intervals. In fact, there are works that try to use these kind of data to estimate effect size (Hozo *et al.*, 2005). In these cases, it is recommended to use intervals to represent this imprecision or, in D-S theory terminology, a conversion to a compound hypothesis (*e.g.*, {NE, WN});
- **Comparative data:** only comparative data is reported, usually in terms of percentage difference. This is the only case where the dismembering operation is indicated along with the respective conversion rules (Table 7). In all other cases, descriptive theoretical structures for each (compared) cause should be modeled and from them the comparative theoretical structure is obtained. For the cases where, apart from comparative data, some additional but imprecise data is reported, more precise conversions than the suggested in Table 7 can be justified.

Recommendations given in this section cover the main heuristics and procedures necessary to identify concepts and relationships when modeling theoretical structures.

There was no particular order in which these recommendations were presented. Moreover, although aspects for concepts and relationships were described separately, we should highlight that the identification of concepts and relationships does not need to be performed in distinct moments and, in fact, they complement each other. In other words, eliciting some concepts can lead to the identification of new relationships which, in turn, influence uncovering other 'hidden' concepts, and so on. In fact, it can be argued that the extraction for and translation to a theoretical structure is a kind of synthesis (with an interpretative nature). Indeed, this combination of interpretative and integrative synthesis is inherent to most research synthesis methods (Dixon-Woods *et al.,* 2005). In SSM, this combination manifests in steps 4 and 5, which are iteratively performed as also are interpretative and integrative phases in other methods.

SSM does not define any specific recommendation for any of the ten semantic constructs present in theoretical structures. For instance, there is no discussion about how a researcher should use 'part of' relationships or how a moderator can be identified in primary study reports. Besides the detailed description presented about these semantic constructs, we assume that further understanding of what these semantic constructs represent and how they can be used in modeling is a basic skill set for researchers with knowledge about primary study research methodologies, particularly in SE where much of these constructs are used in ontologies and languages such as UML.

Furthermore, although it is indicated the direct extraction and translation from studies' reports to theoretical structures, this does not eliminate the possibility of using other kinds of information management and organization. Notes, worksheets and tables are all examples of instruments that can be used to that end. Tables, for instance, are specifically indicated in some methods such as meta-ethnography (Silva *et al.*, 2013). And information in extraction forms (Biolchini *et al.*, 2005) can be of great value. In summary, researchers are free to use any supporting instrument to organize information for modeling theoretical structures for aggregation in step 5.

5.4.5 Step 5: aggregation and analysis

Aggregation is essentially formed by the procedures described in Section 5.3, where the method core, called 'Structured Aggregation', was described. Even though all other defined steps and procedures are indispensable for conducting high quality research syntheses with SSM, it is in the manner that the aggregation is performed that SSM differentiate itself from other methods. SSM takes advantage from a formal diagrammatic representation to make most of the reasoning regarding the aggregation explicit. For that, SSM reduces the problem of evidence comparison to the identification of a correspondence among theoretical structures. This does not eliminate some degree of subjectivity in the process as a whole, since knowledge transformation, especially when modeling theoretical structures, is intense in researchers' interpretation and analysis. This is a strong aspect of SSM as a research synthesis method, given that it balances the transparency that comes from more formal procedures and representations, usually present in quantitative methods, with the flexibility that emerges from more analytical forms of investigations, commonly found in qualitative methods.

After aggregation, with the determination of compatibility between theoretical structures and computation of combined belief values of effects and moderators, it is time to analyze the obtained results. We identify two typical scenarios for aggregation results analysis. The first scenario occurs when the conflict level among evidence is low, making the belief values of the resulting combined theoretical structure to increase. This is usually the expected result from a research synthesis, since evidence results reinforce findings from each other turning the final result stronger. The second scenario requires a more careful analysis from researchers. It occurs when there is high conflict level among aggregated evidence, either with differences in the intensity or direction of effects and moderations. Depending on the case (difference in intensity or direction) and on the conflict level (in D-S theory terms), alternative options (presented in Section 5.3.2) should be considered: ignore, explain and incorporate. In either case, the researcher should also indicate the number of studies aggregated for each effect in the face of the belief value found, considering the two scenarios described above.

Analysis must concentrate particular attention to make a direct link to the research question. Researchers should report and justify the increasing (consensus) or decreasing (conflict) of effects and moderators belief values, and their direction and intensity. In addition, the analysis should try to explain the findings with additional facts about the observation context, study design, technologies involved, or any other element that can help understanding mechanisms behind how these effects and moderators operate. This is important not only to better answer the research question at hand, but also for anyone interested in observe the same results in their context – especially practitioners which are usually interested in the benefits and drawbacks of software technologies. There should be also considerations about generalizations made in the aggregation, particularly discussing if the results can be generalized beyond the aggregated studies' context and presenting appropriate validity threads in that regard.

Indeed, another important aspect of aggregation results analysis is a careful examination of possible validity threads. Apart from aggregation procedure and possible generalizations involved in it, many other aspects of SSM are subject to validity threads. For instance, as SSM first three steps are basically inherited from systematic literature reviews, the same typical threads found in it are also applicable to SSM such as the definition of the primary study sources, incomplete search string, improper criteria for study selection, and the manual selection of studies based on these criteria.

Still, there are many other validity threads particular to SSM studies. Arguably the most important, although this same feature is also considered one of its strengths, is the presence of procedures dependent of researchers reasoning and, thus, susceptible to some level of subjectivity. Some procedures exposed to this issue in SSM are how the quality assessment checklists are used by researchers and how theoretical structures are modeled. This kind of 'subjectivity issue' is inherent to most research processes and, for this reason, there are well established practices commonly adopted to avoid threads associated to this. One is make the process as transparent as possible. This is why, for instance, SSM incorporates the coding process from qualitative methods and the translation procedure from meta-ethnography. A formal representation also improves transparency of aggregation, since the decision about when evidence are compatible are based on explicit criteria defined from the representational constructs. The other recommendation is to include more than one researcher in the process so that ambiguities and distinct interpretations can be resolved along the process. Another common issue associated with research synthesis methods is the quality of the primary studies, the validity to threads that they have themselves and what these threads represent to the aggregation as a whole. This is particularly important in SSM, which is able to aggregate studies with wide variety of quality attributes. Researchers must discuss this explicitly in their analysis so that anyone interested in the aggregated results is aware about the possible drawbacks.

5.5 Conclusion

As a research synthesis method, SSM is proposed with a particular emphasis in knowledge (*i.e.,* evidence) representation considering the heterogeneity of primary studies in SE in terms of qualitative and quantitative approaches used in the area. Its blend of qualitative and quantitative method makes SSM appropriate to depict the important contextual aspects and to inform the trend of the effects (*e.g.*, positive or negative), as well as a confidence estimation about them. In this respect, SSM aggregates neither precise quantitative findings nor rich qualitative descriptions.

From the qualitative research perspective, SSM is a case-sensitive approach (Rihoux, 2006) in the sense that each study is considered individually. With theoretical structures, SSM preserves important information about original studies throughout the process, which play important role in the aggregation procedure to support the identification of compatible evidence. SSM's qualitative essence also manifests in modeling theoretical structures, as it requires a relative amount of interpretative synthesis reasoning from researchers.

From the quantitative research perspective, SSM has distinctive features as well. In terms of scalability, SSM allows researchers to analyze more studies than usually is possible with more qualitative methods. For instance, Silva *et al.* (2013), mentions scalability as an important issue for meta-ethnography given the huge number of data that must be organized to keep translations and translations explicit. Theoretical structures partially solve this issue as it offers a formal representation with well-defined semantics for concepts and relationships. Furthermore, aggregation most important operations are based on mathematical theory which, although laborious if performed manually, are systematic and algorithmic. Together, these two features contributes not only to SSM repeatability, but also in setting important parameters for computational support.

Even though SSM incorporates many methodological aspects from existing research synthesis methods, we can highlight some distinctive features of the method:

- Definition of a confidence level for evidence results (represented by D-S theory belief value), which is computed systematically (evidence type and quality assessment checklist) even for qualitative studies. In this way, simple vote counting or unsystematic qualitative interpretations are replaced by the weight of each individual evidence. In fact, this put SSM closer to statistical meta-analytic method which has as one of its distinctive features the ability to weigh each evidence contributions by statistical means. This is why we have put SSM among integrative methods, even with its prominent interpretive features;
- Usage of diagrammatic formal representation for evidence aggregation seems to be unique among research synthesis methods;
- Usage of a *Likert* scale for effect qualification, although it seems to be an intuitive instrument to give more precision for effect sizes, it is seldom used in research synthesis methods.

On the negative side, on the other hand, we can cite:

- Theoretical structures, in spite of being appropriate to some interpretive needs in terms of concepts definition and their relationships, can be limited in situations where research questions and goals requires comprehensive and detailed explanations using intricate lines of arguments;

- When only quantitative evidence are aggregated, SSM is able to produce consistent outcomes considering the combined evidence results, but the use of the *Likert* scale makes it less precise than pure statistical meta-analytical methods.

The next two chapters will explore some of these qualitative and quantitative characteristics more in depth. An experimental study with SSM will investigate how researchers apply their interpretative skills with theoretical structures and will try to observe the scalability of this process. Next, we describe the definition and development of a computational infrastructure that explores the formal aspects of SSM to support its execution and automates some of its procedures, particularly those involved in computing the D-S theory rule of combination.

6 An experimental study of knowledge translation using theoretical structures

In this chapter, an investigation about the usage of theoretical structures as a representation for evidence is presented. The study was designed as a regular SSM application, even though focused on step 4 to observe how researchers model evidence to translate knowledge of primary studies in SE. In total, ten researchers (graduate students working groups) participated in the study. All groups were able to finish the study tasks modeling four evidence. Considering the common concepts present in the groups' diagrammatic representations, a similarity level of 48% was observed. Although this similarity level seems to be low, we believe that the differences among subjects' interpretation of papers justify this amount which in turn indicates the SSM applicability. In addition, the mean time of about 10 hours to apply SSM for the four papers can suggest its scalability when used as a research synthesis method.

6.1 Introduction

Knowledge translation is the activity in the middle of knowledge creation and application. In fact, knowledge creation is a science endeavor and, thus, beyond of any control from knowledge translation perspective. Only proposition of research guidelines for primary studies or even adaptation or development of new research methods can have some influence on this aspect. Knowledge application, on the other hand, is the immediate goal of knowledge translation. Thus, it is directly influenced by the outcome of knowledge translation either in terms of primary studies interpretation quality or by the format that it is presented.

From the investigative standpoint, in this chapter we are interested in how researchers can use SSM and the evidence representation to translate knowledge – that is, in the translation activity itself. Investigate knowledge application, although an important aspect related to knowledge translation, would require dedicated research efforts since there is no definition about what are the information needs of software engineers and researchers (Budgen *et al.*, 2013). Still, as previously mentioned, we hypothesize a unified representation for evidence can benefit evidence-based practice, as it is supposed to have influenced Medicine and the adoption of statistical methods as the ‘gold standard’. The definition of a more comprehensive set of information necessary for knowledge application (Straus *et al.*, 2013) seems to come after the definition of a common basic format for evidence representation. The evidence representation used in SSM offers a minimal set of information for knowledge application in SE (*i.e.*, effects and moderators of software technologies in addition to contextual information), which is presented in a diagrammatic format. Further definitions for knowledge application could follow and be based on the definitions of SSM’s knowledge representation.

The study reported in this chapter addresses the following research questions:

- **RQ1:** Is the representation used in SSM capable of translating (i.e., representing) primary studies? Does it provide sufficient semantic and syntactic constructs to that end?
- **RQ2:** Do the procedures and steps guiding SSM support the goals of a research synthesis?
- **RQ3:** Do the SSM’s procedures and representation support the identification of what studies can be aggregated?

In summary, the research questions try to capture the SSM feasibility as a research synthesis method, observing if it can be used to represent studies' results (i.e., evidence) and organize them under the same format for aggregation. RQ1 and RQ2 will be explicitly addressed as part of the study design, and RQ3 will be more investigatory based on subjects’ perceptions and understanding of the representation.

The chapter is organized as follows. Section 6.2 describes the study procedures. The study results are presented in Section 6.3 and are discussed in Section 6.4. We conclude this chapter presenting validity to threads in Section 6.5 and final considerations in 6.6.

6.2 Study design and procedures

Using Goal-Question-Metric approach, the goal of the study is defined as follows. To *analyze* SSM *for the purpose of* characterizing *with respect to* its applicability (similarity of subjects results, execution time and participants subjective perceptions) and its representation capability (number of representation faults and percentage usage of the representation semantic constructs) for evidence aggregation *from the viewpoint of* the researcher *in the context of* graduate students from the Experimental Software Engineering class at Federal University of Rio de Janeiro synthesizing primary studies regarding test-driven development. The research questions RQ1, RQ2 and RQ3 were derived from this goal.

The main experimental units in this study are the papers used in the synthesis, the syntheses produced by the groups of participants and the subjects themselves. Ten PhD and MSc students were organized into three groups of pairs, one group with three persons and one individual ‘group’. The intention with this organization was to observe some influence of the number of researchers in the results. Characterization of the participants was made through a questionnaire (Appendix B) that included questions about the experience with data analysis (quantitative and qualitative), primary study research methods, research synthesis methods and SE practices related to test-driven development. From the ten graduate students, seven were PhD students. The software development experience mean was of 8.9 years with a standard deviation of 5.5 and the groups were assembled trying to balance experience among them – except the individual ‘group’ which was formed by the most experienced researcher. Several participants had knowledge about systematic reviews (median of 3 in question about systematic reviews using a scale from 1 to 5) and, in general, were more knowledgeable about quantitative (median of 3 in questions about quantitative analysis using a scale from 1 to 5) than qualitative (median of 2 in questions about qualitative analysis using a scale from 1 to 5) data analysis. Moreover, subjects were not experienced in primary study research methodologies (median of 2 using a scale from 1 to 5).

Participants were trained in a four-hour and other two-hour sessions conducted during normal class days. In the first session, the topic was research synthesis in general, including a short overview of the methods reviewed in Section 5.2 and a detailed view of SSM. Moreover, as part of the course normal syllabus, students had been previously introduced to statistical meta-analysis. The following two-hour session focused on reviewing SSM and carrying out an assignment where groups read two short papers (one quantitative about design patterns (Scanniello *et al.*, 2010) and other qualitative related to technical debt (Siebra *et al.*, 2012)) and outlined the theoretical structures with evidence acquired from the papers. After this, the two groups were asked to sketch the theoretical structures on the whiteboard side-by-side. Each group selected one of the papers. All participants discussed about the difficulties and the possible explanations for the representations' differences.

Including the training sessions, the study was conducted from the end-November’13 to mid-January’14. Two more classes, the half of the second training session and a full four-hour class, were made available for the remaining study tasks. The study was not time constrained to these sessions and subjects were able to hand over the final results later on.

The tasks involved in the study execution were essentially related to the SSM procedures described in the previous chapter. Given the experiment scale restrictions, but also because the first two steps of SSM procedures are elementary to any systematic review, only steps 3, 4 and 5 were exercised. Consequently, subjects were given a pre-defined research question (step 1) and a pre-selected set of studies (step 2). Four studies (Marchenko *et al.*, 2009, George and Williams, 2004, Slyngstad *et al.*, 2008, Erdogmus *et al.*, 2005 – namely, Paper 1, 2, 3 and 4 from now on) from test-driven development literature were selected to address the following question: “what are the expected effects of test-driven development in software projects?”. The papers were taken from a systematic review (Causevic *et al.*, 2011). Still related to the tasks, it is important to notice that step 5 was partially performed. Only the indication of what studies were compatible was requested. As the full aggregation including the application of the Dempster’s rule of combination without tool support is labor intensive it was left out of the study scope.

Along with the four papers, the subjects received a set of instruments to register the SSM procedures. The instruments basically consisted in word processor forms to be filled with study’s information. They received an execution ‘script’ (Appendix B.2) containing a step-by-step guide for study activities. Regarding SSM execution there were forms for quality assessment (step 3 – Appendixes A.1, A.2 and B.3); concepts, relations and explanation textual descriptions (step 4 – Appendixes B.4 and B.5); indication and justification of what studies were compatible (step 5 – Appendix B.6); and the registering of the time spent (all three steps and paper reading – Appendix B.7). The theoretical structures could be draw by hand or using any drawing tool – shapes templates were given for Microsoft Visio.

Regarding the main measures presented in the results Section, it is necessary to give more detailed justification of why they were selected and how they are calculated. Related to the SSM applicability we chose to measure the time spent and similarity of results obtained. Time spent is used to have a sense of the method scalability. In addition, similarity is used to perceive to which degree the method produces comparable results, which may minimally indicate how systematically SSM can be applied – even though, a low similarity between results can indicate more about aspects of SSM application (*e.g.*, text interpretation, modeling activities, among others) than how systematically the method is being applied. In this study the similarity is measured based on APPA (average pairwise percent agreement (Larsson, 1993)). We choose to use APPA because according to Larsson (1993) it is able to capture partial agreements between raters (in this case, study groups). For each paper, we compared pairs of theoretical structures elaborated by the groups and calculated the percentage agreement of each pair dividing the number of common concepts by the sum of total concepts on both theoretical structures. Then, APPA value is obtained from the sum of the percentage agreement of all pairs divided by the number of pairs. APPA was calculated for theoretical structures' relations as well.

Associated with the representation capability, on the other hand, we defined measures that could reveal the difficulties and limitations in using the representation. Hence, we quantified the number of faults and misuses committed when building the theoretical structures to indicate possible difficulties, and the percentage of representational elements indicate the representation understandability and usefulness. Besides all these measures, each subject answered a follow up questionnaire (APPENDIX B.8) to get the subject's perception about the SSM procedures and representation.

6.3 Results

6.3.1 Research question 1: Is the representation used in SSM capable of translating (i.e., representing) primary studies? Does it provide sufficient semantic and syntactic constructs to that end?

We begin this section by presenting the data for similarity among the theoretical structures created by the five groups. Table 10 shows the APPA values for the agreement of concepts. The agreement calculation for the concepts did not take into account the four archetypes concepts, since they are fixed. Group 5, with only one individual, was not considered in the overall APPA, since it was purposely unbalanced with the other groups. Indeed, we were able to find a difference between the APPA within groups 1, 2, 3 and 4, and the APPA between groups 1, 2, 3, and 4, and group 5. This can be an indication that more than one researcher should apply SSM, as with most research synthesis methods. Doing the same analysis for group 3, with three subjects, we were not able find such difference.

Data in Table 10 were also split by the paper data type and the consideration of synonyms. The type of data present in the paper seems to influence the definition of theoretical structures. One hypothesis for this observation is that qualitative papers have less explicit variables and are more open to different interpretations about the relevant study variables and its relations. Quantitative papers, on the other hand, usually state explicitly the independent and dependent variables. The analysis with synonyms was performed in a subjective manner, that is, the authors compared the concepts not only by the used terms but also by its descriptions. Although potentially subject to bias, the analysis with synonyms shows a difference of the APPA values comparing to the analysis without considering possible synonyms. Examining the used terms it was possible to observe that the discrepancy mostly originated because some groups decided to use terms directly extracted from the papers, while others tried to use terms that are more generic or more common in the technical literature than the used in the paper. Another source of difference among concepts used was the variation of the theoretical structures detail level (e.g., less/more concepts for contextual aspects).

Table 10 – APPA(*μ*) for theoretical structures concepts

Analysis	Data type	Groups 1,2,3,4		Groups 1,2,3,4 x 5	
		μ	***σ***	***μ***	***σ***
Considering synonyms	Quantitative	0.64	0.14	0.54	0.07
	Qualitative	0.47	0.11	0.39	0.14
	Both types	0.55	0.19	0.46	0.13
Without considering synonyms	Quantitative	0.53	0.14	0.43	0.08
	Qualitative	0.43	0.15	0.34	0.07
	Both types	0.48	0.17	0.39	0.09

Table 11 lists APPA values for *relations*. In this case, APPA is calculated counting the number of common relations divided by the total number of relations of each pair of theoretical structures compared. A relation is equal to another if the relations are of same type (e.g., property of) and the two concepts linked in the relation are also the same in both relations compared. Relations linking archetypes are included in the counting and concepts are not compared considering possible synonyms. Since comparing relations where both linked concepts are different can take the focus away from the relations types, we complemented the analysis comparing only the relations where at least one of the linked concepts is common to both theoretical structures. Using this analysis gives us a better sense of the difference in ‘modeling’ decisions in developing the theoretical structures. APPA values in this case are closer to the ones for concepts. A common source of differences in the case of relations was the variation in the level of detail adopted by subjects. For instance, some groups placed ‘developer’ directly as *type of* ‘actor’ (archetype) while others opted to put ‘developer’ as *part of* ‘software project’ which in turn was *type of* ‘actor’.

Table 11 – APPA(μ) for theoretical structures relations

Analysis	Data type	Groups 1,2,3,4		Groups 1,2,3,4 x 5	
		μ	σ	μ	σ
All relations	Quantitative	0.27	0.13	0.12	0.05
	Qualitative	0.19	0.13	0.13	0.07
	Both types	0.23	0.14	0.13	0.06
Relations with at least one common concept	Quantitative	0.48	0.23	0.22	0.10
	Qualitative	0.38	0.33	0.30	0.19
	Both types	0.43	0.29	0.26	0.15

More focused on the representation manipulation itself, the data in Table 12 shows the number of faults committed and Table 13 the average usage of representation semantic constructs. The most recurrent fault is not related to a modeling problem, but to the fact that the group that committed these faults drew the diagrams by hand. The remaining faults can be grouped into two main categories: theoretical structure misunderstandings (wrong effect, moderation and context definition and cause-effect strength specification) and interpretational and modeling problems (wrong causal definition and concept naming).

Table 12 – Representational faults using theoretical structures

Fault	Description	#
Missing belief value in diagrams	Missing from diagrams (but their values were registered in the study instruments).	11
Wrong causal definition	Incorrect determination of the main studied cause.	5
Concept naming	The concept name describes more than one characteristic and, thus, could be defined using distinct concepts.	3
Wrong effect definition	Improper usage of the effect element. The concept was not an effect.	3
Wrong moderator definition	Improper usage of the moderation element. The concept was not a moderator.	3
Wrong context description	Incorrect semantics of value concepts and its relationships (e.g., put an activity as part of software component).	1
Cause-effect strength specification	In a comparative theoretical structure, uses a descriptive strength indication (e.g., + or +++) or vice-versa.	1

Even though it was possible to identify some faults related to the theoretical structures modeling, Table 13 shows that only the *part of* and *moderator* semantic constructs were not used by some of the groups. Clearly, the decision to use a semantic construct depends on the presence of the information on the paper that requires this kind of construct and the modeling choices of how that information should be represented. However, it can also be a consequence of misunderstanding of that semantic construct. We tried to detect this examining if there were groups that did not use a semantic construct for any paper. Two groups (G2 and G3) have not used *moderators* in any of their representations, while only one group (G3) did not use in the case of *part of* relations. As G3 subjects were experienced software developers, including in UML, so it is more likely that the group deliberately chose to not use *part of* relations in their representations, leaving out some contextual aspects that other groups considered important. The same argument regarding experience in UML is not valid in the case of moderators' usage. Indeed, one of the groups explicitly stated that they were not confident on their understanding about moderators and could not take advantage from it.

Table 13 – Proportion of groups using each semantic construct (total of 5 groups)

Semantic construct	Papers				
	1	2	3	4	All(%)
Is a	5/5	5/5	5/5	5/5	100
Part of	4/5	3/5	4/5	3/5	70
Property of	5/5	5/5	5/5	5/5	100
Archetype	5/5	5/5	5/5	5/5	100
Cause	5/5	5/5	5/5	5/5	100
Value concept	5/5	5/5	5/5	5/5	100
Moderator	1/5	2/5	2/5	3/5	40
Effect	5/5	5/5	5/5	5/5	100

Table 14 presents post-study questionnaire answers related to RQ1. In general, feedback was mostly positive and several answers were followed by detailed justifications. The questions with lower agreement are 1, 2 and 3. For question 1, participants were concerned with specific missing information. For instance, one of the respondents stated that the link with original study hypotheses is lost in the representation and, thus, may turn the interpretation of the representation ambiguous. Yet, almost all participants agreed that a graphical representation is useful even if some detail is lost. In questions 2 and 3, it seems that even though archetypes appear to be helpful, there is still room for improvement. The activity archetypes seems to be the least important in the perception of subjects. Some subjects suggested the inclusion of new archetypes, such as study limitations and pre-conditions. We understand these concerns, but we believe that archetypes should be associated with SE aspects, not meta-level evidence concerns such as study characteristics. In addition, all situations, in which the suggested study limitations and pre-conditions archetypes would be used in the opinion of participants, could be represented with the four archetypes (using *part of*, *type of*, and *property of* relationships).

Table 14 – Post-study questionnaire questions regarding RQ1 (1: completely disagree; 2: somewhat disagree; 3: somewhat agree; 4: completely agree)

Question		**# of answers**			
		1	**2**	**3**	**4**
1)	Do the theoretical structures are able to capture the main results present in the primary studies?	0	0	4	6
2)	Do the archetypes support identifying which aspects of primary studies should be made explicit in the theoretical structure graphical representation?	0	0	4	6
3)	Are the archetypes equally important? Are they sufficient or other archetypes are needed?	0	2	4	4
4)	Do the theoretical structure could represent the aspects that you (the subject) considered most important?	0	1	2	7
5)	Could the conversion from the quantitative scale to the semi-quantitative (*Likert*) scale used in the theoretical structures be done appropriately?	0	1	1	8
6)	Could the conversion from the qualitative scale to the semi-quantitative (*Likert*) scale used in the theoretical structures be done appropriately?	0	2	1	7

Still regarding the post-study questionnaire, we made one open question related to RQ1 where we asked if participants ‘would modify any aspect of the theoretical structure diagrammatic representation’. Four of them answered with some suggestion or comment. Two complained about the absence of connections between cause and its effects and moderator and the moderated effects. One commented that theoretical structures could not represent how other factors (despite the cause) can influence the results. To justify this, one example from Paper 1 is mentioned because Test-Driven Development (*i.e.,* cause) was adopted with SCRUM. This in fact can be considered an important information to represent and it is precisely the role of *contextual aspects* concepts. Other groups were able to explicit this information, although one of them created a concept with two elements ‘TDD with SCRUM’, which is an indication that modeling this situation is not trivial, while other has not included the information about SCRUM. Particularly in this last case, it is not possible to know if the group deliberately decided to not put this information or if they had difficulties in modeling it. The fourth comment is not about graphical representation itself, but about its meta-model and the interesting possibilities that automated computational processing can bring (*e.g.*, information that the model have). The subject suggested that if a computational infrastructure was available, researchers could search for evidence based on similar context.

Conclusion for RQ1: as the most important question of this investigation, results about RQ1 are definitely not conclusive, but are enough to have promising indications about our hypotheses. The diagrammatic representation of evidence seems to be viable, as all groups were able to represent the evidence from the selected four papers despite their discrepancies in the reporting of their findings. The subjects demonstrated confidence in modeling evidence most important aspects using the representation semantic constructs (question 4 in Table 14). This was supported by the large use of the semantic constructs by most of the groups (Table 13) and the relative unimportance of the faults committed on our subjective evaluation (Table 12). Moreover, relative low similarity of the representation observed in the low APPA values does not seems to be strictly associated with the representation used. Our hypothesis are that this is inherent to any modeling process, which the using of a representation for evidence is subject to, but also to other factors such as the absence of a controlled vocabulary as shown in Table 10 and Table 11.

6.3.2 Research question 2: do the procedures and steps guiding SSM support the goals of a research synthesis?

The data collected for RQ2 are more indirect surrogates than in RQ1, since we had limited means to assure the compliance to the indicated SSM procedures. In fact, although we used APPA as an indication for the representation applicability, the theoretical structures created in the study were also subject to how the SSM procedures were executed. Thus, using APPA for RQ1 assumes that SSM procedures were followed. The fulfillment of the study instruments is an evidence that the process had been followed, since they were partitioned considering the SSM steps. The average time spent by each group in the research synthesis was 594 minutes with standard deviation of 143. This average is composed of paper reading time (μ=165; σ=73), step 3 (μ=137; σ=62), step 4 (μ=261; σ=93) and the partial execution of step 5 (μ=31; σ=23).

The post-study questionnaire questions related to RQ2 are presented in Table 15. Almost all participants agreed that SSM procedures support the research synthesis and it was possible to see that in their perception creating theoretical structures for qualitative studies can be more complex than for quantitative studies. Moreover, the estimative for belief values seems to need improvement. Some participants suggested improvements in the quality checklist (particularly, for unsystematic observations (Shull *et al.*, 2006)) while the most experienced subject was concerned with potential over-confidence in low quality quantitative studies.

Table 15 – Post-study questionnaire questions regarding RQ2 (1: completely disagree; 2: somewhat disagree; 3: somewhat agree; 4: completely agree)

7)	Is the five step process defined for SSM appropriate for the research synthesis needs?	0	0	1	9
8)	Could the extraction step (step 4) be executed without difficulties for quantitative studies?	0	0	3	7
9)	Could the extraction step (step 4) be executed without difficulties for qualitative studies?	0	0	7	3
10)	Is the belief level estimative (step 3), based on the evidence hierarchy and study quality checklist, appropriate for the aggregation purposes?	0	1	5	4

Conclusion for RQ2: Even in a difficult position to observe objectively if the steps and procedures really support the research synthesis tasks, we believe that we have collected interesting feedback for RQ2. From an usage perspective, considering the number of faults (Table 12) and the time spent on the tasks, it is possible to see that the procedures can be followed and concluded in a reasonable time in our evaluation. Strengthening these observations are the subjects' positive impressions, especially regarding the organization in five steps (question 7 in Table 15). Moreover, this is also backed by the fact that SSM steps and procedures are heavily based on well-established research synthesis methods.

At the same time, it was possible to identify necessary improvements such as how better support the extraction of information from qualitative studies. Consistent with the differences in APPA found for qualitative and quantitative (Table 10 and Table 11); subjects reported difficulties in extracting and translating qualitative studies in comparison to quantitative studies. These three following justifications regarding this issue describe the problem in more detail: (i) 'in this case (referring to qualitative data extraction), I had more difficulty because papers do not clearly present data. However, it was only a matter of putting additional attention in interpreting results'; (ii) 'I had some trouble defining effects intensity for qualitative studies, since in this kind of study this definition becomes subjective'; and (iii) 'relatively more laborious (in comparison to quantitative extraction), but if the paper is well written and clearly report results it is possible (to perform the extraction)'[23]. As it is possible to see, the problem seems to be

[23] Text in parenthesis was added to contextualize the justification from the subjects.

more related to how data is presented in papers than to the SSM procedures themselves. Indeed, most qualitative research methods mention this aspect about dealing with qualitative data. Still, there were even subjects who reported the problem for both types of data: (i) ‘the problem is with extraction is that each paper report result in a different way’ and (ii) ‘the easiness of data extraction depends on the availability and clearness of papers information’. As a result from this analysis, we included more details about coding and extraction in step 4.

6.3.3 Research question 3: do the SSM’s procedures and representation support the identification of what studies can be aggregated?

As RQ3 was not the main focus of the experiment, we have only a couple of observations regarding it – still, important for the overall study context and considerations. The first was a question included in the post-study questionnaire where we asked if the theoretical structures were useful to determine if the primary studies could be aggregated. Three possible answers were given: no, the theoretical structures do not help (0 responses); yes, but reading the primary studies’ papers improve aggregation reliability (6 responses); and yes, only the theoretical structures are sufficient (4 responses). All subjects perceived the theoretical structures useful for determining evidence compatibility. However, most of them consider the reading of original papers important to reduce the threats to its validity. The most cited concerns for the necessity of reading the original paper include the evaluation of the theoretical structures correctness and the fact that there are minor contextual aspects that are not captured which can be important for aggregation. Yet, five respondents mentioned that theoretical structures are helpful because it narrows the researcher perspective to the most important aspects – which is exactly the role of visual representations (Worren *et al.*, 2002).

The second observation is related to the groups’ agreement on the step 5 of SSM, as shown in Table 16. It is not possible to affirm that the results in Table 16 are consequence of the use of diagrammatic representations as we would like to observe in RQ3, since the participants read the papers. Indeed, even though we asked subjects to base their decisions only in the theoretical structures, the number of answers to the post-study questionnaire mentioning that papers reading is important for deciding about studies compatibility gives an indication about the fact that reading the paper before has influenced participants. However, most justifications for or against evidence compatibility were based solely on the theoretical structures concepts, which is what is required on step 5. Nevertheless, it was possible to notice that one group (G3) made some generalizations based on their interpretations of the concepts to come to conclusion about results compatibility. In addition, it was interesting to observe that for one pair of papers there is a 100% agreement, especially considering that one is qualitative (paper 1) and the other quantitative (paper 4). There was not a template for comparing the results 'correctness'. Even if the authors provided one for comparison, subjects' results could still be different from ours. The 'mean agreement', considering the agreement level achieved in each of the six pairs of studies, is 0.68.

Table 16 – Agreement on determining studies compatibility for aggregation

Groups pairs	**Papers compatibility agreement (0 means divergence regarding the pair compatibility and 1 otherwise)**					
	1x2	**1x3**	**1x4**	**2x3**	**2x4**	**3x4**
G1xG2	0	1	1	0	0	1
G1xG3	0	0	1	0	1	0
G1xG4	0	1	1	1	1	1
G1xG5	0	0	1	1	1	0
G2xG3	1	0	1	1	0	0
G2xG4	1	1	1	0	0	1
G2xG5	1	0	1	0	0	0
G3xG4	1	0	1	0	1	0
G3xG5	1	1	1	0	1	1
G4xG5	1	0	1	1	1	0
APPA	0.6	0.4	1	0.4	0.6	0.4

Conclusion for RQ3: as a result of our analysis regarding RQ3, it seems that researchers, at least the study subjects, perceive some utility in using diagrammatic representation as a tool for aggregation. We expected a higher agreement on studies compatibility determination (Table 16). Clearly this was influenced by each subject interpretations from the papers reading, which would be highly reduced if they only had access to the evidence diagrammatic representation to conclude about the studies compatibility. This is open for further investigation in which a study design captures this aspect.

6.4 Discussion

Even with its observational orientation, an experimental investigation of a complex research method such as research synthesis is challenging. It is difficult to show objectively the applicability of such kind of method since it involves intense reasoning by highly educated researchers with different background. There are not necessarily right or wrong results, but only different interpretations – particularly, considering the combination of quantitative and qualitative studies. Evidently, even though differences in results can be accepted, major signs of maturity of a research synthesis method are its methodological resources to improve the rigor or systematization of how the results are obtained and to show transparency of what are the steps followed to that end. That is why we cited the simple conclusion of study tasks as an indication of its applicability.

Still, we tried to examine in more detail several aspects of the SSM usage to indicate its applicability, both from the researcher subjective perceptions and from the results perspective (similarity of results, usage of diagrammatic representational elements and number of faults). In general, we believe that the results presented in the previous section contributed for an indication of SSM applicability, especially regarding RQ1 and RQ2. Most of subjects' perceptions were positive, although we could detect some opportunities for improvements as more representative belief estimation and seek better guidance for extracting information from qualitative papers[24].

From the results perspective, we could find an overall 48% of similarity of concepts found in the groups' theoretical structures. We understand this is a low similarity percentage. However, this number should be interpreted considering that concepts present in theoretical structures are subject to how papers are interpreted and have its information extracted. In addition, the situation is worsen if we take into account the absence of a well-defined taxonomy in SE. Thus, for instance, when higher-level concepts are used rather than concepts directly extracted from the papers, it is not difficult to understand why different concepts might be chosen. Moreover, from the type of faults found in the representation, besides not in great quantity, we can hypothesize that this can be partially attributed to the typical learning process of any new research method. Another possibility for the faults committed by subjects is a result attempting to represent an aspect for which the representation was not prepared for. Thus, the type of faults observed combined with the fact that most of the representation semantic constructs were used, suggests the representation's expressiveness suitability in terms of its capacity in addressing the heterogeneity of evidence in SE evidence.

The difference in theoretical structures created by the groups manifests the SSM latent constructivist epistemological disposition. This can be attributed to blending interpretive and integrative methodological features in SSM. The study shows that, although SSM was attempted to be an integrative method, its interpretive nature plays

[24] Even though SSM description evolved after this study, we have opted to keep SSM presentation (chapter 5) before than the study (this chapter), since there were not major changes in SSM presentation and description. Belief estimation has not changed after this study, but we have detailed the extraction procedure.

an important role as theoretical structures are subject to coding process. A practical consequence of this fact is a proper attention to the threats to validity particularly with aspects concerning with the results consistency (with the primary results).

Another interesting aspect of this investigation is that both quantitative and qualitative were aggregated. When evaluating evidence compatibility, there was a 100% agreement in the compatibility of two papers (paper 1 and 4). As we mentioned previously, our study design did not favored any kind of primary study or data type. Thus, this result regarding decidability about compatibility of evidence from papers 1 and 4, respectively quantitative and qualitative, is representative from SSM purposes standpoint.

6.5 Threads to validity

The most important limitations in this study are related to internal and construct validity threats. For internal validity, we can cite the possible bias from the subjects' background both in research methods and in test-driven development. Moreover, although study execution was not time constrained, it was possible to observe in some moments a certain level of fatigue with the amount of work to be done (e.g., read the papers carefully, then evaluate its quality and extract information). In fact, subjects took about 2:45 hours to read the paper. To our understanding, this is a not a realistic time unless participants focused on specific paper's sections, which is our hypothesis. Furthermore, as previously mentioned, the interpretation of papers and the absence of taxonomy might also influenced the findings.

Related to the construct validity, the major threats are associated with the appropriate definition of surrogates for observing the SSM applicability and the theoretical structure representation capability. We tried to minimize this issue by triangulating those indicators with the subjective perceptions collected in the follow-up questionnaire.

There are also external validity threats, and as an initial study, we are not in the position to claim any kind of generalization. Nevertheless, it should be noticed that participants, except for regional considerations, are graduate students in SE with varying formation as in found in many other institutions. Besides, we believe that the research synthesis topic, in this case test-driven development, has least influence in the results than the type of studies or kind of evidence aggregated. The four studies used in aggregation were purposely selected given their differences in this regard.

6.6 Conclusion

In this chapter, we investigated a central element of SSM which is the diagrammatic evidence representation. The study was designed to observe how theoretical structures are used by researchers to represent evidence for knowledge translation. Thus, the study was controlled in different factors, especially steps 1 and 2, and part of step 5. Furthermore, we believe to had set up a scenario fairly similar to real settings including a relevant SE theme (test-driven development), heterogeneity of evidence types (qualitative and quantitative) as well as diversity in reporting styles (mixing papers from conferences and journals) and even in scale (4 papers).

Our understanding from the study results, at least in this stage of SSM characterization, is the feasibility of using theoretical structures as an instrument for knowledge translation in SE and a stronger confidence in SSM applicability as a research synthesis method. Researchers seem to be able to follow SSM guidance when modeling theoretical structures to represent evidence from research papers. In addition, we could have a perception about the method scalability with a mean of 10 hours to perform most part of SSM steps for four papers.

Still, some aspects of SSM were left out in this investigation. We believe that part of them do not affect the conclusion of this study while others can be further investigated. Defining a research question (step 1) and selecting research papers (step 2) are typical activities for most researchers and, thus, does not seem to affect the perception of SSM characterization. Regarding SSM aggregation step, researchers read papers before deciding their compatibility based on theoretical structures, but it was still mandatory to justify compatibility decisions with some information from theoretical structures. While this is a typical situation in research synthesis methods, where researcher read papers before aggregation, we would like to also observe if the diagrammatic representation is sufficient for aggregation purposes. It could be also argued that participants did not compute pooled belief values using D-S theory. However, this can be easily automated with computational support and is precisely the theme of the next chapter. Another aspect not investigated in this study, is that we have only explored theoretical structure ‘production’ not its ‘usage’. Although it can be hypothesized that if researchers are able to develop evidence models they will also be able to read it, we would like to explore if professionals from industry would also be able to read evidence models and if they could use them to support decisions in practice. This is an essential part of the Figure 1 model for evidence-based practice in SE.

7 The Evidence Factory computational infrastructure

This chapter presents the tool support for SSM named Evidence Factory. We detail the design decisions involved in its development, considering the scientific knowledge engineering conceptual framework. In addition, the tool's facilities described linking them to SSM definitions.

7.1 Introduction

A research synthesis study is a complex task whether we consider the amount of extracted information to manage or data to aggregate manually. In addition, the effective communication of research results, particularly considering the importance of synthesis outcome, can be amplified with web platforms. This is the case of digital libraries and specialized online networks (e.g., www.cochrane.org).

Tool support for SSM is thus almost imperative. The first two steps of SSM have extensive investigations in the context of SE seeking to identify the appropriate support for systematic reviews. Marshall and Brereton (2013) present a mapping study of tools to support systematic literature reviews in SE in which 11 tools are identified. The authors describe several techniques and technologies have been explored in these tools including visualization techniques, text mining algorithms, search heuristics and ontologies. There also proposals focusing in supporting systematic review process as a whole. This illustrates how systematic literature review process demand careful organization to its planning, execution, data analysis and packaging.

In this chapter, we present a web computational infrastructure to support the research synthesis activities based on the conceptual proposal of SSM. The tool is named Evidence Factory in a reference to the Experience Factory (Basili *et al.*, 1992), and is also used in an allusion to evidence 'construction' (*i.e.,* modeling) and publishing facilities which form the basis of the Figure 1 (presented in Section 1.2) model for evidence-based practice in SE based on a unified evidence representation.

We follow the process for scientific knowledge engineering proposed in Chapter 4 to describe the computational infrastructure. However, as most aspects for the definition of the computational infrastructure according to that process were covered in previous chapters, we split the text into two main parts. Part one, which is presented in the next

Section 7.2, resumes the steps that were already covered making a link between each phase of the process and previous chapters. Then, the second part on Section 7.3, details the aspects not yet covered, which is essentially the last step of scientific knowledge engineering process ‘computational infrastructure implementation’. We also describe in detail most important facilities of Evidence Factory in Section 7.4. Both Section 7.3 and 7.4 are heavily based on the work published of Santos *et al.* (2015). At last, as not all facilities are implemented at the current development stage of the infrastructure, we conclude this chapter pointing to future works.

7.2 Scientific knowledge engineering: the case of Evidence Factory

7.2.1 Familiarization

This step was facilitated by the author experience and expertise in both Software Engineering and research methods as researcher in the Experimental Software Engineering area. For this reason, we did not collect and read a representative set of papers with the specific purpose of finding out how scientific investigation is described in the area and what its main characteristics are. Our first intuition was that a focus on meta-analysis research synthesis in SE would conflict with the high heterogeneity of evidence types (i.e., qualitative and quantitative) in the area. This intuition was corroborated with the small number of systematic literature reviews in SE that used meta-analysis for research synthesis (Cruzes and Dybå, 2011b, Silva *et al.*, 2013). This diagnostic was also based on a huge number (Cruzes and Dybå, 2011b) – in fact, most part – of works in SE which present research syntheses without being explicit about the research synthesis method. We interpreted this as an indication of the complexity of performing this kind of study in SE.

Therefore, our first insight was to develop an instrument that could represent all evidence types on the same perspective so that they could be aggregated. A secondary, but not less important objective, was that this instrument could be relatively easily used by both scientists and practitioners supporting knowledge translation.

Thus, the result of this initial familiarization step was a decision to concentrate efforts on cause-effect studies, independently from research methods or type of data collected and analyzed. This decision was based on the notion that this study type is common in SE and also that it is simpler to structure it, based on the expectative that fewer knowledge formalization constructs would be needed to relate causal concepts than more complex and detailed knowledge from specific SE domains.

7.2.2 Knowledge initial organization

Since our scope comprised the whole Software Engineering area and given the diversity of Software Engineering study types, contexts and technologies, we concluded that most relevant directions for knowledge organization could come from epistemological orientations. Based on this premise, we revisited the Experimental Software Engineering research method handbooks (e.g., Shull *et al.* (2007)) and searched for publications that discussed whether the way evidence is described in study reports affects the development of the Software Engineering body of knowledge (e.g., Rainer *et al.* (2005); Shull *et al.* (2006); Dybå *et al*. (2007)). These works increased our confidence that cause-effect relations were amongst the most general notion associated with scientific results in the area. Plus, as regards the organization of the body of knowledge, it became apparent the need for a more structured representation for evidence (including scientific findings and practical experiences) so that scientists and software engineers could capitalize on it.

At that point, it was still not clear whether a graphical representation could help and which representation (e.g., graphs or UML) would be the most appropriate. Yet, on the other hand, we left this stage with a clear justification for using the cause-effect notion to link different types of studies and qualitative and quantitative research results using the arguments found in the work of Mahoney and Goertz (2006). According to them, both quantitative and qualitative studies are usually concerned in uncovering or observing cause-effect relationships. Qualitative research "explains individual cases; using the causes-of-effects approach" and quantitative research "estimates average effects of independent variables; using the effects-of-causes approach". Thus, as causal notion can be present in almost any type of study we concluded it could be used as a focal point of evidence aggregation.

7.2.3 Domain conceptualization and knowledge representation definition

We describe these two steps together as they were somewhat developed at the same time. Our initial move was to first search for works in the area that proposed conceptualizations or representations for evidence or any description of organization for scientific knowledge in SE, especially from an epistemological standpoint which had been previously defined as a target. The idea was that even if papers focused on technical representation aspects it would be possible to abstract conceptualizations useful for the main goal: representation of causal evidence. The search resulted in the identification of two papers – amongst others – none of which addressed the theme directly, but had useful conceptualizations for our goals. Ivarsson and Gorschek (2012) present a tool support for disseminating SE practices – which could be conceptualized as evidence – used in an organization. And Sjøberg *et al.* (2008) propose a framework for describing Software Engineering theories as already described in Chapter 2. The first work, despite its relatively general applicability, had too much detail in its frame-based knowledge structure for practices and experiences and, given the wide scope we were aiming at, it was not used for our domain conceptualization. Still, it was helpful to perceive the minimal requirements for the application of any evidence representation for software engineers in industry. The second paper, on the other hand, had most of the elements we were searching for in terms of domain conceptualization and knowledge representation. In fact, given its UML lineage it have most elements already defined in terms of knowledge representation. Although not formally defined, the description given in Sjøberg *et al.* (2008), detailed in Chapter 2, defines all possible concept types and relationships between them.

Interestingly enough, Sjøberg *et al.* (2008) did not propose the representation to be used for evidence, but to support theory building. Nevertheless, we could identify almost all elements needed for knowledge translation. Besides, it was a straightforward to map the cause-effect notion into the cause-effect and moderation relationship types of the notation. The generalist nature of theories embodied in the representation was an essential feature for the needs of research synthesis in SE. Thus, at this point we were convinced that the notation would be applicable to most of the situations. At least, to what we could conceive. Still, it was necessary to add a minor extension to Sjøberg *et al.* (2008) proposal, which was an ordinal seven point *Likert* scale for the cause-effect relationships as described in Chapter 5.

7.2.4 Knowledge utilization process definition

In this step, we followed our initial determination of backing definitions and decisions on epistemological foundations. Given our primary goal of evidence representation and aggregation, we understood that the kind of procedures needed for process definition would come from the common practices in research synthesis methods. Common to virtually any research synthesis method is how knowledge extracted from studies undergoes intense transformation. This transformation usually aims at translating individual results to a representation that allows them to be analyzed in the same perspective. This is exactly what was described in Chapter 5, when SSM process was presented.

7.2.5 Inference strategy specification

Using ontological elements from UML (e.g., *type of* or *part of*), it was possible to apply description logics to develop inferences needed for the goal of aggregating evidence. For instance, the results from two different studies describing the effect of *ad hoc* software inspection and checklist software inspection on software source code quality are, at first glance, not comparable: they are distinct software methodologies (ad hoc and checklist). But if we say that both methodologies are *type of* inspection, then we can represent both evidence under the same representation, generalizing both software methodologies (technology archetype) as inspections. This shows how description logics can be used to detect *compatible* evidence.

Determining the compatibility among pieces of evidence was part of the aggregation problem. It was still necessary to define how evidence results would be aggregated. In that regard, we extended the knowledge representation with uncertainty formalisms to represent the strength on the observed outcome for each study. An uncertainty formalism captures the confidence strengthening on the outcome when the evidence converges or its weakening when it diverges. The mathematical theory of evidence was used as the uncertainty formalism as described in chapters 3 and 5 (particularly on Section 5.3.2). The connection between the knowledge representation (defined in 'knowledge representation definition' step) and the inference strategy using the mathematical theory of evidence was materialized through the definition of the seven-point *Likert* scale for effects intensity as the *frame of discernment*.

Inference strategy specification concludes the sixth of the seven steps scientific knowledge engineering process. As it could be noticed, these steps were already covered in previous chapters. The focus of this chapter is the last step of the process which is the computational infrastructure implementation, presented in the next section.

7.3 Computational infrastructure implementation

All infrastructure's requirements, architecture and design decisions were defined after the SSM proposal following the steps defined above. 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 focus on 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.

Functional requirements are categorized into four groups (requirements are enumerated in Appendix D):

- Storage and processing: these are the essential requirements upon which the infrastructure facilities are built. They are associated with knowledge formal representation for the theoretical structures, the implementation of the D-S theory uncertainty formalism, and the support to determine theoretical structures (i.e., evidence) compatibility for aggregation. 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 of SSM. The tool should also maintain all data provenance 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 studies.
- Facilities for practitioners: as the representation is intended to be used by professionals, there are conceptual foundations already defined to support professionals in using the representation to create evidence. Thus, the requirements associated with facilities for professionals define how practitioners can take their experiences into and from the computational infrastructure as part of their continuous improvement cycles. Requirements are based on iterative processes for improvement (Salo and Abrahamsson, 2007) and inherits concerns from Experience Factory regarding how evidence about practice can be accumulated (Basili *et al.*, 1992). They aim for allowing theoretical structure knowledge formalization as part of professional regular activities.
- 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 indicate the appropriateness of each.

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, several 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 Figure 19. The tool is deployed within Experimental Software Engineering group at COPPE domain http://evidencefactory.lens-ese.cos.ufrj.br/.

7.3.1 Architecture and formal evidence model

The infrastructure was constructed as a typical web application architecture inspired on a Model-View-Controller style (Figure 19). 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 as discussed in Chapter 4. It is also interesting to notice the evidence graphical 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.

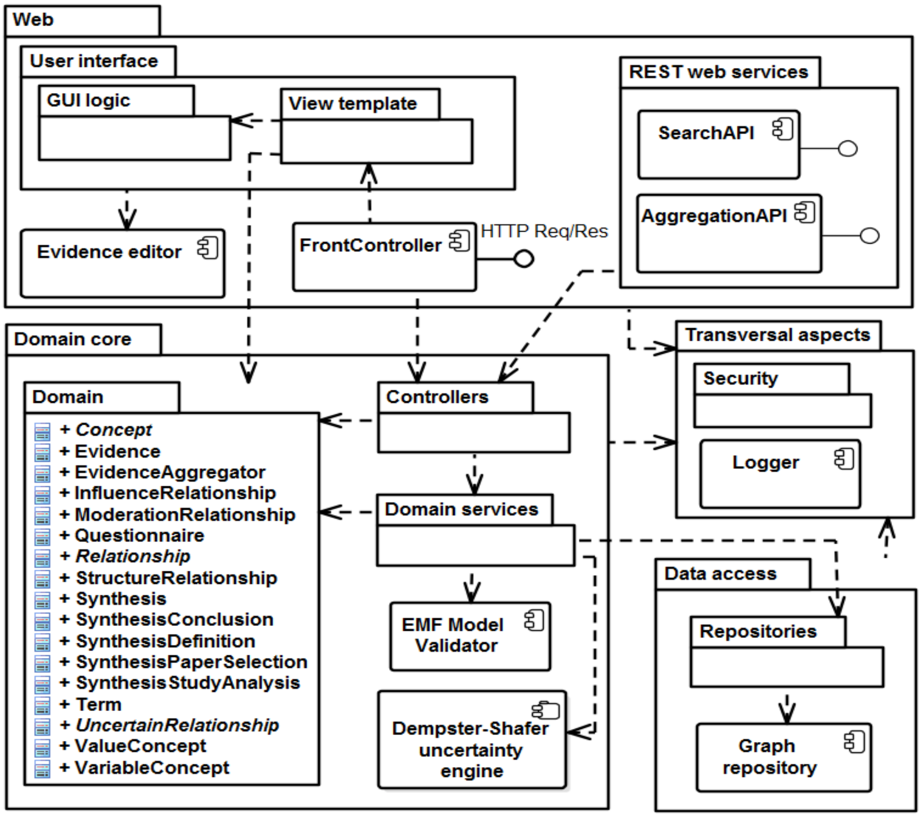


Figure 19 - Infrastructure architecture

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). EMF is based on the concept of classes which hold typed attributes and operations with typed parameters (Gerber and Raymond, 2003). It is a Java framework with coding generation facility for building tools and other applications based on a structured model (Mohamed *et al.*, 2007). As a Java framework from Eclipse Foundation (http://www.eclipse.org/org/), it can be used as part of Eclipse IDE or standalone. Using it as an Eclipse plug-in, EMF provides an editor for EMF models (*i.e.,* the meta-models) where the meta-model elements can be instantiated and edited according to the specified meta-model restrictions. When used standalone, its code generation facility is used to create a package with Java classes representing the meta-model[25], which can be imported into Java programs as any other

[25] EMF started with the Meta Object Facility (MOF) standard of OMG and was further developed within Java Community Process to produce the Java Metadata Interface Standard (https://www.jcp.org/en/jsr/detail?id=40). EMF and MOF can be fully mapped into one another.

library. The generated Java classes are used to programmatically instantiate an EMF model. EMF also provides a validation API (application programming interface), which takes an instantiated model and validate it according to the meta-model restrictions. A meta-model can be extended with additional validation operations, which are invoked during the validation. These operations supplements the structural validation based on the meta-model object-oriented structure generated from the meta-model.

EMF models (also known as Ecore models) are formed by the following main elements[26]:

- EClass: represents a class, with zero or more attributes and zero or more references.
- EAttribute: represents an attribute which has a name and a type.
- EReference: represents one end of an association between two classes. It has flags to indicate if it represents a containment or a reference class to which it points.
- EDataType: represents the type of an attribute, e.g., int, float or java.util.Date

Using these elements, the concrete syntax of theoretical structures are defined in the meta-model shown in Figure 20. The meta-model's classes (EClass) associations (EReference) define the evidence model structural restrictions, which are used to validate an instance of the meta-model. Both model concepts and relationships are represented as meta-model classes (EClass), and related to each other using normal Ecore associations (EReference). It was necessary to model theoretical structure relationships as classes (EClass) because we needed a type to each kind of relationship. In addition, this strategy was useful to explicit information associated with relationships (e.g., a description text) and to represent the relationships types' hierarchy. All relationships classes have references to a source and a target concept (e.g., *fromConcept* and *toValueConcept*, respectively). In a bidirectional way, concepts can also be a source of a relationship (e.g., *outCausalRelationship*) or a target (e.g., *inStructuralRelationship*).

For instance, *variable concepts* can only be *property of value concepts*. This is represented in the model by the triple *VariableConcept* — *PropertyOfRelationship* —

[26] There are many other elements forming the EMF meta-model used to create EMF models. Full meta-model details can be obtained at http://download.eclipse.org/modeling/emf/emf/javadoc/2.9.0/org/eclipse/emf/ecore/package-summary.html

ValueConcept. For this particular case, the following aspects are highlighted to understand how the model structure represents the theoretical structure restrictions. *PropertyOfRelationship* is a type of *StructuralRelationship*, which connects a *Concept* (*fromConcept* reference) to a *ValueConcept* (*toValueConcept* reference). *VariableConcept*, which is a type of *Concept*, can only be source of *PropertyOfRelationship* as defined in the *outPropertyRelationship* (*Moderator* can be source of *PropertyOfRelationship* and *ModerationRelationship* through the *outModerationsRelationship*). *ValueConcept*, in turn, can be target of any *StructuralRelationship* through the *inStructuralRelationship* reference, but it is not specified as a source of any relationship, since while *ContextualAspect* can be source of a relationship (e.g., *outCausalRelationship* reference) *Archetype* can not.

Mapping this example to a real instance, we reference to the Figure 12 (in Section 5.3.1) theoretical structure, particularly the *property of* relationship between Maintainability and Source Code. Maintainability is an *Effect* which is type of *VariableConcept* and Source Code is a *ContextualAspect* which is type of *ValueConcept*, both are connected through an instance of *PropertyOfRelationship*. EMF validates the relationship between Maintainability and Source Code is well formed by checking the associations (EReference) between these EClass and their cardinality. The following associations must exist to form the *property of* relationship between Maintainability and Source Code: (i) from the *PropertyOfRelationship* instance there should exist two associations, one *toValueConcept* (referencing Source Code) and one *fromConcept* (referencing Maintainability); (ii) from the *Effect* instance (*i.e.*, Maintainability) there should be one association *outPropertyRelationship* referencing the *PropertyOfRelationship* instance; and (iii) from the *ContextualAspect* instance (*i.e.*, Source Code) there should be one association *inStructuralRelationship* also referencing the *PropertyOfRelationship* instance. 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, which are structural 'holes' in the theoretical structure meta-model.

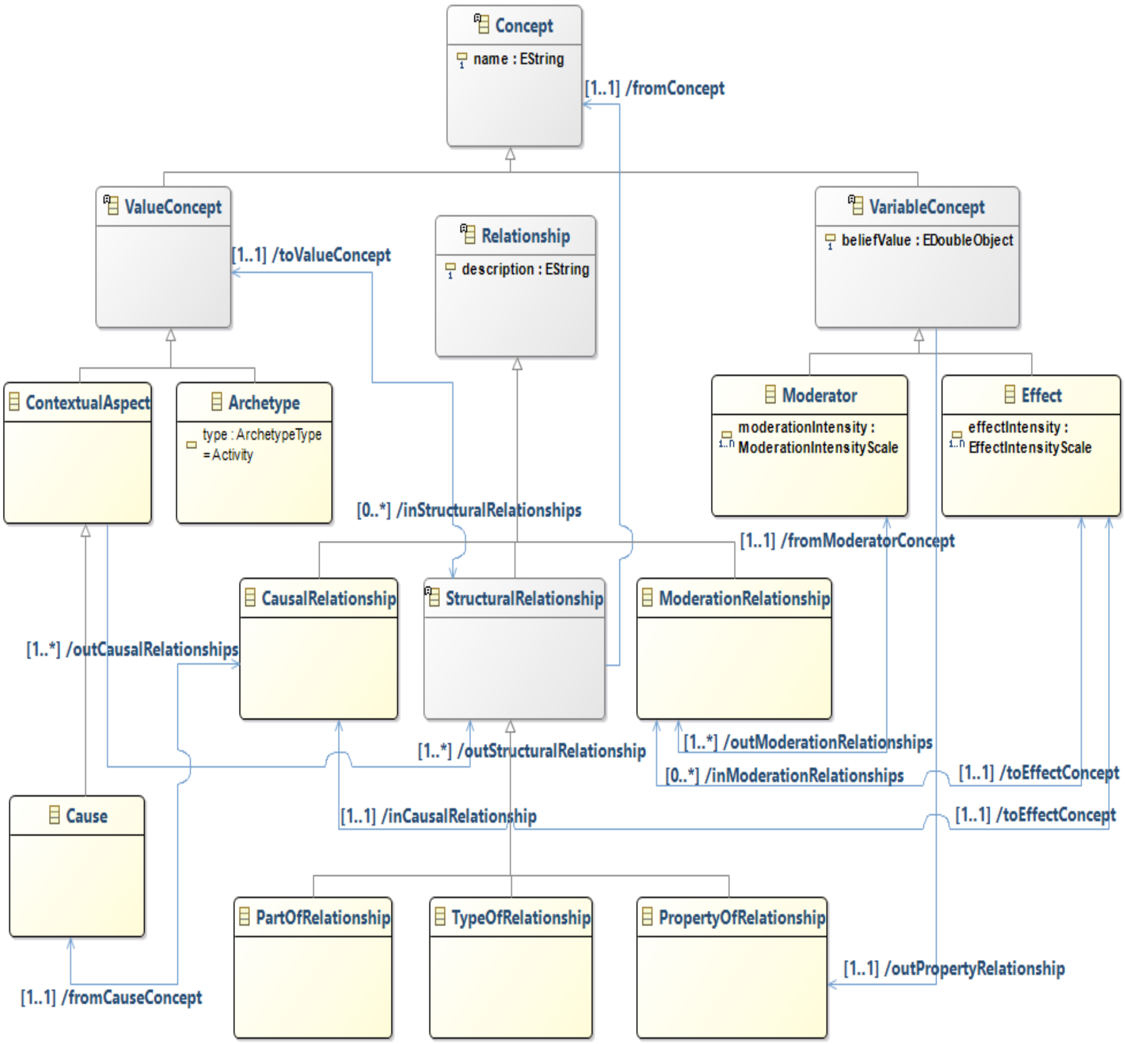

Concept
name : EString
[1..1] /fromConcept
ValueConcept
[1..1] /toValueConcept
Relationship
description : EString
VariableConcept
beliefValue : EDoubleObject
ContextualAspect
Archetype
type : ArchetypeType = Activity
Moderator
moderationIntensity : ModerationIntensityScale
Effect
effectIntensity : EffectIntensityScale
[0..*] /inStructuralRelationships
[1..1] /fromModeratorConcept
CausalRelationship
StructuralRelationship
ModerationRelationship
[1..*] /outCausalRelationships
[1..*] /outModerationRelationships
[1..*] /outStructuralRelationship
[0..*] /inModerationRelationships
[1..1] /toEffectConcept
Cause
[1..1] /inCausalRelationship
[1..1] /toEffectConcept
PartOfRelationship
TypeOfRelationship
PropertyOfRelationship
[1..1] /outPropertyRelationship
[1..1] /fromCauseConcept

Figure 20 – Evidence representation meta-model

7.3.2 Navigational structure and supporting facilities

The Evidence Factory facilities and navigational definitions are centered around the currently implemented use cases: (i) glossary maintenance, (ii) evidence search and aggregation, and (iii) synthesis creation. Figure 21 shows the navigational structure in which is possible to identify the application main screens and functionalities.

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 look for keywords in the whole evidence model (concepts, scope description and explanation detail) and is ordered according to the number of terms occurrence in it. From the evidence result list, it is possible to select which one will be aggregated using the facilities 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.

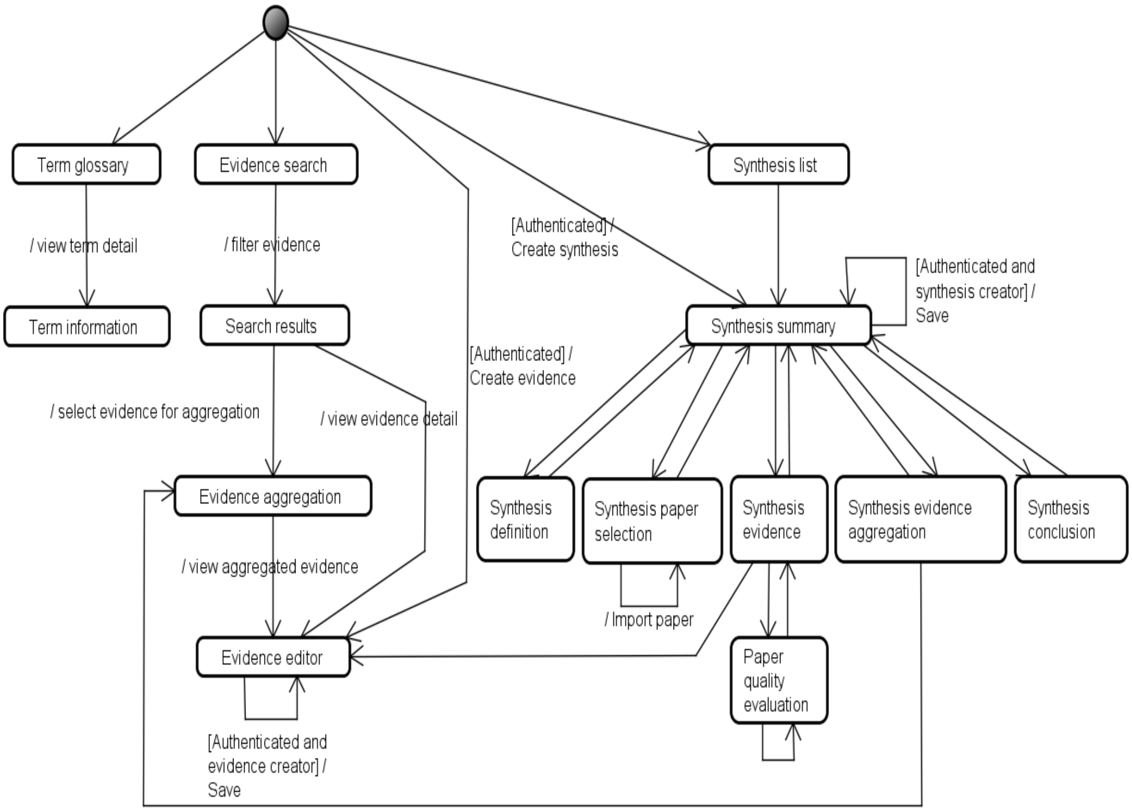


Figure 21 – Infrastructure navigational structure (the start activity symbol represents accessible menu items and the final symbol was omitted)

7.3.3 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.

Following the structured aggregation definitions (Section 5.3), the algorithm at first only considers the evidence structural relationships, ignoring causal and moderation relationships. 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 (Figure 22b). The removal takes the specified node off the tree and places its children under the parent of the removed node (Figure 22c). 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 (Figure 22d).

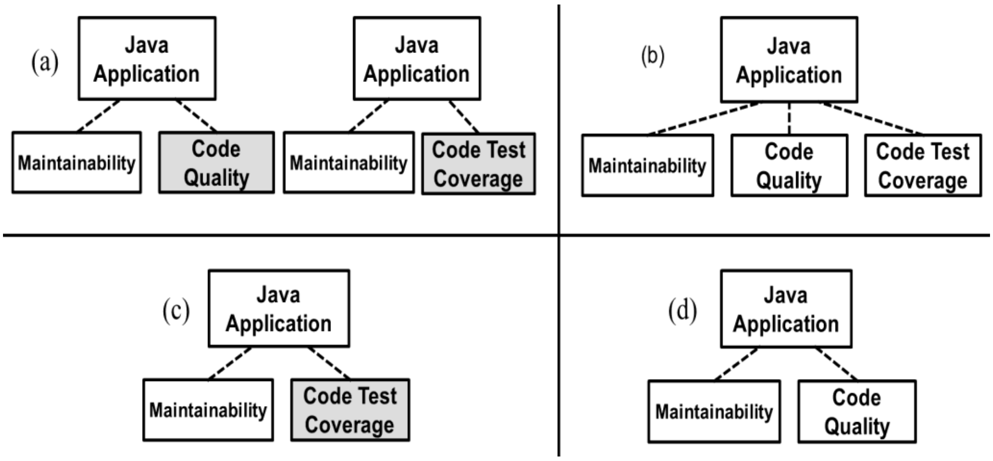


Figure 22 – Possible resolution for the conflicts on evidence fragment (a): add both concepts (b), remove one (c), or join them (d)

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 aggregated evidence model only with the structural relationships. After that, the algorithm continue its computations with the aggregation of *causal* and *moderation* relationships. For each influence relationship in either original evidence, if both of the related concepts (*e.g.,* cause and effect) are present in the resulting evidence, a new relationship is placed between them. The resulting belief value for the *causal* or *moderation* relationship is then calculated using the D-S theory described earlier.

7.4 Tool's facilities

As described in Figure 21, the Evidence Factory primary use is associated with evidence search and evidence modeling (usually within a synthesis). The tool glossary is also another important feature controlling the terms used in evidence creation, search and aggregation. Screenshots shown in this section use real research syntheses which are detailed in next chapter.

7.4.1 Evidence search

Search is performed with keywords (Figure 23), which are used to find evidence based both on evidence scope description and evidence concept names and description.

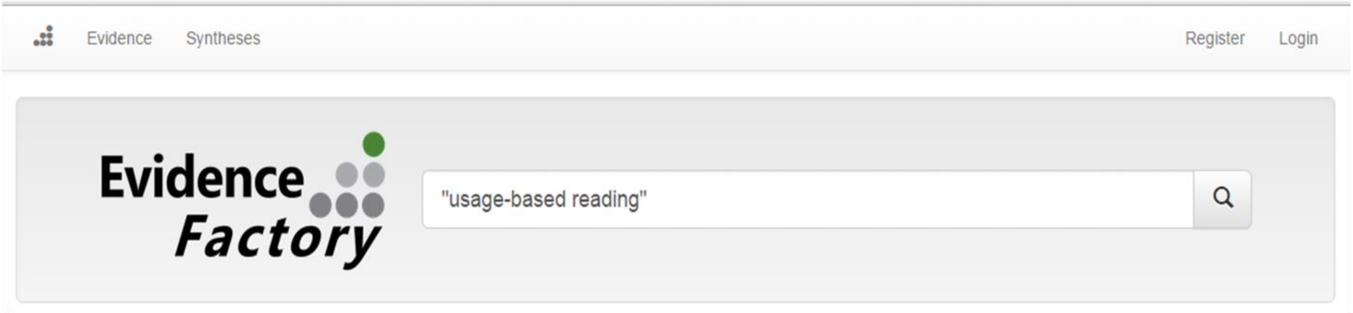
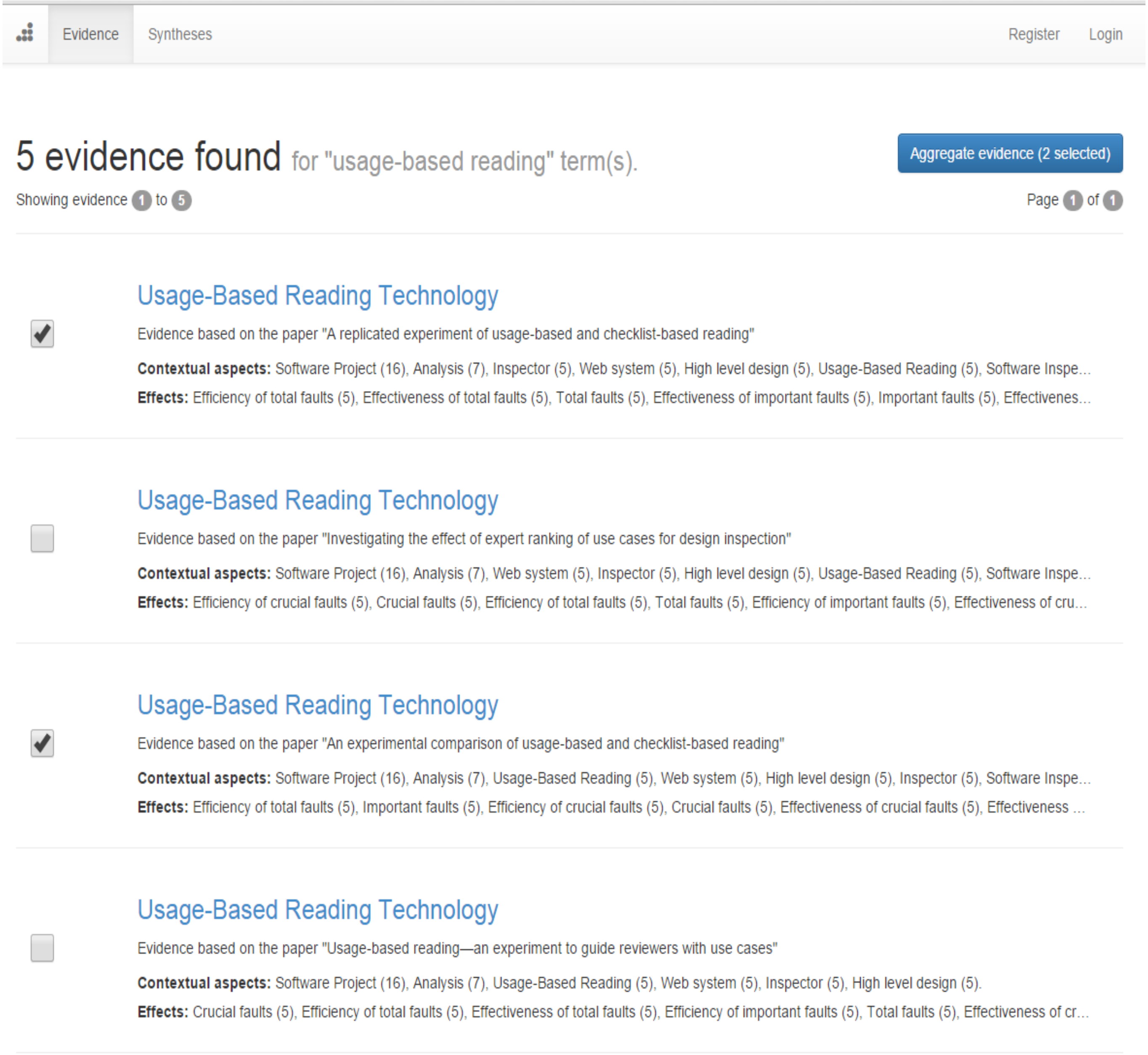


Figure 23 – Evidence search form

The search result lists the evidence found (Figure 24). Each evidence has a link title which is formed by the causal concept name plus the archetype which it inherits from – in the case of Figure 24 example, all evidence found are related to the 'usage-based reading' cause which was modeled as a type of technology. The links points to the evidence detail which is the evidence model itself (Figure 25). Additional information provided in result list include the evidence description and the enumeration of contextual aspects, effects and moderation concepts. The numbers beside each concept indicates how many evidence it was used.

Apart the evidence listing itself, evidence can be selected for aggregation. Thus, based on the terms used in the search string and the information provided for each evidence researchers would be able to identify and select evidence for aggregation. Researchers can also go back and forth from the search result and evidence model if they need detailed information. Evidence aggregation as part of the evidence search is provided only for opportunistic and exploratory aggregation to see what evidence 'say together' using the aggregation procedure based on the D-S theory. For throughout research syntheses it is recommended to follow SSM process as will be shown in a later section.



5 evidence found for "usage-based reading" term(s).

Showing evidence 1 to 5

Page 1 of 1

☑ Usage-Based Reading Technology

Evidence based on the paper "A replicated experiment of usage-based and checklist-based reading"

Contextual aspects: Software Project (16), Analysis (7), Inspector (5), Web system (5), High level design (5), Usage-Based Reading (5), Software Inspe...

Effects: Efficiency of total faults (5), Effectiveness of total faults (5), Total faults (5), Effectiveness of important faults (5), Important faults (5), Effectivenes...

□ Usage-Based Reading Technology

Evidence based on the paper "Investigating the effect of expert ranking of use cases for design inspection"

Contextual aspects: Software Project (16), Analysis (7), Web system (5), Inspector (5), High level design (5), Usage-Based Reading (5), Software Inspe...

Effects: Efficiency of crucial faults (5), Crucial faults (5), Efficiency of total faults (5), Total faults (5), Efficiency of important faults (5), Effectiveness of cru...

☑ Usage-Based Reading Technology

Evidence based on the paper "An experimental comparison of usage-based and checklist-based reading"

Contextual aspects: Software Project (16), Analysis (7), Usage-Based Reading (5), Web system (5), High level design (5), Inspector (5), Software Inspe...

Effects: Efficiency of total faults (5), Important faults (5), Efficiency of crucial faults (5), Crucial faults (5), Effectiveness of crucial faults (5), Effectiveness ...

□ Usage-Based Reading Technology

Evidence based on the paper "Usage-based reading—an experiment to guide reviewers with use cases"

Contextual aspects: Software Project (16), Analysis (7), Usage-Based Reading (5), Web system (5), Inspector (5), High level design (5).

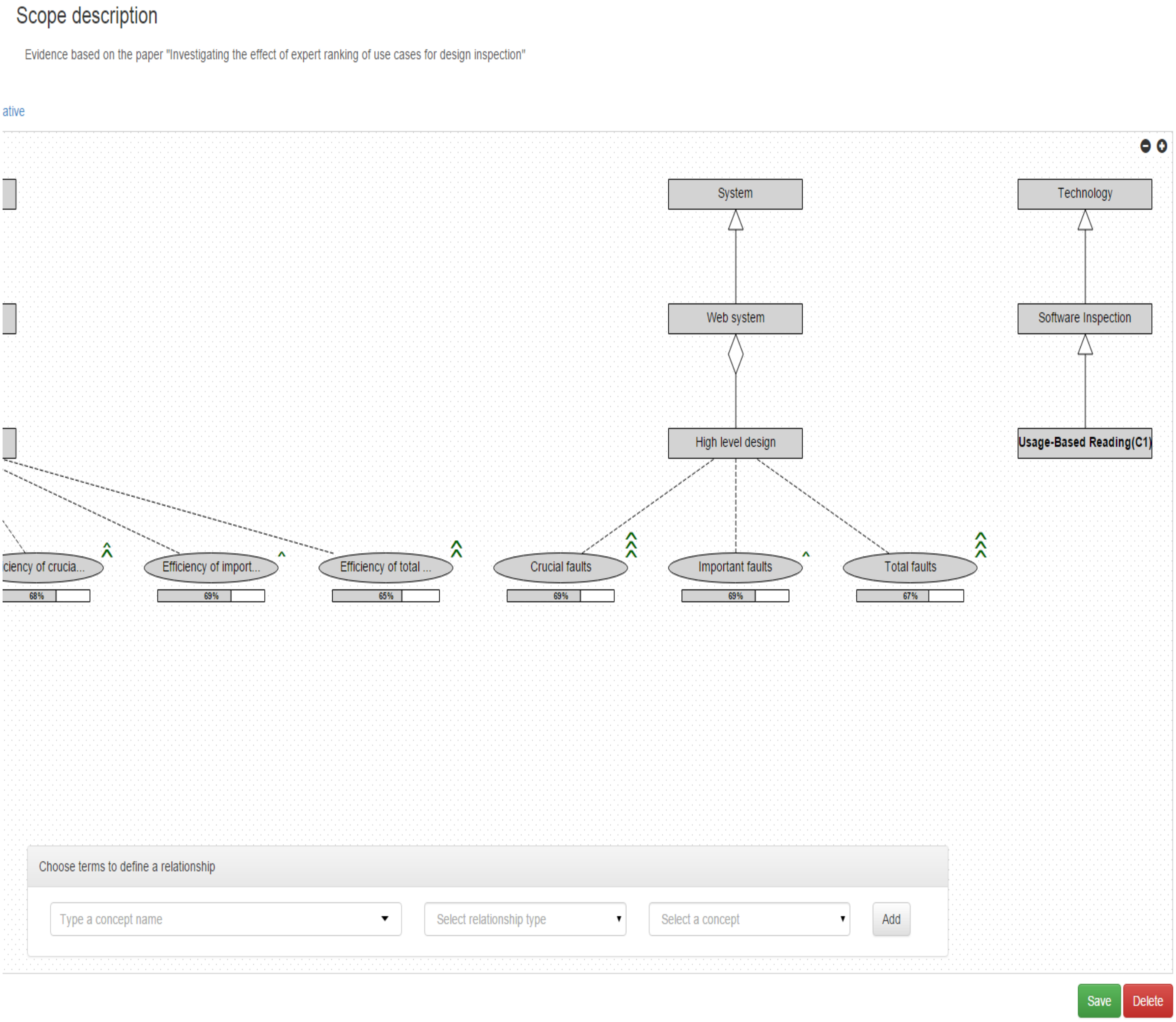
Effects: Crucial faults (5), Efficiency of total faults (5), Effectiveness of total faults (5), Efficiency of important faults (5), Total faults (5), Effectiveness of cr...

Figure 24 – Evidence search result

7.4.2 Evidence modeling

The evidence editor is used both for modeling evidence and to show evidence detail when coming from evidence search (Figure 25). To simplify the implementation in a web environment, concepts and relationships are added through a simple form at the bottom of the diagram where the researcher input a relationship triple <concept><relationship type><concept> (Figure 26). When adding a relationship, the first concept is necessarily a new concept and the second one already included in the model. As the vocabulary for evidence model concepts is controlled, the tool provides an auto-completion feature assisting the researcher in selecting appropriate concepts.

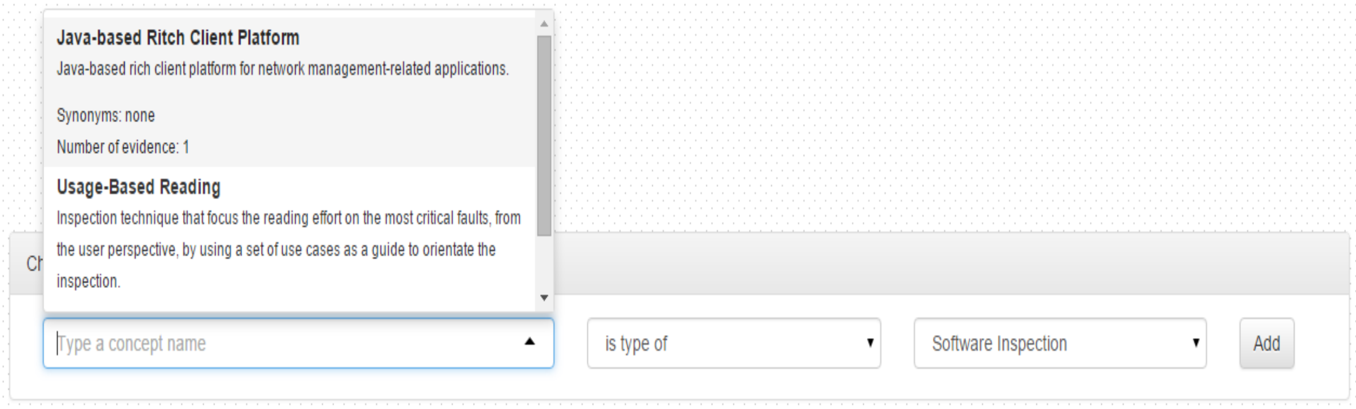
Moreover, the auto-completion feature shows the detailed concept description and the number of evidence into which the term was used. Other actions are provided through a context menu (Figure 27), such as concept deletion, based on the concept type (archetype, contextual aspect and properties). After editing the evidence model evidence must be saved. Currently, there is no undo/redo mechanisms available.



Scope description

Evidence based on the paper "Investigating the effect of expert ranking of use cases for design inspection"

Figure 25 – Evidence model editor (partial screenshot)



Java-based Ritch Client Platform

Java-based rich client platform for network management-related applications.

Synonyms: none

Number of evidence: 1

Usage-Based Reading

Inspection technique that focus the reading effort on the most critical faults, from the user perspective, by using a set of use cases as a guide to orientate the inspection.

Ch

Type a concept name ▲ | is type of ▾ | Software Inspection ▾ | Add

Figure 26 – Adding a relationship to the diagram

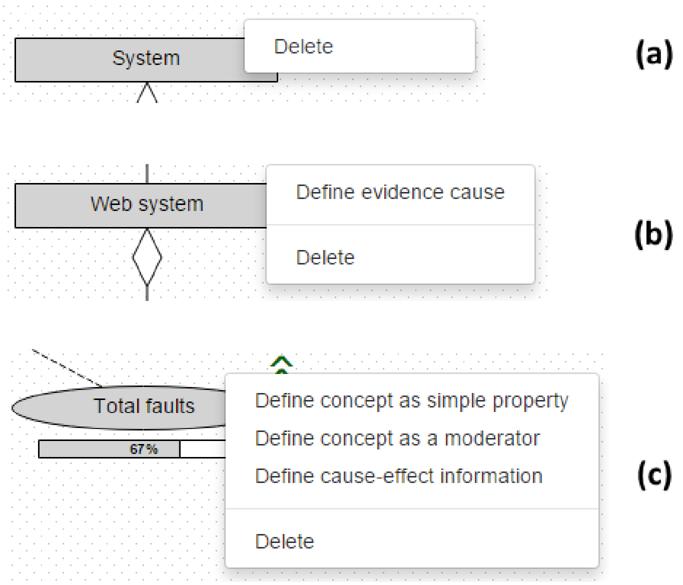
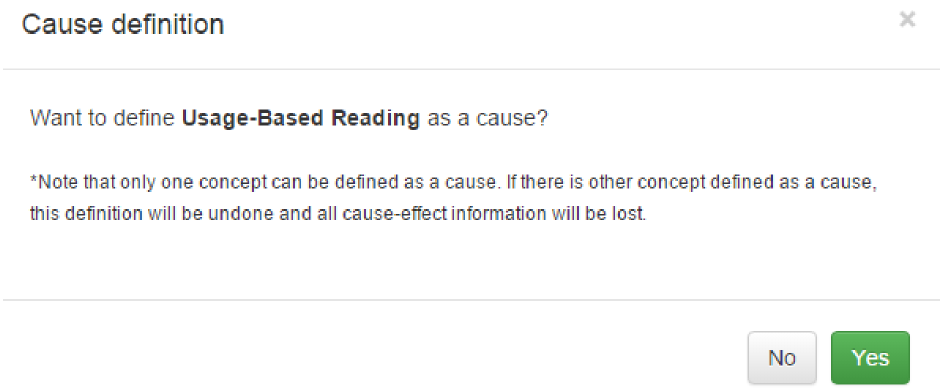

System
Delete
(a)
Web system
Define evidence cause
Delete
(b)
Total faults
67 %
Define concept as simple property
Define concept as a moderator
Define cause-effect information
Delete
(c)

Figure 27 – Available actions in the evidence model editor: (a) archetypes, (b) contextual aspects, and (c) properties

Apart from inclusion of concepts and their relationships, there are additional definitions related to the evidence modeling that the evidence editor support. It is necessary to define which concept is the evidence cause as shown in Figure 28. Once the cause is defined all effects are automatically related to it so that it is not necessary to manually create each cause-effect relationship. In the case of effects, the researcher must input the effect intensity and provide an explanation describing why or how it was observed (Figure 29). It is interesting to notice that the belief value field is disabled as it is estimated based on the study type and the quality assessment (when evidence is not part of a research synthesis the belief value can be manually informed). Using the discounting operation from D-S theory a *p-value* can be informed adjusting the belief value. Moderators have similar definitions of effects as shown in Figure 30, but it requires the indication of which effects are going to be moderated.

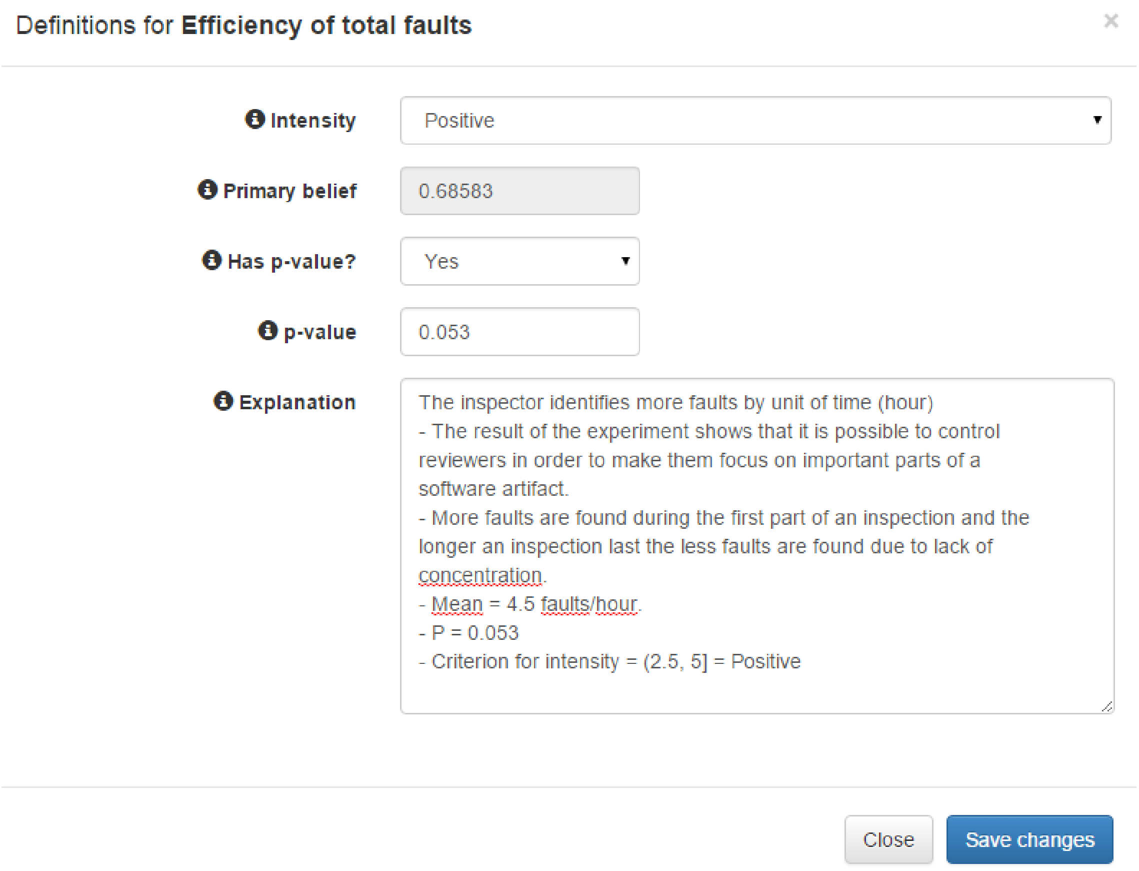


Cause definition

Want to define **Usage-Based Reading** as a cause?

*Note that only one concept can be defined as a cause. If there is other concept defined as a cause, this definition will be undone and all cause-effect information will be lost.

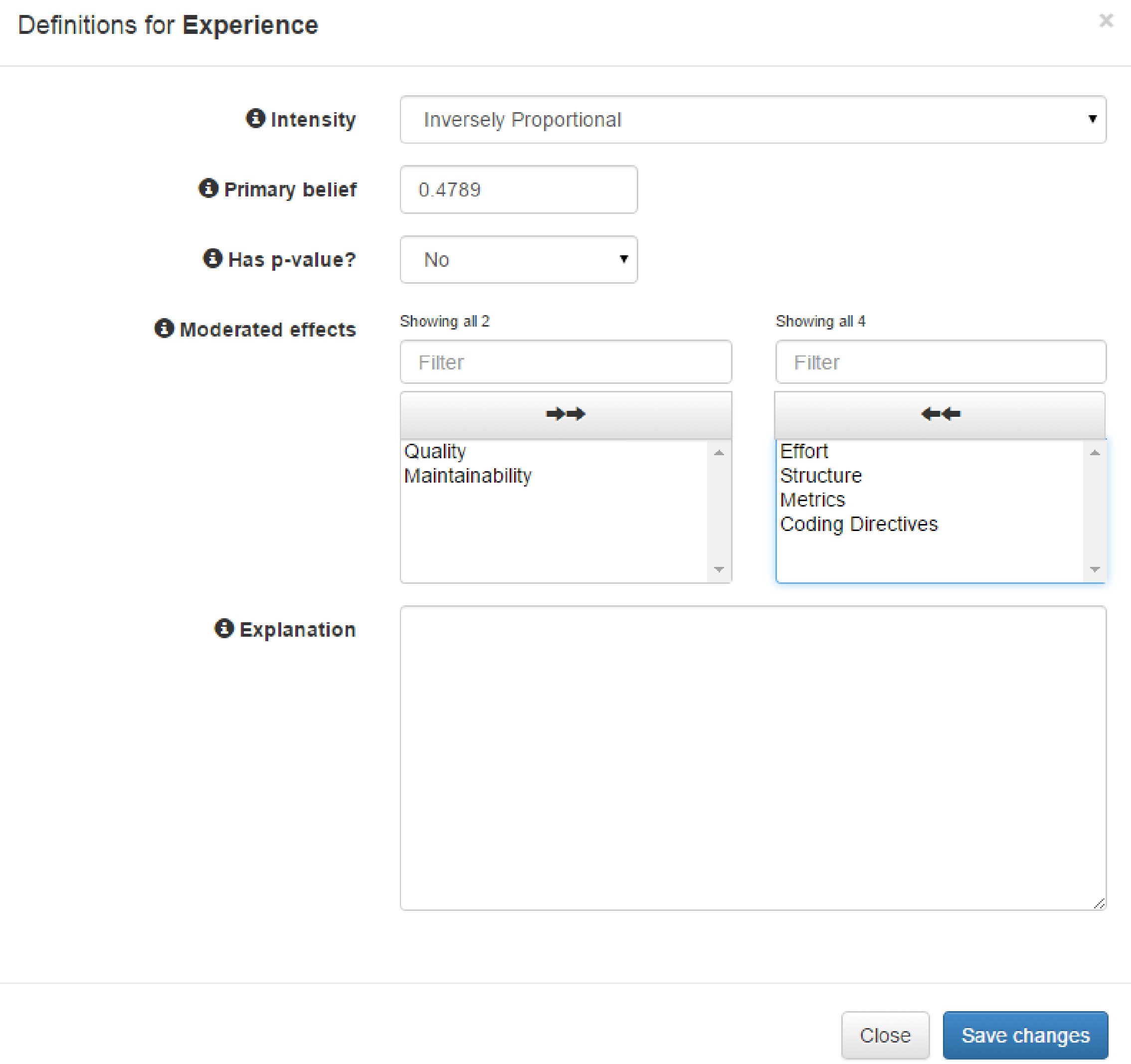
Figure 28 – Cause definition of an evidence model



Definitions for **Efficiency of total faults**

Intensity	Positive
Primary belief	0.68583
Has p-value?	Yes
p-value	0.053
Explanation	The inspector identifies more faults by unit of time (hour) - The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact. - More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration. - Mean = 4.5 faults/hour. - P = 0.053 - Criterion for intensity = (2.5, 5] = Positive

Figure 29 – Definitions for effects



Definitions for **Experience**

Intensity: Inversely Proportional

Primary belief: 0.4789

Has p-value?: No

Moderated effects:

Showing all 2
Quality
Maintainability

Showing all 4
Effort
Structure
Metrics
Coding Directives

Explanation:

Figure 30 – Definitions form for moderators

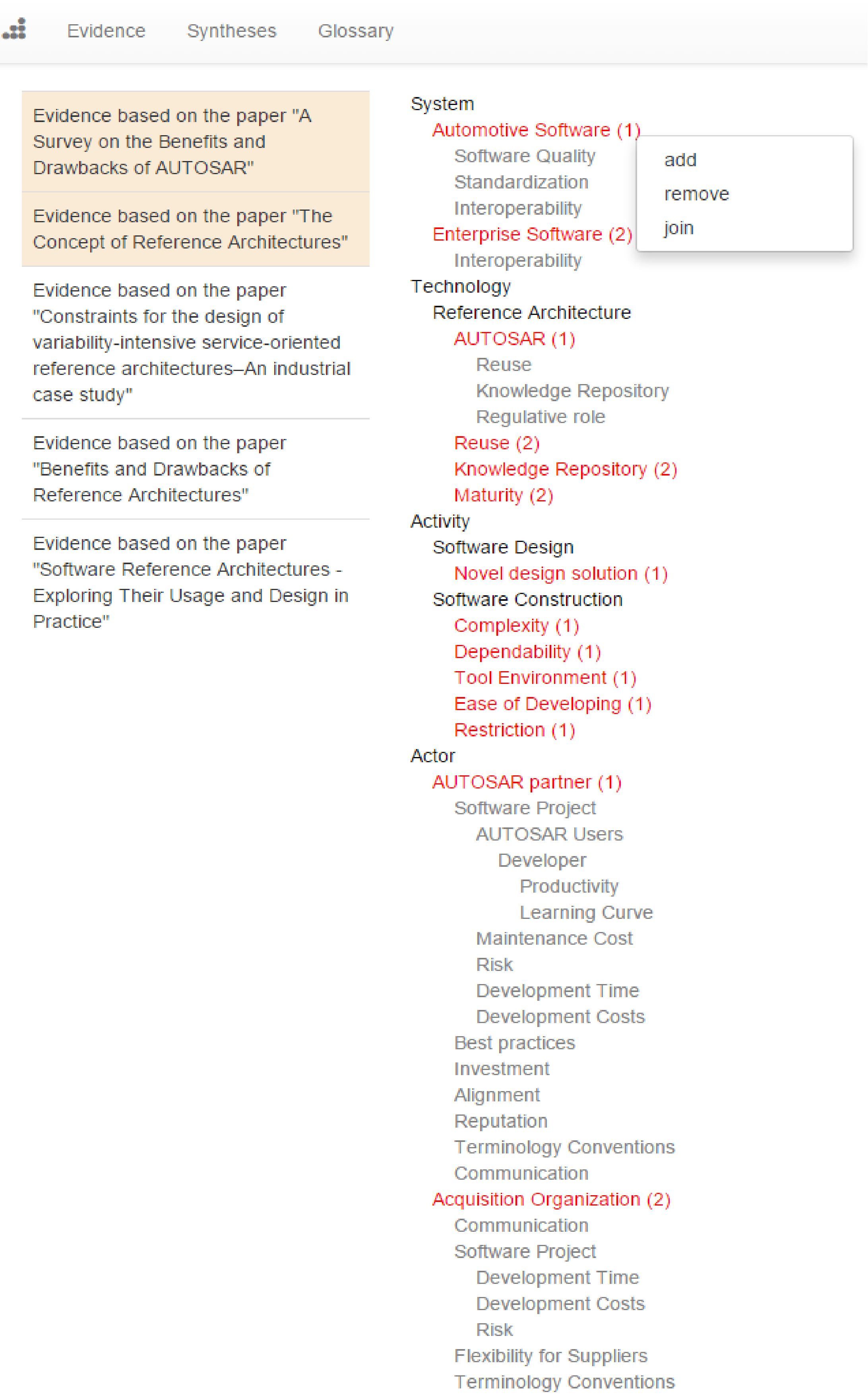
7.4.3 Evidence aggregation

Using the mechanism for conflict resolution described in Section 7.3.3, evidence aggregation interface uses a tree structure[27] to display the differences between evidence. Aggregation is performed with two evidence at a time, where the result of the previous combination is aggregated with another evidence. As shown in Figure 31, the tree structure contains terms in red, black and gray colors. Black terms indicates that the models are compatible up to that level in the tree. Red terms indicates a conflict.

[27] Although not described as a tree in previous chapters, it is not difficult to see the mapping between theoretical structures and trees. In fact, it is a forest with four roots, one for each archetype.

The number besides the term is a reference to which evidence the term comes from. It follows the order of evidence listed in the left. Conflicts occurs in the same level of each evidence model (*i.e.*, tree) and there are three available actions to resolve them as previously discussed (and is shown in Figure 31). Gray terms are below the terms which are conflicting. The tool is only able to evaluate lower levels in the tree when the levels above are resolved. As soon as the tool finds a conflict it stops the compatibility evaluation and marks the terms that were not evaluated with gray color.

After all conflicts are resolved, the evidence editor shows the aggregated evidence with the respective combined belief values.



Evidence based on the paper "A Survey on the Benefits and Drawbacks of AUTOSAR"

Evidence based on the paper "The Concept of Reference Architectures"

Evidence based on the paper "Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study"

Evidence based on the paper "Benefits and Drawbacks of Reference Architectures"

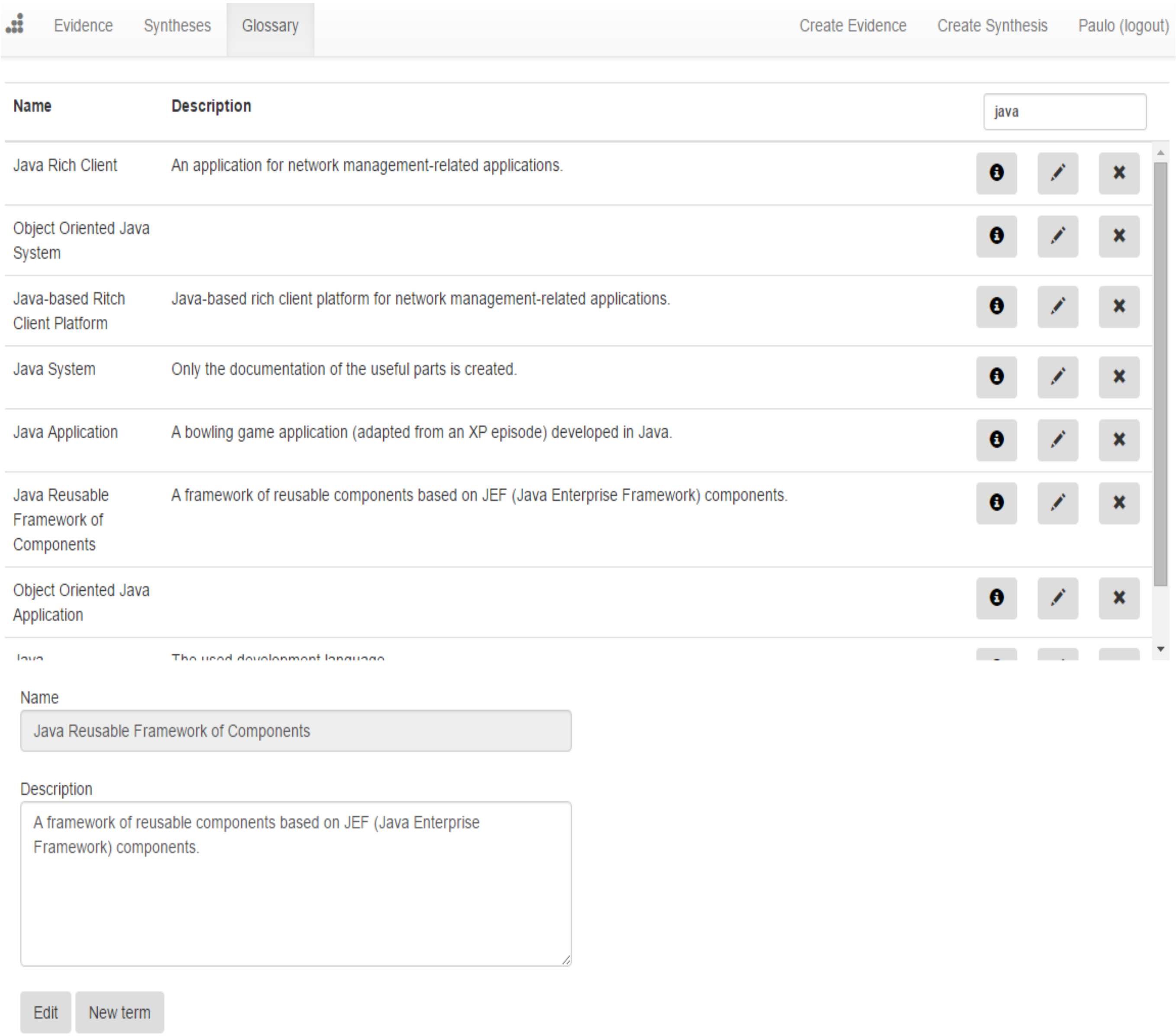
Evidence based on the paper "Software Reference Architectures - Exploring Their Usage and Design in Practice"

- System
  - Automotive Software (1)
    - Software Quality
    - Standardization
    - Interoperability
  - Enterprise Software (2)
    - Interoperability
- Technology
  - Reference Architecture
    - AUTOSAR (1)
      - Reuse
      - Knowledge Repository
      - Regulative role
    - Reuse (2)
    - Knowledge Repository (2)
    - Maturity (2)
- Activity
  - Software Design
    - Novel design solution (1)
  - Software Construction
    - Complexity (1)
    - Dependability (1)
    - Tool Environment (1)
    - Ease of Developing (1)
    - Restriction (1)
- Actor
  - AUTOSAR partner (1)
    - Software Project
      - AUTOSAR Users
        - Developer
          - Productivity
          - Learning Curve
      - Maintenance Cost
      - Risk
      - Development Time
      - Development Costs
    - Best practices
    - Investment
    - Alignment
    - Reputation
    - Terminology Conventions
    - Communication
  - Acquisition Organization (2)
    - Communication
    - Software Project
      - Development Time
      - Development Costs
      - Risk
    - Flexibility for Suppliers
    - Terminology Conventions

Figure 31 - Evidence conflict resolution for aggregation

7.4.4 Glossary of terms

Terms definition is straightforward (Figure 32). They are listed in a table where there are actions to add, edit, remove or detail a specific term. The term details (Figure 33) shows its definitions, lists evidence in which the term is used and allows the definition of synonyms for terms. Synonyms information are used to resolve conflicts in evidence aggregation.

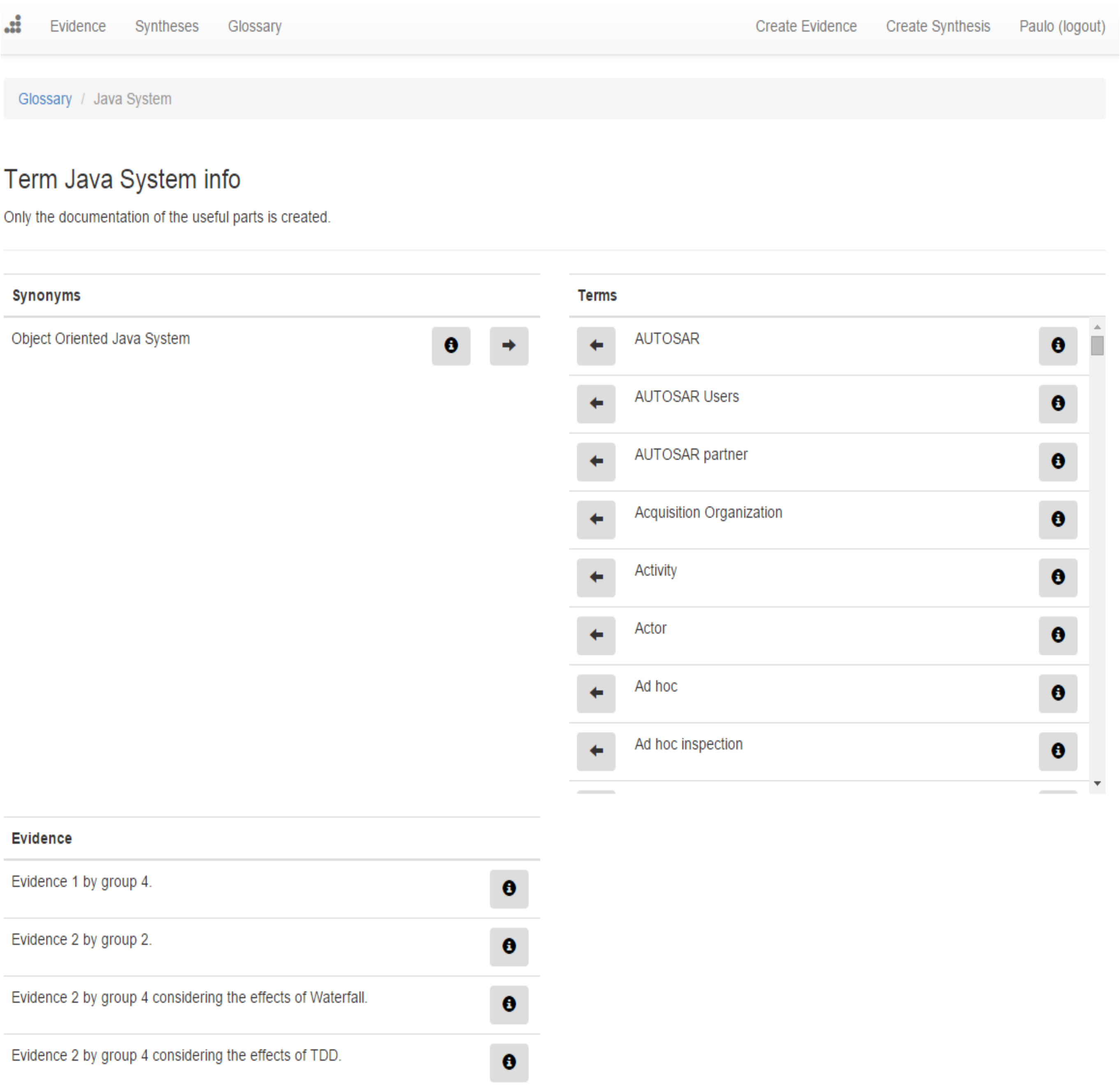


Name	Description
Java Rich Client	An application for network management-related applications.
Object Oriented Java System	
Java-based Ritch Client Platform	Java-based rich client platform for network management-related applications.
Java System	Only the documentation of the useful parts is created.
Java Application	A bowling game application (adapted from an XP episode) developed in Java.
Java Reusable Framework of Components	A framework of reusable components based on JEF (Java Enterprise Framework) components.
Object Oriented Java Application	
Java	The used development language

Name: Java Reusable Framework of Components

Description: A framework of reusable components based on JEF (Java Enterprise Framework) components.

Figure 32 – Terms listing



Term Java System info

Only the documentation of the useful parts is created.

Synonyms

Object Oriented Java System

Terms

AUTOSAR

AUTOSAR Users

AUTOSAR partner

Acquisition Organization

Activity

Actor

Ad hoc

Ad hoc inspection

Evidence

Evidence 1 by group 4.

Evidence 2 by group 2.

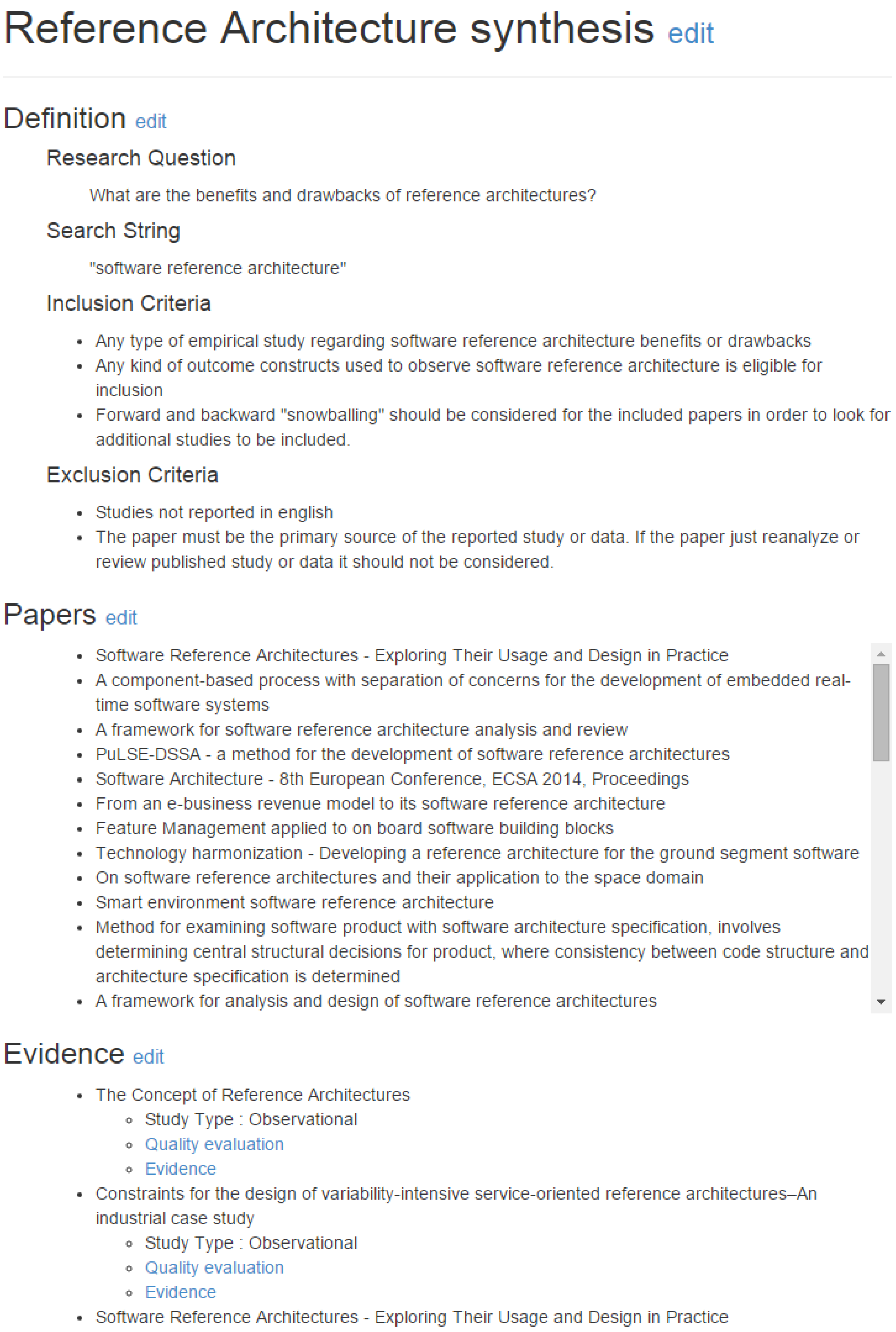
Evidence 2 by group 4 considering the effects of Waterfall.

Evidence 2 by group 4 considering the effects of TDD.

Figure 33 – Term details

7.4.5 SSM execution support

The first page of a research synthesis shows a resume of all SSM steps to execute or already executed (Figure 34). Each step has links to the specific pages associated with the respective step, including access to the quality assessment, evidence models, selected papers and the aggregated evidence model.



Reference Architecture synthesis edit

Definition edit

Research Question

What are the benefits and drawbacks of reference architectures?

Search String

"software reference architecture"

Inclusion Criteria

- Any type of empirical study regarding software reference architecture benefits or drawbacks
- Any kind of outcome constructs used to observe software reference architecture is eligible for inclusion
- Forward and backward "snowballing" should be considered for the included papers in order to look for additional studies to be included.

Exclusion Criteria

- Studies not reported in english
- The paper must be the primary source of the reported study or data. If the paper just reanalyze or review published study or data it should not be considered.

Papers edit

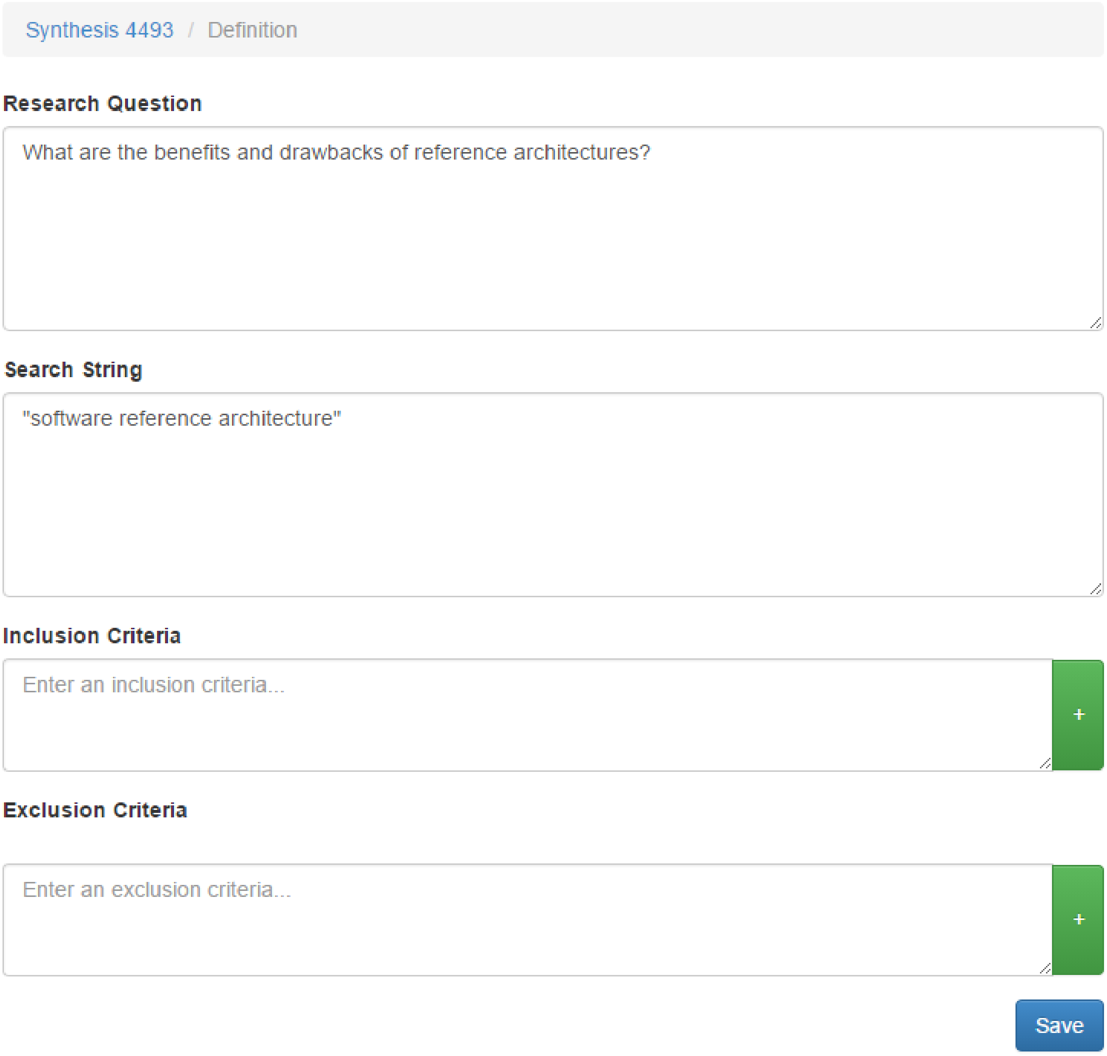
- Software Reference Architectures - Exploring Their Usage and Design in Practice
- A component-based process with separation of concerns for the development of embedded real-time software systems
- A framework for software reference architecture analysis and review
- PuLSE-DSSA - a method for the development of software reference architectures
- Software Architecture - 8th European Conference, ECSA 2014, Proceedings
- From an e-business revenue model to its software reference architecture
- Feature Management applied to on board software building blocks
- Technology harmonization - Developing a reference architecture for the ground segment software
- On software reference architectures and their application to the space domain
- Smart environment software reference architecture
- Method for examining software product with software architecture specification, involves determining central structural decisions for product, where consistency between code structure and architecture specification is determined
- A framework for analysis and design of software reference architectures

Evidence edit

- The Concept of Reference Architectures
  - Study Type : Observational
  - Quality evaluation
  - Evidence
- Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study
  - Study Type : Observational
  - Quality evaluation
  - Evidence
- Software Reference Architectures - Exploring Their Usage and Design in Practice

Figure 34 - Evidence synthesis resume (partial screenshot)

The page associated with the study definition just register information about the research synthesis as defined in the SSM method (Figure 35). It has a simple form where the research must define a research question, a search string and explicit the inclusion/exclusion criteria.



Synthesis 4493 / Definition

Research Question

What are the benefits and drawbacks of reference architectures?

Search String

"software reference architecture"

Inclusion Criteria

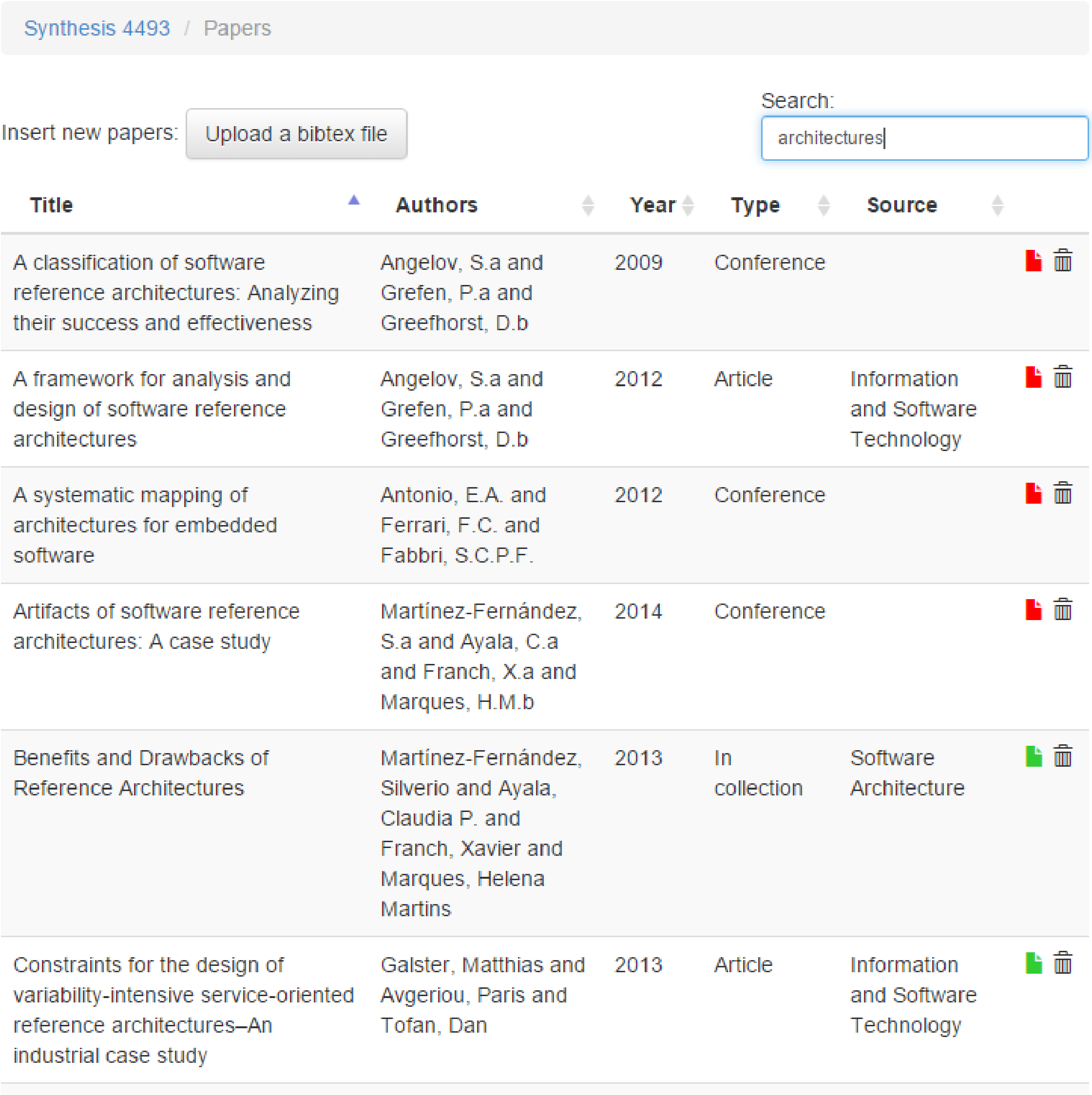
Enter an inclusion criteria...

Exclusion Criteria

Enter an exclusion criteria...

Figure 35 - Research synthesis definition page

In SSM, the next step is related to the papers selection. Evidence Factory allows importing the paper list by uploading a file in the Bibtex format. At this point, the researcher can also indicate what papers will be included in the synthesis, which are the ones marked in green (Figure 36). Only included papers will be considered for the next SSM steps.



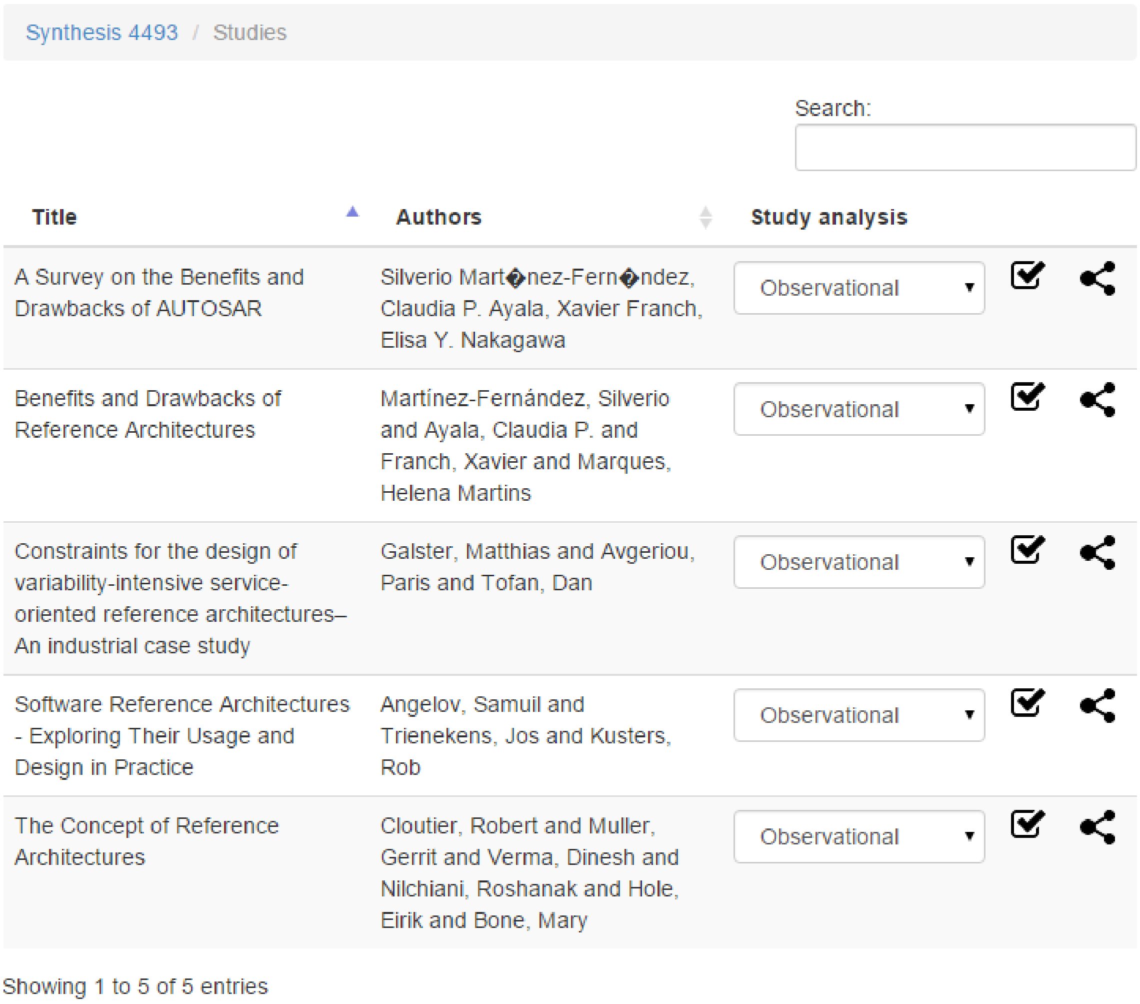
Insert new papers: Upload a bibtex file

Search: architectures

Title	Authors	Year	Type	Source	
A classification of software reference architectures: Analyzing their success and effectiveness	Angelov, S.a and Grefen, P.a and Greefhorst, D.b	2009	Conference		🟥 🗑
A framework for analysis and design of software reference architectures	Angelov, S.a and Grefen, P.a and Greefhorst, D.b	2012	Article	Information and Software Technology	🟥 🗑
A systematic mapping of architectures for embedded software	Antonio, E.A. and Ferrari, F.C. and Fabbri, S.C.P.F.	2012	Conference		🟥 🗑
Artifacts of software reference architectures: A case study	Martínez-Fernández, S.a and Ayala, C.a and Franch, X.a and Marques, H.M.b	2014	Conference		🟥 🗑
Benefits and Drawbacks of Reference Architectures	Martínez-Fernández, Silverio and Ayala, Claudia P. and Franch, Xavier and Marques, Helena Martins	2013	In collection	Software Architecture	🟩 🗑
Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study	Galster, Matthias and Avgeriou, Paris and Tofan, Dan	2013	Article	Information and Software Technology	🟩 🗑

Figure 36 – Research synthesis paper selection

Regarding the third and fourth steps of SSM, Evidence Factory requires the definition of the study type, the creation of the evidence model and the answering of the quality assessment questionnaire (Figure 37). The two quality questionnaires used in SSM are entirely embedded in the tool, where the researcher is able to answer them (Figure 38). It is also important to notice that as soon as it is created, the evidence can be searched by any other user and is available to be used in other aggregations. Evidence models for the research synthesis are created using the evidence editor shown previously.



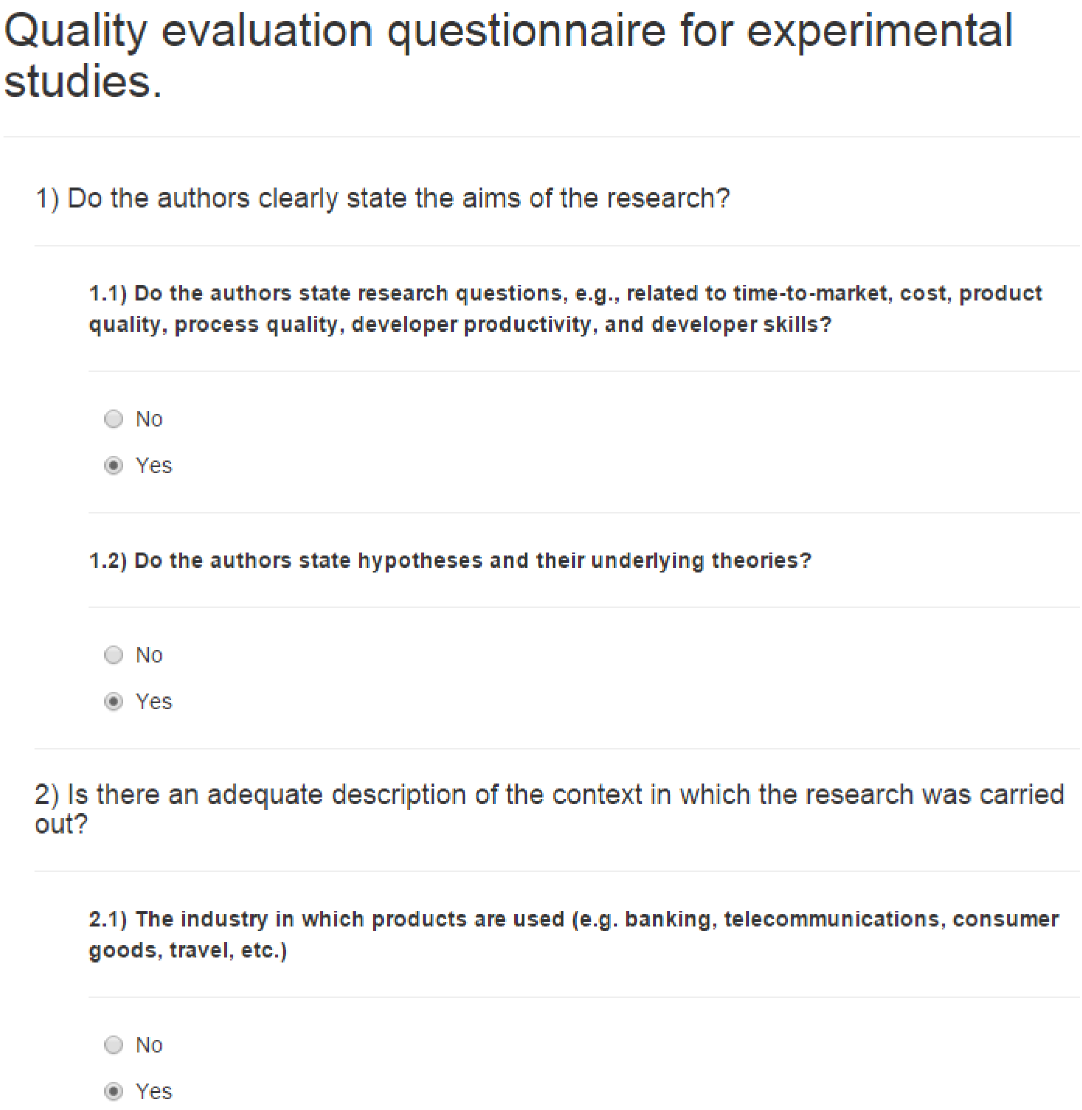
Synthesis 4493 / Studies

Search:

Title	Authors	Study analysis		
A Survey on the Benefits and Drawbacks of AUTOSAR	Silverio Mart�nez-Fern�ndez, Claudia P. Ayala, Xavier Franch, Elisa Y. Nakagawa	Observational ▾	☑	⋖
Benefits and Drawbacks of Reference Architectures	Martínez-Fernández, Silverio and Ayala, Claudia P. and Franch, Xavier and Marques, Helena Martins	Observational ▾	☑	⋖
Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study	Galster, Matthias and Avgeriou, Paris and Tofan, Dan	Observational ▾	☑	⋖
Software Reference Architectures - Exploring Their Usage and Design in Practice	Angelov, Samuil and Trienekens, Jos and Kusters, Rob	Observational ▾	☑	⋖
The Concept of Reference Architectures	Cloutier, Robert and Muller, Gerrit and Verma, Dinesh and Nilchiani, Roshanak and Hole, Eirik and Bone, Mary	Observational ▾	☑	⋖

Showing 1 to 5 of 5 entries

Figure 37 – Study type indication and links for quality assessment and evidence model creation



Quality evaluation questionnaire for experimental studies.

1) Do the authors clearly state the aims of the research?

1.1) Do the authors state research questions, e.g., related to time-to-market, cost, product quality, process quality, developer productivity, and developer skills?

○ No

● Yes

1.2) Do the authors state hypotheses and their underlying theories?

○ No

● Yes

2) Is there an adequate description of the context in which the research was carried out?

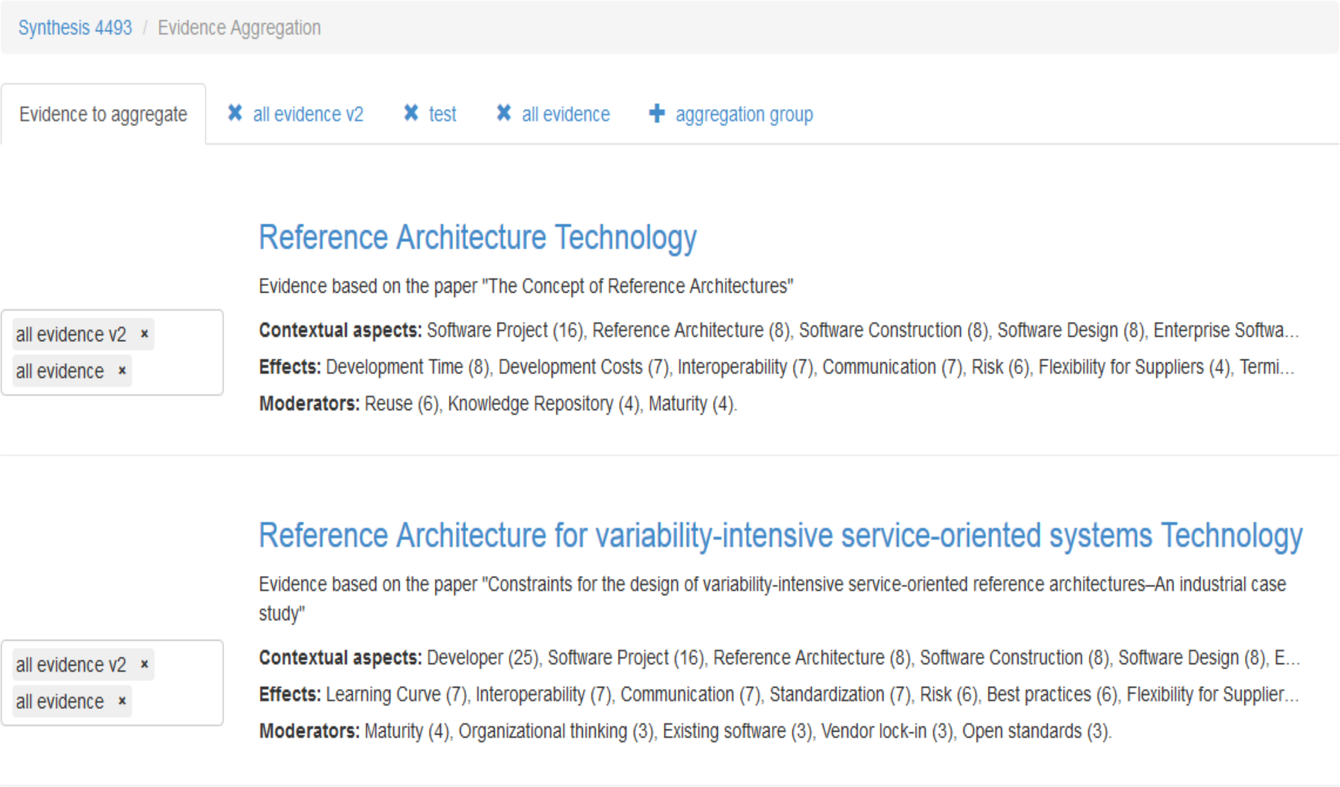
2.1) The industry in which products are used (e.g. banking, telecommunications, consumer goods, travel, etc.)

○ No

● Yes

Figure 38 - Studies quality assessment (partial screenshot)

The evidence aggregation is one of the main system parts (Figure 39). At this moment, the researcher can group compatible or at least most similar evidence. This is performed by creating groups and then assigning one or more group categories to the considered evidence. In the example of Figure 39, all evidence were known to be very similar, so only one group named ‘all evidence’ was created (a ‘test’ group was also created with exploratory purposes). 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 conflict resolution mechanism shown earlier can be used to adjust their compatibility. 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 different colors. The red color indicate the aggregation has not yet been done, green color indicate evidence fully matched or yellow to indicate that the informed resolutions are not sufficient to match evidence (Figure 40). After the aggregation is completed, the generated aggregated evidence is rendered as any other evidence model in the evidence editor and is also available for other users through the evidence search. The aggregated model also has the details of D-S theory computation for all effects and moderators (Figure 41).



Reference Architecture Technology

Evidence based on the paper "The Concept of Reference Architectures"

Contextual aspects: Software Project (16), Reference Architecture (8), Software Construction (8), Software Design (8), Enterprise Softwa...

Effects: Development Time (8), Development Costs (7), Interoperability (7), Communication (7), Risk (6), Flexibility for Suppliers (4), Termi...

Moderators: Reuse (6), Knowledge Repository (4), Maturity (4).

Reference Architecture for variability-intensive service-oriented systems Technology

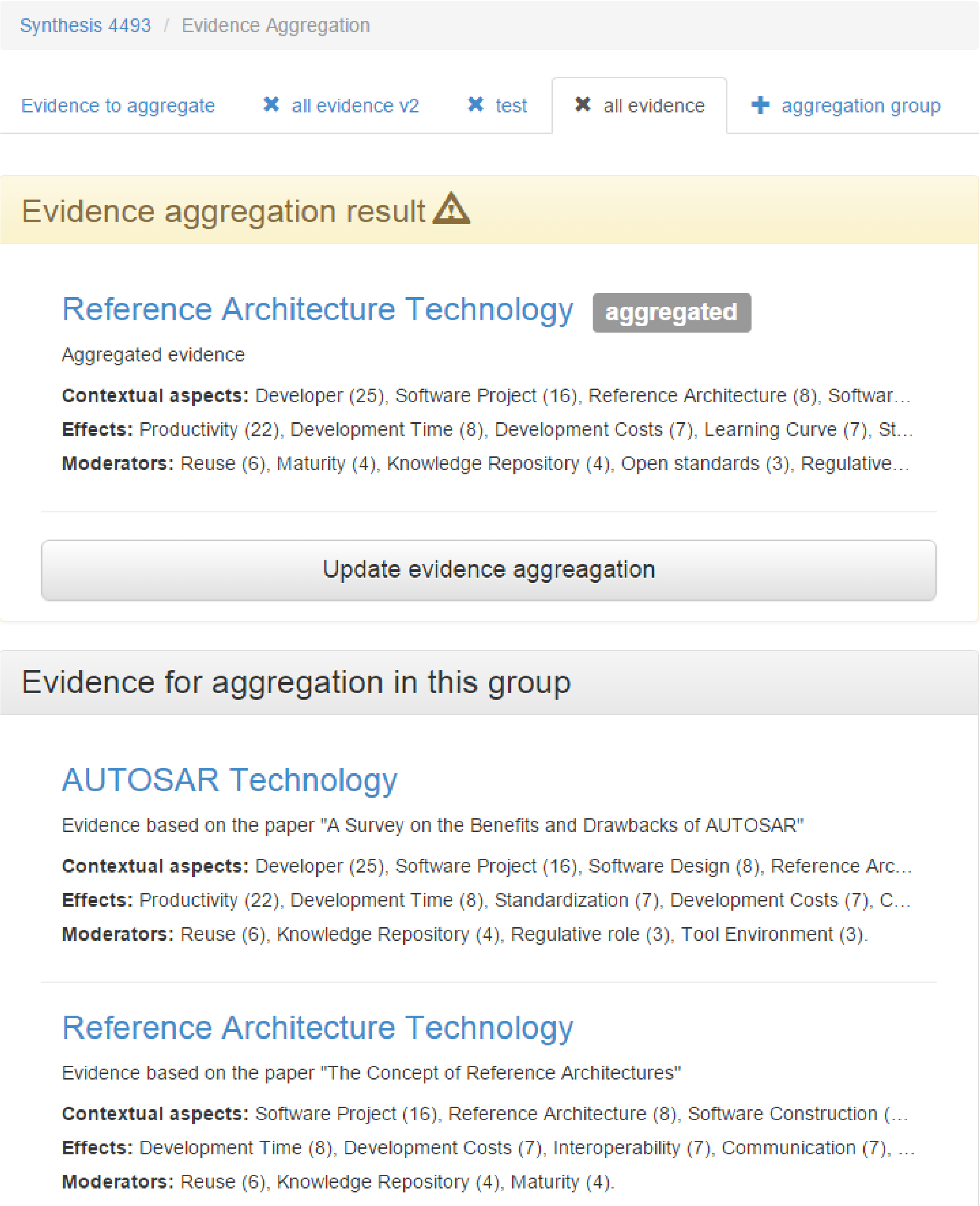
Evidence based on the paper "Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study"

Contextual aspects: Developer (25), Software Project (16), Reference Architecture (8), Software Construction (8), Software Design (8), E...

Effects: Learning Curve (7), Interoperability (7), Communication (7), Standardization (7), Risk (6), Best practices (6), Flexibility for Supplier...

Moderators: Maturity (4), Organizational thinking (3), Existing software (3), Vendor lock-in (3), Open standards (3).

Figure 39 – Grouping evidence for aggregation



Synthesis 4493 / Evidence Aggregation

Evidence to aggregate ✖ all evidence v2 ✖ test ✖ all evidence ✚ aggregation group

Evidence aggregation result ⚠

Reference Architecture Technology aggregated

Aggregated evidence

Contextual aspects: Developer (25), Software Project (16), Reference Architecture (8), Softwar...

Effects: Productivity (22), Development Time (8), Development Costs (7), Learning Curve (7), St...

Moderators: Reuse (6), Maturity (4), Knowledge Repository (4), Open standards (3), Regulative...

Update evidence aggreagation

Evidence for aggregation in this group

AUTOSAR Technology

Evidence based on the paper "A Survey on the Benefits and Drawbacks of AUTOSAR"

Contextual aspects: Developer (25), Software Project (16), Software Design (8), Reference Arc...

Effects: Productivity (22), Development Time (8), Standardization (7), Development Costs (7), C...

Moderators: Reuse (6), Knowledge Repository (4), Regulative role (3), Tool Environment (3).

Reference Architecture Technology

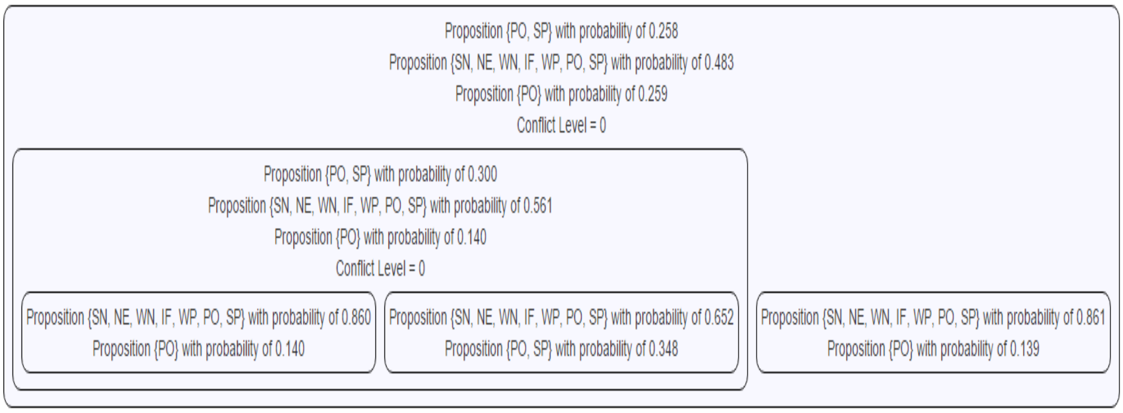
Evidence based on the paper "The Concept of Reference Architectures"

Contextual aspects: Software Project (16), Reference Architecture (8), Software Construction (...

Effects: Development Time (8), Development Costs (7), Interoperability (7), Communication (7), ...

Moderators: Reuse (6), Knowledge Repository (4), Maturity (4).

Figure 40 - A group defined for evidence aggregation



Proposition {PO, SP} with probability of 0.258
Proposition {SN, NE, WN, IF, WP, PO, SP} with probability of 0.483
Proposition {PO} with probability of 0.259
Conflict Level = 0

Proposition {PO, SP} with probability of 0.300
Proposition {SN, NE, WN, IF, WP, PO, SP} with probability of 0.561
Proposition {PO} with probability of 0.140
Conflict Level = 0

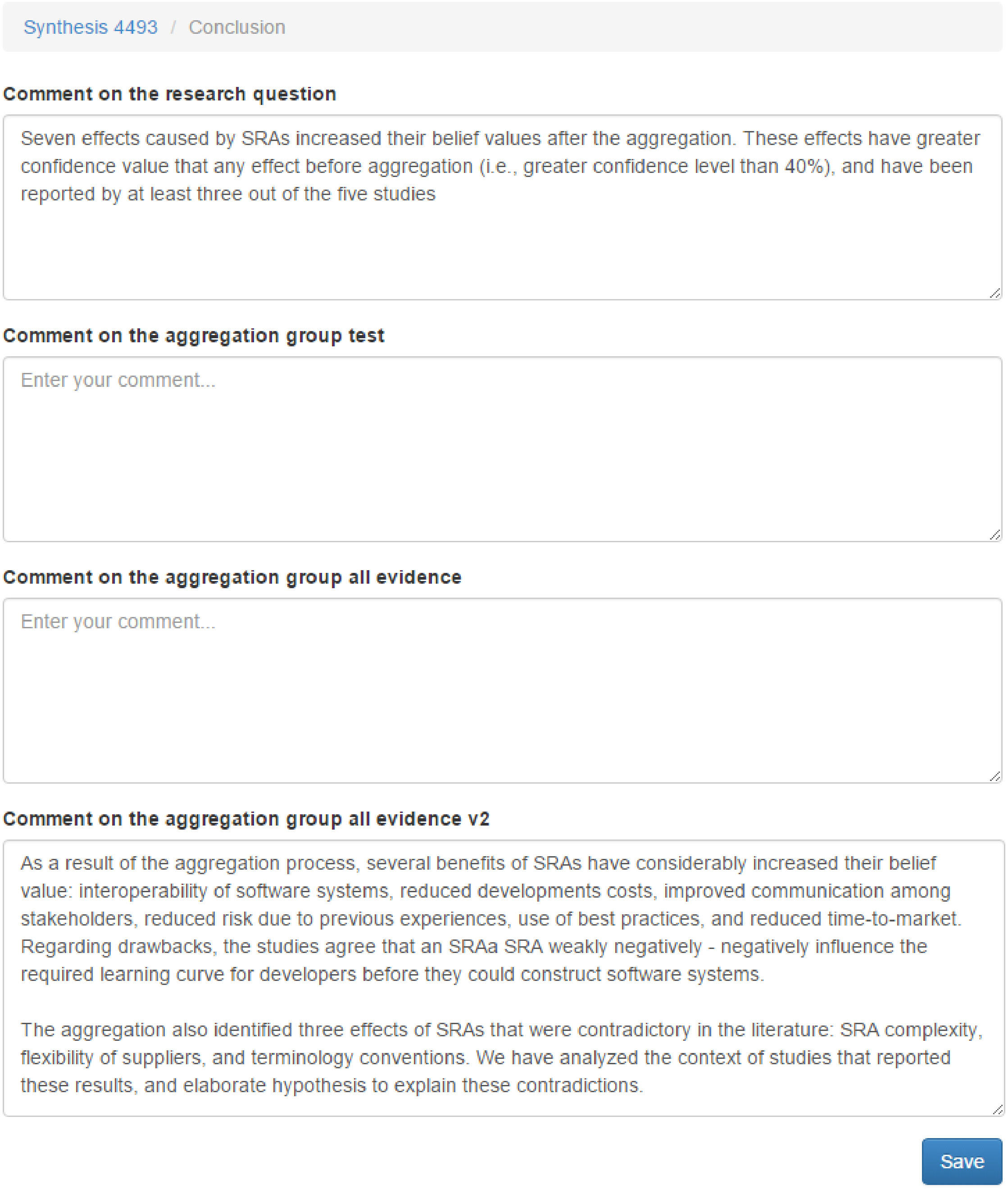
Proposition {SN, NE, WN, IF, WP, PO, SP} with probability of 0.860
Proposition {PO} with probability of 0.140

Proposition {SN, NE, WN, IF, WP, PO, SP} with probability of 0.652
Proposition {PO, SP} with probability of 0.348

Proposition {SN, NE, WN, IF, WP, PO, SP} with probability of 0.861
Proposition {PO} with probability of 0.139

Figure 41 – D-S theory computation details of evidence aggregation. In this example, three evidence are being combined (two at a time). The result of each combination is shown above the combined pair

The last step of SSM is reserved for the analysis of the found results. As this is a non-structured activity, Evidence Factory provides very basic support as shown in Figure 42. There is only a text field for each group defined for aggregation and one additional field for comments related to the research question. The idea is to only register the most important information related to the analysis of the results.



Comment on the research question

Seven effects caused by SRAs increased their belief values after the aggregation. These effects have greater confidence value that any effect before aggregation (i.e., greater confidence level than 40%), and have been reported by at least three out of the five studies

Comment on the aggregation group test

Enter your comment...

Comment on the aggregation group all evidence

Enter your comment...

Comment on the aggregation group all evidence v2

As a result of the aggregation process, several benefits of SRAs have considerably increased their belief value: interoperability of software systems, reduced developments costs, improved communication among stakeholders, reduced risk due to previous experiences, use of best practices, and reduced time-to-market. Regarding drawbacks, the studies agree that an SRAa SRA weakly negatively - negatively influence the required learning curve for developers before they could construct software systems.

The aggregation also identified three effects of SRAs that were contradictory in the literature: SRA complexity, flexibility of suppliers, and terminology conventions. We have analyzed the context of studies that reported these results, and elaborate hypothesis to explain these contradictions.

Figure 42 - Research synthesis analysis

7.5 Future works

There are many opportunities to evolve the Evidence Factory infrastructure. They are essentially associated with the limitations of the current stage of its implementation. Several specified functionalities are not implemented yet. The priority focus should be on the collaborative synthesis and on developing more facilities related to the step 4 of SSM on evidence diagram editor page. For instance, it would be interesting if researchers could keep the trace from the paper text and the concepts created. There are many other requirements specified that were not implemented (Appendix D) considering the time frame of this research.

After these short-term improvements, research efforts should be focused on experimentally evaluating the infrastructure. There is an intention to conduct a study about the system usefulness and perceived ease-of-use using the TAM (Technology Acceptance Model) definition of these two variables.

8 Worked examples: using SSM to synthesize research in Software Engineering

This chapter presents two research synthesis studies. The goal is both to describe in detail how SSM is applied, but also to use them as a kind of evaluation in terms of its applicability. Furthermore, we expect that the examples can be used as a supplementary material for SSM.

8.1 Introduction

As pointed in several parts of this thesis, the objective evaluation of scientific methodologies, particularly research synthesis methods, is a complex endeavor. Aware of this challenge, we tried our best to perform an experimental investigation concerned with the SSM feasibility and applicability to SE. But also historically aligned with previous paths followed by other scientific methodologies, we present in this chapter two real applications of SSM as an attempt to allow other researchers to understand the method innards and to provide a supplementary reference material for applying SSM to synthesize research in SE.

The first synthesis is in the classical domain of Software Inspection. This domain was deliberately chosen because it is a well-known domain in SE, particularly within the Empirical Software Engineering community where it has been extensively investigated and was one of the first topics to be target of experimental studies. Thus, we used this in an attempt to draw attention to the application of the SSM method itself rather than to the synthesis results. Chronologically, this first synthesis was performed before the experimental study presented in Chapter 6. At that time, SSM was not as complete as presented in Chapter 5 since, although already proposed as a five stage method, its steps were still not detailed. Therefore, it was our first attempt to show the mechanisms of aggregation procedure (as shown in Section 5.3) with particular attention to the use of D-S theory for evidence aggregation. Still, for this chapter the study presentation was updated to reflect all definitions of SSM.

The second synthesis is in a relatively less experimentally investigated domain of Software Reference Architecture. In this case, the focus was in the synthesis result itself and not in showing SSM features. This study started as an unplanned collaboration between the Experimental Software Engineering group (ESE) at COPPE

– UFRJ and the Software and Service Engineering group (ESSI) at UPC – Barcelona Tech. At that time, the researchers at ESSI group had conducted some studies about the application of reference architectures (Martínez-Fernández *et al.*, 2013, Martínez-Fernández *et al.*, 2015a) and a systematic review related to the same theme (still unpublished, but its protocol can be found in http://www.essi.upc.edu/~smartinez/ProESEM15.pdf). Thus, at their point of research they were interested in consolidating the results and have a chance to analyze them together. After a visit of one of ESSI researchers to the ESE group, the opportunity for collaboration came up and the research efforts towards conducting a research synthesis study about reference architectures using SSM begun.

This chapter presents both synthesis in detail. Most part of the presented material is a direct transcription of the related published papers (Santos and Travassos, 2013, Martínez-Fernández *et al.*, 2015b) with adaptations in its structure and extensions in its text. Hence, this chapter is split in two parts – sections 8.2 and 8.3– each dedicated to one of the research synthesis studies. We discuss our experiences in applying SSM and the Evidence Factory tool support in Section 8.4 and add final remarks in Section 8.5.

8.2 Research synthesis 1: Usage-Based Reading

Usage-Based reading (UBR) is an inspection technique whose primary goal is to drive reviewers to focus on crucial parts of a software artefact from the user's point-of-view. In UBR, faults are not assumed to be of equal importance, and the technique aims at finding the faults that have the most negative impact on the users' perception of system quality. For this, reviewers are given use cases in a prioritized order and inspect the software artefacts following the usage scenarios defined in the ordered use cases. Therefore, a central aspect on focusing inspection effort in UBR is the prioritization of use cases. UBR assumes that the set of use cases can be prioritized in a way reflecting the desired focusing criterion. If the inspection aims at finding the faults that are most critical to a certain system quality attribute, the use cases should be prioritized accordingly.

The research synthesis about UBR follows the five-stage process of SSM, which are described in the next sections. All details of this research synthesis, particularly all theoretical structures, can be found in the Evidence Factory tool at http://evidencefactory.lens-ese.cos.ufrj.br/synthesis/editor/291.

8.2.1 Planning and definition

Using the structure suggested in SSM, the research question was defined as follows:

> *What are the expected effects from Usage-Based Reading inspection technique when it is applied for inspecting high-level design artifacts produced in analysis phase of software development?*

The research question captures aspects related to technology, activity and system leaving out any consideration about the actors' characteristics. Thus, no characteristics about organization, team or persons, such as software process and software development experience, were determinant for the research question scope.

We defined 'Usage-Based Reading' as the only term of the search string. This was possible because UBR is a very specific software technology. Therefore, making the search string more detailed would only add the risk of leaving out papers which did not include terms about the defined activity and system characteristics. As a result, we decided to consider the aspects about activity and system characteristics in the paper inclusion criteria. For exclusion criteria, on the other hand, we eliminated theoretical or analytical papers and papers that are not written in English. The last definition for paper selection is the digital libraries to be used, which in this case was Scopus (http://www.scopus.com).

8.2.2 Selection

We were able to find 15 technical papers in Scopus with the given search string, from which 4 were selected following the inclusion and exclusion criteria. The selection was performed in November'12. Among the excluded papers, one was a duplicate, one classified as theoretical (analyzing the contributions of three included papers), and the others did not fulfill the inclusion criteria. The full list of papers can be obtained in the tool.

The four studies form a family of experiments aiming at investigating UBR performance in identifying faults on software artefacts. Two researchers participated in three of them. The first experiment (Thelin *et al.*, 2001 – Study S1) compared UBR with ad-hoc inspection. The second experiment compared UBR with a checklist-based technique. And the other three studies (Thelin *et al.*, 2003 – Study S2), (Thelin *et al.*, 2004 – Study S3) and (Winkler *et al.*, 2004 – Study S4) compared UBR against a checklist based reading (CBR).

8.2.3 Quality assessment

Following SSM definitions, quality assessment was performed through the use of quality checklists. Based on the study type, as all studies are *quasi*-experiments, the belief values for them have an inferior limit of 0.50. Then, we add to that base value the result from the scoring schema for systematic studies (Appendix A.1). Table 17 presents the computed belief values for the four studies.

Table 17 – Belief values for moderation and causal relationships of theoretical structures

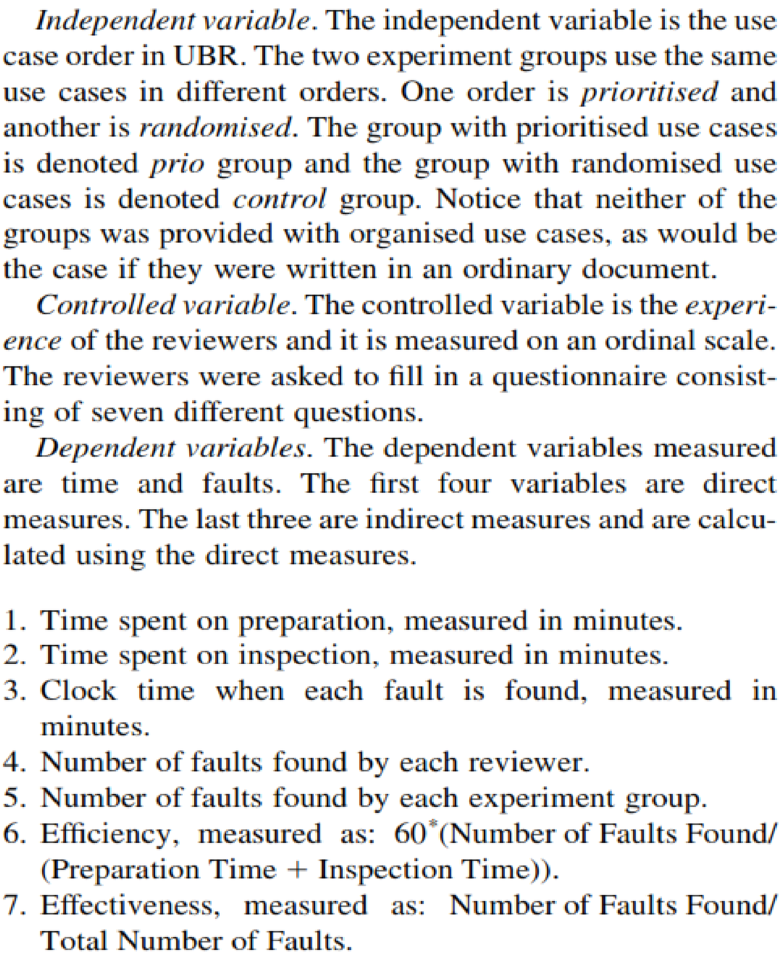
Study	Base belief value	Increase factor based on the study quality	Final belief value
S1	0.50	0.1858 (of 0.25)	0.6858
S2	0.50	0.2042 (of 0.25)	0.7042
S3	0.50	0.2042 (of 0.25)	0.7042
S4	0.50	0.1858 (of 0.25)	0.6858

It is possible to see that belief values are similar. This seems to be a direct result from the fact that the first three papers have common authors. Thus, they tend to share the same textual structure when describing the procedures, analysis, and results. In the case of the fourth study, it is an external replication, which explains why the authors focused in reporting the same aspects in order to facilitate further comparison between the studies.

8.2.4 Extraction and translation

All the experiments used the same set of instruments. Subjects inspected a real-world high-level design document, which consisted of an overview of the software modules and communication signals that are sent to/received from the modules. The application domain regards a taxi management system and the design document specifies the three modules that composes the system: one taxi module used in vehicles, one central module for the operators, and one acting as a communication link between them. All faults were classified into three classes depending on the fault importance from the user's point-of-view. Class A or crucial faults represent faults in system functions that are crucial for a user (i.e., functions that are important for users and that are often used). Class B or important faults represent those which affect important functions for users (i.e., functions that are either important and rarely used or not as important but often used). Class C or minor faults are those which do not prevent the system from continuing to operate.

Information extraction was largely facilitated given the quantitative nature of the studies. Each paper clearly enumerated dependent and independent variables (Figure 43), so that it was straightforward to identify theoretical structures concepts. Context of experiments were enough detailed, which in controlled studies tend to be simpler than observational studies (Figure 44). Moreover, translation procedures were mostly unnecessary since studies' design were similar and used the same set of variables as surrogates. Causal relationships were extracted from the statistical tests used for answering the research questions. It is important to say that extraction and translation are solely based on what is reported. Thus, even when researchers know important variables regarding the object of study at hand, modeled theoretical structures should have only what is in papers' text. For instance, we are aware that several, if not most, studies on software inspection consider the inspector's experience as a variable. Still, we could not include this variable into the theoretical structures as the four studies did not observed this aspect.



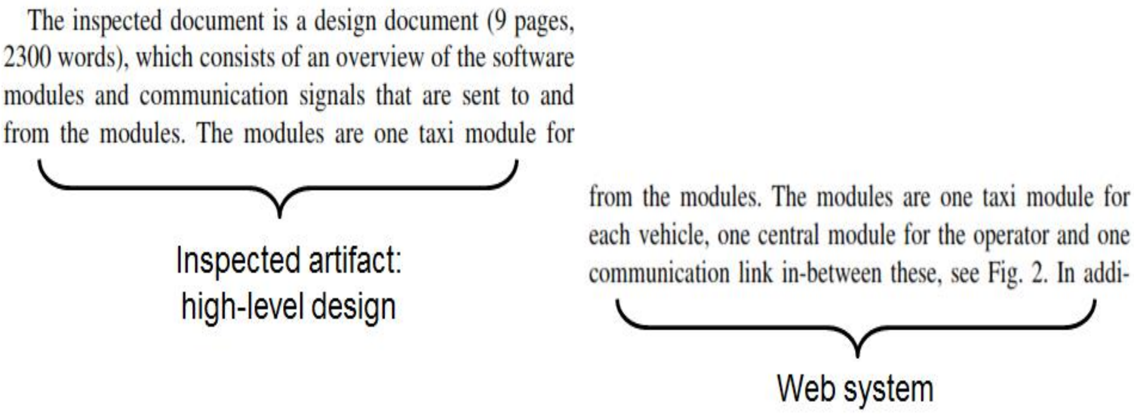
Independent variable. The independent variable is the use case order in UBR. The two experiment groups use the same use cases in different orders. One order is *prioritised* and another is *randomised*. The group with prioritised use cases is denoted *prio* group and the group with randomised use cases is denoted *control* group. Notice that neither of the groups was provided with organised use cases, as would be the case if they were written in an ordinary document.

Controlled variable. The controlled variable is the *experience* of the reviewers and it is measured on an ordinal scale. The reviewers were asked to fill in a questionnaire consisting of seven different questions.

Dependent variables. The dependent variables measured are time and faults. The first four variables are direct measures. The last three are indirect measures and are calculated using the direct measures.

1. Time spent on preparation, measured in minutes.
2. Time spent on inspection, measured in minutes.
3. Clock time when each fault is found, measured in minutes.
4. Number of faults found by each reviewer.
5. Number of faults found by each experiment group.
6. Efficiency, measured as: 60*(Number of Faults Found/(Preparation Time + Inspection Time)).
7. Effectiveness, measured as: Number of Faults Found/Total Number of Faults.

Figure 43 – Study S1 variables listing



The inspected document is a design document (9 pages, 2300 words), which consists of an overview of the software modules and communication signals that are sent to and from the modules. The modules are one taxi module for

from the modules. The modules are one taxi module for each vehicle, one central module for the operator and one communication link in-between these, see Fig. 2. In addi-

Figure 44 – Examples of concept identification for theoretical structure modeling

Given the similarity between studies, the theoretical structures for the four studies share the same concepts and relationships. Figure 45 depicts the theoretical structure modeled for the studies based on the information extracted. The only difference between theoretical structures from the four studies is related to the dependent variables. Two papers do not consider minor defects (class C) in their analysis. There are not any explicit justification for that, but we conjecture that it can be associated to publication space restrictions. Table 18 enumerates all effects along with its intensity and belief value (already adjusted with the discount from the *p-value*).

Table 18 – Effects reported in UBR primary studies

Study **Effect**	**Effects shown as: intensity (belief value)**			
	S1	**S2**	**S3**	**S4**
Efficiency (total faults)	{SP} (0.66)	{SP} (0.67)	{WP, PO} (0.68)	{PO} (0.65)
Efficiency (crucial faults)	{PO, SP} (0.69)	{PO, SP} (0.70)	{WP, PO} (0.70)	{WP, PO} (0.68)
Efficiency (important faults)	{PO} (0.68)	{WP} (0.60)	{WP} (0.70)	{IF, WP} (0.69)
Efficiency (minor faults)		{WP} (0.52)	{WP} (0.70)	
Effectiveness (total faults)	{WP, PO} (0.64)	{PO} (0.63)	{PO} (0.70)	{SP} (0.67)
Effectiveness (crucial faults)	{PO, SP} (0.68)	{PO, SP} (0.68)	{PO, SP} (0.70)	{SP} (0.69)
Effectiveness (important faults)	{PO} (0.68)	{WP, PO} (0.58)	{PO} (0.70)	{IF, WP} (0.69)
Effectiveness (minor faults)		{IF, WP} (0.60)	{WP} (0.70)	
# Total faults	{SP} (0.69)	{PO} (0.63)	{PO} (0.70)	{SP} (0.67)
# Crucial faults	{PO, SP} (0.69)	{PO, SP} (0.68)	{PO, SP} (0.70)	{SP} (0.69)
# Important faults	{WP, PO} (0.69)	{WP, PO} (0.58)	{PO} (0.70)	{IF, WP} (0.69)
# Minor faults	{WP} (0.69)	{IF, WP} (0.60)	{WP} (0.70)	

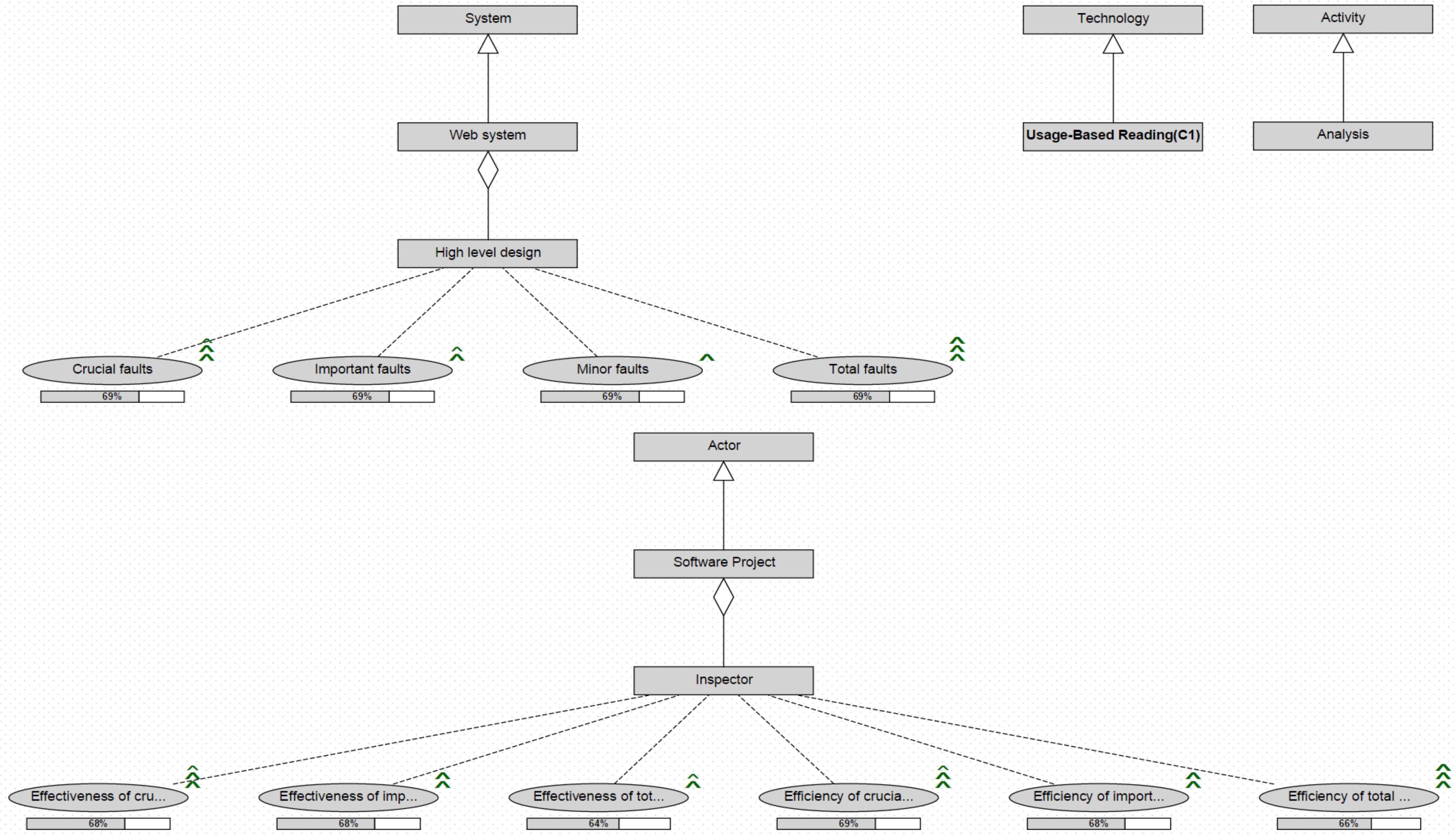


Figure 45 – Evidence model representing study S1 results (Thelin et al., 2001)

It is important to notice at this point that, although we are focusing on the theoretical structures for UBR, they were modeled using the dismembering operation. This means that first we modeled comparative theoretical structures (comparing UBR with ad-hoc or CBR) and, then, based on the differences of the comparative cause-effect relationships, we determined the intensity of effects for UBR. We choose this strategy, instead of extracting two descriptive theoretical structures from comparative studies as recommended in SSM, because papers contained percentage difference in most cases. Still, when individual data about each technology was present, we used it to calibrate the dismembering operation – that is, making it more precise than defined in Table 7. Even indirect data, such as graphical data and boxplots, were used to that end. In Table 19, we list the effects for study S1 detailing how they were dismembered (Appendix C details dismembering for all studies).

Table 19 – Dismembering operation values for study S1

Effect	Comparative	Descriptive for ad-hoc	Descriptive for UBR
Efficiency (total faults)	{WS}	{PO}	{SP}
Efficiency (crucial faults)	{SU}	{WP}	{PO, SP}
Efficiency (important faults)	{WS}	{WP}	{PO}
Effectiveness (total faults)	{WS}	{WP}	{WP, PO}
Effectiveness (crucial faults)	{SU}	{WP}	{PO, SP}
Effectiveness (important faults)	{WS}	{WP}	{PO}
# Total faults	{WS}	{PO}	{SP}
# Crucial faults	{SU}	{WP}	{PO, SP}
# Important faults	{WS}	{WP}	{WP, PO}
# Minor faults	{WS}	{WP, PO}	{WP}

As both comparative and descriptive values (when available) were extracted to define the effects intensity, it was required to define the respective conversion rules as shown in Table 20. It is important to say, however, that as both comparative and descriptive rules were defined, in some cases they conflicted. For instance, in the case of ‘efficiency (crucial faults)’ the percentage difference between the inspection techniques is 95% and the mean values of identified faults per hour (obtained from an approximation of boxplot values – Hozo *et al.*, 2005) is 1.293 and 2.533 for ad-hoc and UBR, respectively. Therefore, if only the percentage difference is considered, then the descriptive values obtained from dismembering operation should have two units of distance (*e.g.*, WP and SP), since the 95% percentage difference is converted to {SU}. On the other hand, the approximate values of 1.293 and 2.533 are converted to {WP} and {PO} according to the defined rules. In these conflicting cases, in order to make the comparative and descriptive conversion rules compatible, we reduced the converted effects intensity precision. As a result, in this same example, the comparative value {SU} was dismembered to {WP} and {PO, SP} instead of {WP} and {PO}.

Table 20 – Conversion rules for effects quantitative values

Effect		Comparative qualitative intensity/difference	Quantative rule range
Comparative	Efficiency Effectiveness # defects	Indifferent (IF)	[0%, 0%]
		Weak difference (WI or WS)	(0%, 50%]
		Moderate difference (IN or SU)	(50%, 100%]
		Strong difference (FS or FI)	–[28]
Descriptive	Efficiency	Indifferent (IF)	0
		Weak impact (WN or WP)	(0, 2.5]
		Moderate impact (NE or PO)	(2.5, 5]
		Strong impact (SN or SP)	(5, ∞]
	Effectiveness	Indifferent (IF)	0
		Weak impact (WN or WP)	(0, 0.33]
		Moderate impact (NE or PO)	(0.33, 0.66]
		Strong impact (SN or SP)	(0.66, 1]
	# defects	Indifferent (IF)	0
		Weak impact (WN or WP)	(0, 4]
		Moderate impact (NE or PO)	(4, 8]
		Strong impact (SN or SP)	(8, 12]

8.2.5 Aggregation and analysis

In order to answer the research question defined for this study, we used only the dismembered theoretical structures relative to UBR. Given their similarity, we were not able to identify any incompatibility between them. Thus, all four studies were analyzed together in a single aggregation. Some studies did not analyze (or report) some variables related to minor faults, but this is not impeditive for the aggregation since in

[28] As we observed that the compared technologies are always able to identify defects (positive effects), we decided to not use strong difference.

SSM each effect is individually aggregated considering the papers in which they are present.

After this compatibility analysis and given the confidence level of each effect, Dempster's rule of combination could be computed. The combined theoretical structure is shown in Figure 46 and the detailed aggregation results are listed in Table 21. The first column shows the reported effect (i.e., benefit or drawback). The second column indicates the number of papers that have reported this effect. The third column shows the aggregated intensity about how the SRA causes such effect (e.g., positive or negative). The fourth column represents the aggregated belief on such effect. This is one of the most interesting results of the aggregation. The fifth column lists conflict levels computed in each combination for the respective effect. For instance, the aggregation of four pieces of evidence leads to three combinations. Conflicts are always shown in the same order ((S1 $\oplus$ S3) $\oplus$ S4) $\oplus$ S2. This order was applied by the Evidence Factory tool, based on the evidence order of the randomly given IDs. The sixth column registers the difference between maximum belief value of individual evidence for the respective effect and the aggregated value. The effects that were most strengthened where effectiveness and number of crucial faults.

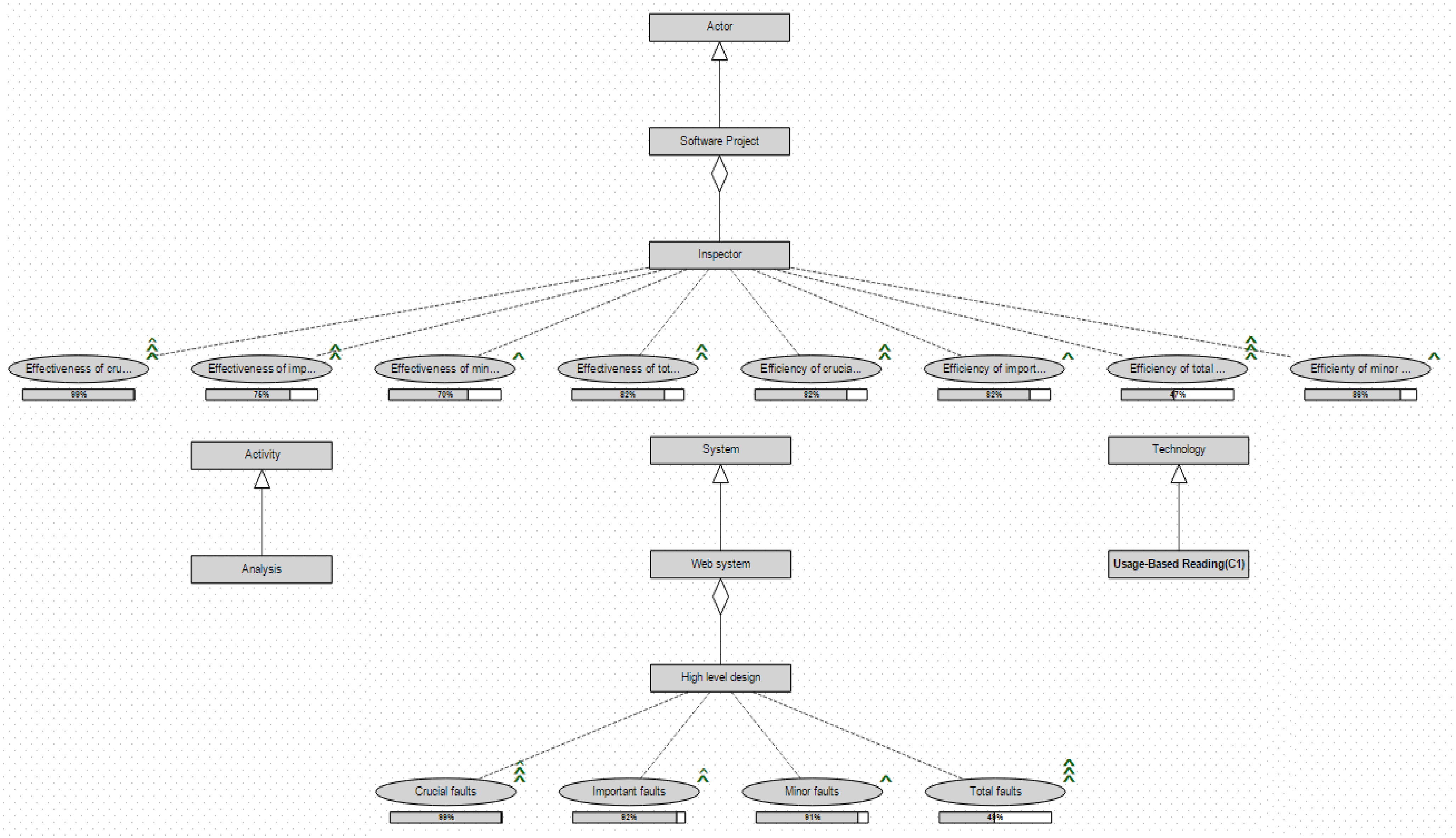


Figure 46 – Aggregated theoretical structure for UBR synthesis

Table 21 – Aggregated effects of UBR

Effect	**Aggregation Results**				
	#Papers	***Intensity***	***Belief***	***Conflicts***[29]	***Difference***[30]
Efficiency (total faults)	4	{SP}	0.47	0.45, 0.25, 0.49	-0.21
Efficiency (crucial faults)	4	{PO}	0.82	0.00, 0.00, 0.00	0.12
Efficiency (important faults)	4	{WP}	0.82	0.48, 0.27, 0.10	0.12
Efficiency (minor faults)	2	{WP}	0.86	0.00	0.16
Effectiveness (total faults)	4	{PO}	0.82	0.00, 0.60, 0.12	0.12
Effectiveness (crucial faults)	4	{PO, SP}	0.99	0.00, 0.00, 0.00	0.29
Effectiveness (important faults)	4	{PO}	0.75	0.00, 0.64, 0.00	0.05
Effectiveness (minor faults)	2	{WP}	0.70	0.00	0.00
# Total faults	4	{SP}	0,49	0.48, 0.28, 0.46	-0.21
# Crucial faults	4	{PO, SP}	0.99	0.00, 0.00, 0.00	0.29
# Important faults	4	{WP, PO}	0,93	0.00, 0.48, 0.00	0.23
# Minor faults	3	{WP}	0,91	0.00, 0.00	0.21

Before analyzing the combination results, we should first define how conflicts occurred in the aggregation should be resolved. Although we have not had any incompatibility between theoretical structures, we can notice major conflicts between study results (*i.e.*, effects intensity). There are three main factors associated with these conflicts. The first comes from the fact that we dismembered results from comparisons between UBR and ad hoc, and UBR and CBR. Therefore, it is expected some differences among results. The second aspect is related to the dismembering operation itself. As defined in SSM, dismembering is imprecise and suggested to be used only in some specific situations. Thus, it is a potential source of differences between results.

[29] Lists all conflict levels computed in each combination for the respective effect. For instance, the aggregation of 4 evidence leads to 3 combinations. Conflicts are always presented in the same order $((S1 \oplus S3) \oplus S4) \oplus S2$. This order was applied by the Evidence Factory tool, based on the evidence order of the randomly given IDs.

[30] The difference column measures the difference among the max value of belief in previous single papers, and the gained confidence after the aggregation.

The last aspect considered for explaining results is that the second combination (between S4 and resulting aggregation from S1 and S3) has the highest frequency of conflict occurrence. Interestingly enough, this is the combination involving the study S4, which is the only study that is an external experiment of UBR.

Combined belief values presented in Table 21 were computed using the basic conflict resolution strategy of SSM, which ignores the conflict by redistributing it among hypotheses. However, to use this strategy SSM recommends that conflicts larger than 0.50 or mean conflict above 0.33 (as in this particular case of 3 combinations we have 1/3 = 0.3333) should be ignored. So, based on the explanation for the conflicts, we understood that the best strategy to handle conflicts in this aggregation was incorporation. In other words, by using the dismembering function and aggregating results from comparison of different techniques, we are much more interested in the trend than the specific result within the *Likert* scale. This is directly related to the incorporation conflict strategy, which tend to produce relatively more imprecise results. Next, in Table 22 the new belief values, after conflicts resolution, are presented.

Table 22 – Aggregated effects of UBR after conflicts resolution by incorporation

Effect	**Aggregation Results**				
	#Papers	***Intensity***	***Belief***	***Conflicts***	***Difference***
Efficiency (total faults)	4	{PO, SP}	0.85	*(INCORPORATED)*	0.17
Efficiency (crucial faults)	4	{PO}	0.82	0.00, 0.00, 0.00	0.12
Efficiency (important faults)	4	{WP}	0.82	0.48, 0.27, 0.10	0.12
Efficiency (minor faults)	2	{WP}	0.86	0.00	0.16
Effectiveness (total faults)	4	{PO, SP}	0.87	*(INCORPORATED)*	0.17
Effectiveness (crucial faults)	4	{PO, SP}	0.99	0.00, 0.00, 0.00	0.29
Effectiveness (important faults)	4	{WP, PO}	0.77	*(INCORPORATED)*	0.07
Effectiveness (minor faults)	2	{WP}	0.70	0.00	0.00
# Total faults	4	{PO, SP}	0.99	*(INCORPORATED)*	0.29
# Crucial faults	4	{PO, SP}	0.99	0.00, 0.00, 0.00	0.29
# Important faults	4	{WP, PO}	0,93	0.00, 0.48, 0.00	0.23
# Minor faults	3	{WP}	0,91	0.00, 0.00	0.21

We also present details of one conflicting aggregation to illustrate the conflict incorporation procedure (Table 23). As previously defined in SSM, instead of redistributing the conflict among all hypotheses, the idea of incorporation is to stretch the range of effect intensity by putting the conflict value into a contiguous range that includes the conflicting pair of hypotheses sets. For instance, in the first combination of Table 23 (between studies S1 and S3), there is a conflict value of 0.455 between the hypotheses {SP} from study S1 and {WP, PO} from study S3 which is assigned to the hypothesis {WP, PO, SP}. Thus, in this case, we have the positive trend for the effect that includes all positive values of the *Likert* scale ({WP, PO, SP}) and not a precise intensity ({WP}, {PO} or {SP}) for it. The same operation is performed in the other conflicts. After the three combinations, the results of aggregation are presented in the bottom of Table 23. The hypothesis {PO, SP} was chosen based on the selection procedure defined in Figure 16. $Bel_{1,2,3,4}$({WP,PO,SP}) has the largest belief value of 0.987, but $Bel_{1,2,3,4}$({PO, SP}) has more than 75% of its value, since 0.854/0.987 = 0.865.

Table 23 – Details of calculations for combining results of 'efficiency (total faults)' effect incorporating conflicts

Combination of studies S1 and S3		
m_3 ***m_1***	**{WP,PO} (0.678)**	**Θ (0.322)**
{SP} (0.656)	Ø (0.445)	{SP} (0.211)
Θ (0.344)	{WP,PO} (0.233)	Θ (0.111)
Combination of study S4 with resulting combination of studies S1 and S3		
m_4 ***$m_{1,3}$***	**{PO} (0.649)**	**Θ (0.351)**
{SP} (0.211)	Ø (0.137)	{SP} (0.074)
{WP,PO} (0.233)	{PO} (0.151)	{WP,PO} (0.082)
{WP,PO,SP} (0.445)	{PO} (0.289)	{WP,PO,SP} (0.156)
Θ (0.111)	{PO} (0.072)	Θ (0.039)
Combination of study S2 with resulting combination of studies S1, S3 and S4		
m_2 ***$m_{1,3,4}$***	**{SP} (0.675)**	**Θ (0.325)**
{SP} (0.074)	{SP} (0.050)	{SP} (0.024)
{PO, SP} (0.137)	{SP} (0.092)	{PO,SP} (0.045)
{PO} (0.512)	Ø (0.346)	{PO} (0.166)
{WP,PO} (0.082)	Ø (0.055)	{WP,PO} (0.027)
{WP,PO,SP} (0.156)	{SP} (0.105)	{WP,PO,SP} (0.051)
Θ (0.039)	{SP} (0.026)	Θ (0.013)

Final combined probabilities and belief values	
***$m_{1,2,3,4}$*({SP})** = 0.050 + 0.092 + 0.105 + 0.026 + 0.024 = 0.297	***$Bel_{1,2,3,4}$*({SP})** = 0.297
***$m_{1,2,3,4}$*({PO, SP})** = 0.346 + 0.045 = 0.391	***$Bel_{1,2,3,4}$*({PO, SP})** = 0.166 + 0.297 + 0.391 = 0.854
***$m_{1,2,3,4}$*({PO})** = 0.166	***$Bel_{1,2,3,4}$*({PO})** = 0.166
***$m_{1,2,3,4}$*({WP,PO})** = 0.027	***$Bel_{1,2,3,4}$*({WP,PO})** = 0.166 + 0.027 = 0.193
***$m_{1,2,3,4}$*({WP,PO,SP})** = 0.055 + 0.051 = 0.106	***$Bel_{1,2,3,4}$*({WP,PO,SP})** = 0.166 + 0.297 + 0.391 + 0.027 + 0.106 = 0.987
***$m_{1,2,3,4}$*(Θ)** = 0.013	***$Bel_{1,2,3,4}$*(Θ)** = 1

Now, with conflicts discussed and resolved, we focus on the results themselves. It is noticeable the large agreement between studies regarding results associated with crucial faults. This is manifested in the high belief value of 0.99 observed in efficiency, effectiveness and number of crucial faults. The high belief values resulting from aggregation should not be analyzed from the absolute value itself, as each aggregation has its own specificities. In this case, the 0.99 belief value should not be necessarily interpreted as an ‘almost certainty’ (*i.e.*, belief value of 1), but rather as a virtually full agreement among four strong evidence (*i.e.*, *quasi*-experiments). Thus, in other words, the current body of knowledge indicates that UBR seems to have a direct impact to crucial faults, since it is possible to observe similar results in four different studies which even compare different technologies (ad hoc and CBR).

Another interesting finding that can be observed in the aggregated results is the relative difference between intensity of effects associated with crucial and minor faults. The results suggest that UBR has larger impact on crucial faults than minor faults. This is precisely the most important aspect of UBR as it focuses inspections to the most important type of faults. It was observed in all dimensions explored in studies: efficiency, effectiveness and number of faults. UBR has a {PO} impact over efficiency relative to crucial faults while it has {WP} for efficiency relative to minor faults. For effectiveness we found {PO, SP} for crucial faults and {WP} for minor faults. This was the same for number of crucial faults. Thus, this consistency in the difference between crucial and minor faults among the studies is another important result strengthened in the aggregation.

Based on this analysis and overall results of Table 22, we have enough input to answer the research question defined for this synthesis. UBR inspection technique can safely be used for identifying most important (*i.e.*, crucial) faults in high-level design, with high level of efficiency and effectiveness. It still can be used for less important faults, although with relatively less efficacy. These effects seems to result from the basic mechanism behind UBR, which is the assumption that the proper prioritization of use cases can help identifying relatively more important faults.

The scope in which the aggregation findings can be claimed to be valid are explicit in the aggregated theoretical structure (Figure 46). In all studies, the same Web system’s high-level design models were inspected using UBR. Thus, it is difficult to argue about any kind of generalization beyond this context. Still, the cause for the observed effects is theoretically reproducible in other contexts with different kinds of systems and software artefacts, since UBR working mechanism is based on use case prioritization which is, at least theoretically, independent of the inspected software artefacts. Moreover, the studies did not explicitly considered the participation of graduation students as an important factor influencing the findings. Arguably, this is due to the fact that most subjects have experience in SE industry. Following this line of reasoning, we understand that industry professionals can be included within the findings external validity. That is why we used the concept ‘Inspector’ to generically refer to the actor.

Besides external validity, we should extend our considerations to other types of validity threads. We believe that the most important internal validity threat is the potential bias associated with the fact that the synthesis was conducted by the same researcher that authored SSM. Thus, from the studies selection to the definition of concepts and their relationships, practically all steps were subjected to this issue. This was the main motivation for choosing an inspection technique as theme for research synthesis, so that the domain aspects would not represent a confound factor during the synthesis process. Regarding construct validity, we should point the use of the dismembering operation, which represent a validity threat in itself as it increase the imprecision of effects intensity. To minimize this lack of accuracy, when apart from the percentage difference the absolute quantitative values were available they were used to improve the effects precision.

8.3 Research synthesis 2: Software Reference Architecture

Nakagawa *et al.* (2011) define Software Reference Architecture (SRA) as '*an architecture that encompasses the knowledge about how to design concrete architectures of systems of a given application domain; therefore, it must address the business rules, architectural styles (sometimes also defined as architectural patterns that address quality attributes in the SRA), best practices of software development (for instance, architectural decisions, domain constraints, legislation, and standards), and the software elements that support development of systems for that domain*'. All of this must be supported by a unified, unambiguous, and widely understood domain terminology'. SRAs mainly appear in organizations where the multiplicity of similar software systems (*i.e.*, systems developed at multiple locations, by multiple vendors, and across multiple organizations) triggers a need for life-cycle support for all system (Cloutier *et al.*, 2010). Therefore, SRAs are attractive when organizations become large and distributed in order to develop new software systems or new versions of existing ones. In this context, organizations need to analyze whether or not to acquire a SRA.

Angelov *et al.* (2012) distinguish between different types of SRAs. These different types are classified by the following characteristics:

- Goal: to *standardize* concrete architectures of software systems (aiming at system/component interoperability) or to *facilitate* the design of concrete architectures (aiming at providing guidelines/inspiration for the design of systems).
- Organizations in which the SRA is used: a SRA can be used in a *single organization*, or in *multiple organizations* that share a certain property (e.g., car manufactures).

- Definition type: a *preliminary* SRA is defined when the technology, software solutions, or algorithms demanded for its application do not yet exist in practice whereas a *classical* SRA is defined when these artifacts exist by the time of its design and have been tested in practice.

Examples of well-known SRAs are: AUTOSAR, which is a classical SRA that aims to standardize the software architecture of electronic control units in modern vehicles, and targets multiple organizations (e.g., many car manufactures) (Martínez-Fernández *et al.*, 2015a); classical service-oriented SRAs, such the one studied in Galster *et al.* (2013) that facilitates the design of variability-intensive service-based applications, and targets many municipalities in the Netherlands using e-government systems.

The research synthesis about SRA follows the five-stage process of SSM, which are described in the next sections. All details of this research synthesis, particularly all theoretical structures, can be found in the Evidence Factory tool at http://evidencefactory.lens-ese.cos.ufrj.br/synthesis/editor/4493.

8.3.1 Planning and definition

The research question is defined as:

> *'What are the trends on available empirical evidence about the benefits and drawbacks of SRAs for acquisition organizations?'*

We focused on the benefits and drawbacks for organizations that introduce a SRA for designing and constructing a family of software systems. Therefore, we focused on the SRA 'usage' perspective, rather than other perspectives, *e.g.*, SRA 'design'.

For the search strategy, we considered the same data sources and search string as in a systematic review about SRA engineering that has been conducted in conjunction with the LabES-USP group, whose protocol is available at www.essi.upc.edu/~smartinez/ProESEM15.pdf. Therefore, the following electronic databases were used: Scopus (scopus.com), Web of Science (isiknowledge.com), IEEE Xplore (ieee.org/web/publications/xplore), ACM Digital Library (dl.acm.org), ScienceDirect (sciencedirect.com), and Springer (link.springer.com).

To conduct the search, we used the following search string, which aims to find scientific studies of SRAs for the concrete architecture of software systems:

(“reference architecture?”) AND (“software architecture?” OR “software structure?” OR “software design?” OR “system architecture?” OR “system structure?” OR “system design?”)

The search was conducted using filters on the titles, abstracts, and keywords of the studies. Besides searching and collecting the papers, we used the forward and backward snowballing strategy for the included papers. Experts or reviewers suggestions were also accepted, which is essential for studies that are not indexed or published yet (i.e., in press).

As inclusion criteria, we defined any empirical study reporting findings based on evidence about the benefits and drawbacks of adopting SRAs. Concerning exclusion, we defined three exclusion criteria: (i) studies whose findings were based on opinions rather than evidence; (ii) studies that were not the primary source of the reported study or data (i.e., papers that reanalyze or review a published study or do not consider data); and (iii) studies that were not reported in English.

8.3.2 Selection and quality assessment

The search string, performed in September’ 14, retrieved 492 non-duplicated studies. From these studies, the empirical studies reporting evidence were manually identified. From this list, we looked for those focusing on reporting the benefits and drawbacks of SRAs.

Three papers that report empirically grounded results about SRA benefits and drawbacks were found (Cloutier *et al.*, 2010, Angelov *et al.*, 2013, Martínez-Fernández *et al.*, 2013). Searching through the references and citations of these three papers, Galster *et al.* (2013) was added to the included studies. Also, Martínez-Fernández *et al.* (2015a) was included by convenience, as it was not published yet, but we were aware of its existence because it was conducted by three of the authors of the original publication of Martínez-Fernández *et al.* (2015b).

Finally, we ended up with five included primary studies reporting evidence on the benefits and drawbacks of using SRA in an organization. The most important details of each of the five papers can be found in Table 24. In addition, we determine the belief values for each study based on SSM definitions, *i.e.*, the study type and quality assessment. For detailed information about the quality assessment, the aforecited link to this synthesis in Evidence Factory tool should be used. Calculated belief values around 0.40 indicate the moderate confidence in the evidence. Thus, it is our interest in this synthesis to analyze how confidence in and which benefits and drawbacks reported in the studies are going to be strengthened or weakened.

Table 24 – Primary studies regarding SRA selected for synthesis

Study Id.	Study Type: Instruments	Participants	SRA Application Domain	SRA goal[31]	SRA used in[32]	SRA type[33]	Belief & evidence type	Year
[S1] Cloutier *et al.* (2010)	Focus group: presentations, discussions	Architects from the System Architecture Forum	Defense and commercial equipment	Standard & Facilitation	Single & multiple Organizations	Preliminary & classical	0.25+0.10=0.35 Qualitative	2010
[S2] Martínez-Fernández *et al.* (2013)	Case study: interviews, questionnaires, docs.	28 sw. architects and developers from IT consulting	Banks, insurers, public administration, utilities, and industries	Standard & Facilitation	Single Organizations	Classical	0.25+0.19=0.44 Qualitative & quantitative	2013
[S3] Angelov *et al.* (2013)	Survey: questionnaires	90 sw. architects and developers from worldwide	n/a	Standard & Facilitation	Single & multiple Organizations	Preliminary & classical	0.25+0.15=0.40 Qualitative & quantitative	2013
[S4] Galster *et al.*(2013)	Case study: interviews, docs., meetings	20 sw. architects, managers and experts from local e-goverment	Variability-intensive service-oriented systems	Facilitation	Multiple Organizations	Classical	0.25+0.15=0.40 Qualitative	2013
[S5] Martínez-Fernández *et al.* (2015a)	Survey: questionnaires	51 practitioners from AUTOSAR partners	Automotive systems	Standard	Multiple Organizations	Classical	0.25+0.17=0.42 Qualitative & quantitative	2015

[31] To check the possible values and description for ‘goal’ see the introduction of this section (8.3).

[32] To check the possible values and description for ‘used in’ see the introduction of this section (8.3).

[33] To check the possible values and description for ‘type’ see the introduction of this section (8.3).

8.3.3 Extraction and translation

The first and the second authors of Martínez-Fernández *et al.* (2015b) divided the five papers into two sets and then individually modeled each evidence. After that, we reviewed the models created by each other, including several meetings to discuss if we have a common understanding about the models. The remaining authors performed a final revision of the models, and the resulting aggregated model. It is interesting to notice that the identification of concepts and relationships is an iterative process, and the modeling of evidence can be a trigger to review the others. This is important to make concepts and evidence structures more consistent, and particularly important for evidence synthesis, which is described in the next section.

We present details of one model to illustrate our perspective when extracting and translating findings reported in papers to theoretical structures. In Figure 47, we show the theoretical structure model for study S1. In the study S1, the driving forces for SRAs are elicited from the discussions of the System Architect Forum (http://architectingforum.org). The authors also present real-world SRAs from different domains to help justifying some of the driving forces elicited. Since the results presented in the study S1 are an outcome of an analysis of discussion between professionals with different background, we decided to use general *value* concept for context description including 'Enterprise Software' and 'Acquisition Organization'. In models of other papers where the context is specific, such as the study S5 describing a SRA for the automotive domain, specific value concepts were used to model it (e.g., 'Automotive Software', and 'AUTOSAR partner').

Apart from the context description, the effects were relatively straightforward to identify as they were explicitly listed in the paper's text. For instance, 'Terminology Conventions' concept in S1 was identified from the following excerpt: '*Reference Architecture can also serve as a framework and lexicon of terms and naming conventions, as well as structural relationships within a company, industry or a domain*'. However, moderators were not so unequivocal, since the authors do not report them as moderators, but rather as particular conditions important to augment some effects. This can be seen in another excerpt where the authors put 'Maturity' as a particular aspect to SRA produce its effects, particularly regarding 'Risk': '*Risk reduction is another potential benefit through the use of proven and partly prequalified architectural elements. The general maturity and experience level associated with a Reference Architecture also bears the promise of a higher quality end product*'. Based on this we defined 'Maturity' as a moderator for 'Risk' effect, which can be seen in

Figure 47 where both concepts are linked through the model textual hint M_3. It is also interesting to notice that the benefit associated with 'Risk' is its reduction. Thus, a *benefit* is a *positive* effect, which is the *reduction* of 'Risk'.

Following this line of analysis, all other theoretical structures were modeled for the selected studies. Table 25, provides a summary of each evidence model with the list of all effects caused by SRAs. For the 23 identified effects, only 4, 'Latest technologies', 'Investment', 'Reliability', and 'Reputation', were reported by just one study.

Table 25 – Effects reported in SRA selected studies

Study Effect	Effects summary shown as: intensity (belief value)				
	S1	S2	S3	S4	S5
Interoperability	{PO, SP} (0.35)		{PO} (0.15)	{WP} (0.40)	{PO, SP} (0.22)
Development costs	{PO, SP} (0.35)	{PO} (0.36)	{PO} (0.04)		{PO} (0.16)
Communication	{PO} (0.35)		{PO} (0.09)	{PO} (0.40)	{PO, SP} (0.20)
Risk	{PO, SP} (0.35)			{PO} (0.40)	{PO} (0.10)
Best practices			{PO} (0.31)	{PO} (0.40)	{PO} (0.13)
Learning curve		{SN, NE} (0.36)	{NE} (0.13)	{NE, WN} (0.40)	{NE} (0.22)
Development time	{PO, SP} (0.35)		{PO} (0.14)		{PO} (0.14)
Maintenance cost		{PO} (0.35)			{PO} (0.14)
Productivity		{PO, SP} (0.30)			{PO} (0.11)
Ease of developing		{PO} (0.30)	{PO} (0.07)		{WP, PO} (0.03)
Alignment		{WP, PO} (0.19)			{WP} (0.07)
Restriction		{NE} (0.13)	{NE} (0.06)		{NE, WN} (0.07)
Standardization		{WP, PO} (0.14)	{PO} (0.16)	{WP} (0.40)	{SP} (0.37)
Latest technologies		{WP} (0.30)			
Investment					{NE} (0.25)
Reliability		{WP, PO} (0.14)			
Dependability		{SN, NE} (0.09)			{NE, WN} (0.12)
Reputation					{WP} (0.06)
Software quality			{NE} (0.06)		{WN} (0.04)
Novel design solution			{PO} (0.05)		{WP} (0.04)
Complexity		{WN} (0.06)			{SN, NE} (0.27)
Terminology conventions	{WP, PO} (0.35)				{NE} (0.17)
Flexibility of suppliers	{PO} (0.35)			{WN, IF} (0.40)	

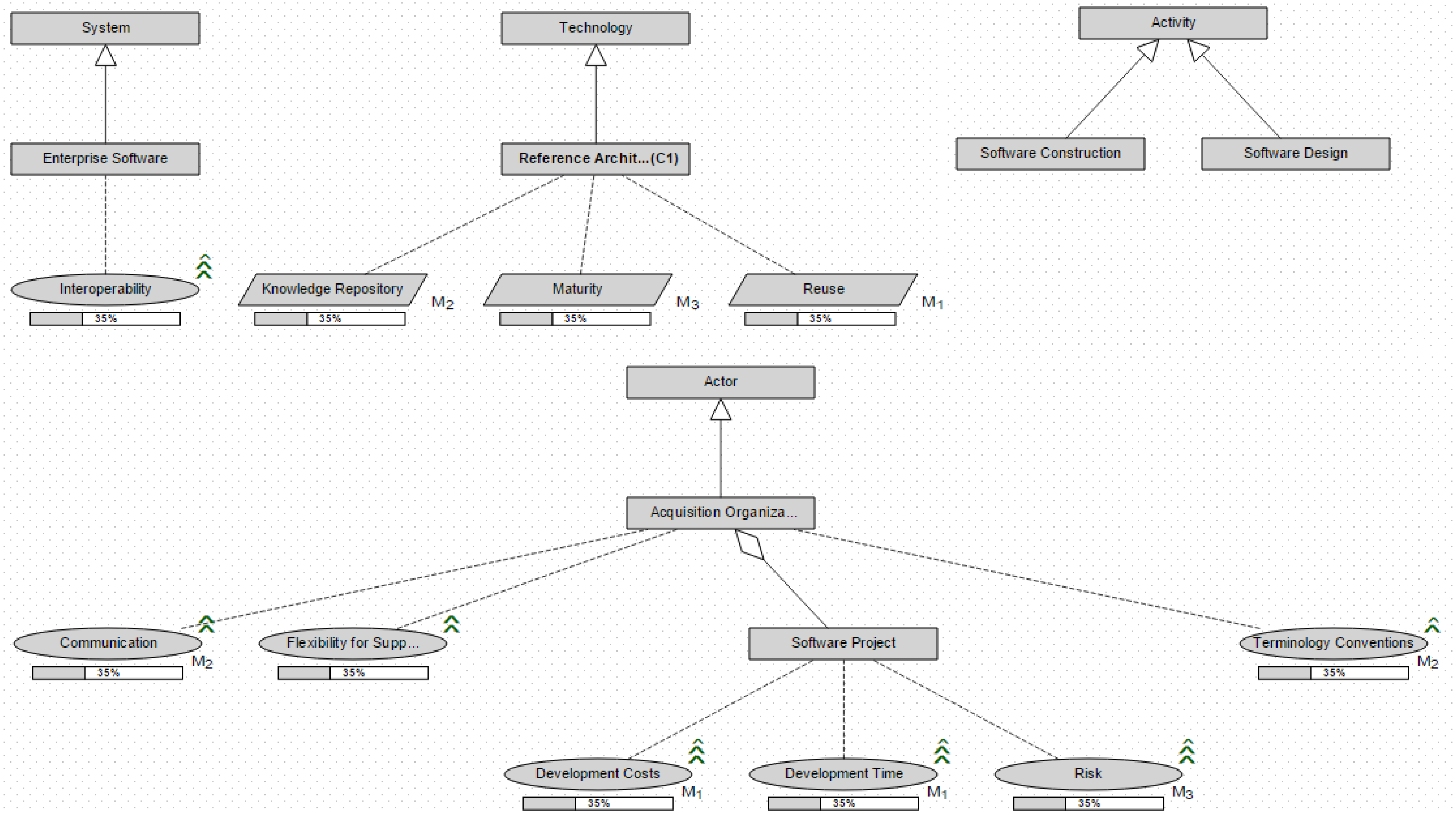


Figure 47 – Evidence model representing study S1 results

Regarding the effect intensity, it is interesting to say that we had to interpret it from the textual descriptions. Thus, if the textual description did not qualify the effect with particular adjectives indicating the intensity, then we chose a default value (e.g., {PO}), rather than a weak (e.g., {WP}) or strong value (e.g., {SP}). Similarly, if the paper gave an ambiguous description we defined a range for the effect intensity (e.g., {WP, PO}). For instance, the 'learning curve' drawback in the study S2 is described as '*additional high or medium learning curve for using the SRA features*'. The belief values were based on the study type and the quality assessment as previously described. Additionally, in survey papers (which also report quantitative evidence), we weighted the belief values with the number of respondents that actually perceived the effect as result of SRA usage. This weighting was performed using the DST discount operation. We used the number of respondents as estimation for the discount value calculated as: 1 - number of respondents for the question / total participants.

8.3.4 Aggregation and analysis

8.3.4.1 Aggregation results

As in the previous synthesis, we summary main aggregation results in a tabular format. Table 26 shows the results after performing the aggregation of evidence on the benefits and drawbacks of SRAs. The individual study with the highest belief for an effect was S4, with 40% belief for the 'Interoperability' effect (see Table 25). However, after aggregating the results from primary studies, some effects caused by SRAs were reinforced. Effects that have higher belief of 40% after the aggregation are shown in bold. After that, we also enumerate the conflict level and the difference between the max value of the belief in individual papers and the gained confidence after the aggregation. The effects are ordered by the difference on the belief after the aggregation.

Table 26 – Aggregated effects of SRA

Effect caused by SRA	**Aggregation Results**				
	#Papers	***Intensity***	***Belief***	***Conflict***	***Difference***
Interoperability	4	{PO, SP}	**74%**	-	34%
Development costs	4	{PO, SP}	**67%**	-	31%
Communication	4	{PO}	**65%**	-	25%
Risk	3	{PO, SP}	**65%**	-	25%
Best practices	3	{PO}	**64%**	-	24%
Learning curve	4	{NE, WN}	**60%**	-	20%
Development time	3	{PO, SP}	**52%**	-	17%
Maintenance cost	2	{PO}	**44%**	-	9%
Productivity	2	{PO}	38%	-	8%
Ease of developing	3	{PO}	35%	-	5%
Alignment	2	{WP, PO}	24%	-	5%
Restriction	3	{NE}	18%	-	5%
Standardization	4	{WP, PO}	43%	-	3%
Latest technologies	1	{WP}	30%	-	0%
Investment	1	{NE}	25%	-	0%
Reliability	1	{WP, PO}	14%	-	0%
Dependability	2	{NE, WN}	12%	-	0%
Reputation	1	{WP}	6%	-	0%
Software quality	2	{NE}	6%	-	0%
Novel design solution	2	{PO}	5%	-	0%
Complexity	2	{SN, NE}	26%	0.02	-1%
Terminology conventions	2	{WP, PO}	31%	0.06	-4%
Flexibility of suppliers	2	{WN, IF}	31%	0.14	-9%

Next, we respectively report the effects that: a) increased, b) slightly increased, c) did not change, and d) decreased their belief after the aggregation:

Effects of SRAs that have their belief increased

Seven effects caused by SRAs increased their belief values after the aggregation. These effects have greater confidence value than any effect before aggregation (i.e., greater confidence level than 40%, see Table 25), and have been reported by at least three out of the five studies. Next, we enumerate these seven effects and their moderators.

SRAs positively - strongly positively improve the *interoperability* of the software systems (74% belief). Studies reported that SRAs: aim at '*interoperability to improve compliance for a given context'* [S1]; '*act as communication center for information exchange*' [S5]; and integrate software into (and become part of) a SRA [S4]. As we can see in the last example, *existing software* in the organization proportionally moderates interoperability.

SRAs positively - strongly positively impact the *development costs* of software projects (67% belief). *Reuse* of common assets proportionally moderates development costs from not having to start from scratch [S1]-[S2].

SRAs positively improve the *communication* inside their acquisition organizations (65% belief). SRA stakeholders share the same architectural mindset, fostering an improved communication, i.e., "*people talk the same language*" [S5]. *Organizational thinking* proportionally moderates such communication: "*when a service-based SRA is implemented, different departments within an organization need to a) share information with other departments, but also b) get things from other department*" [S4]. Also, the role of a SRA as a *knowledge repository* proportionally moderates knowledge transfer and communication. To sum up, SRA aids the understanding of architectural and design principles [S1].

SRAs positively - strongly positively influence the *risk* of software projects (65% belief). The *maturity* of SRA proportionally moderates its risk. Maturity relates to the degree of formality and optimization of processes, from *ad-hoc* practices, to formally defined steps, to managed result metrics, to active optimization of the processes, e.g., "*a mature architecture follows principles for 'good' design, such as high cohesion, high modularity and low coupling*" [S4]. Risk reduction is achieved through the use of proven and partly prequalified architectural elements. The general maturity and experience level associated with SRA also bears the promise of a higher quality end-product [S1]. If no mature architecture exists, designing and introducing a SRA is likely to fail [S4].

SRAs positively improve the use of *best practices* inside their acquisition organizations (64% belief). The studies do not report the type of best practices. This is proportionally moderated by the *maturity* of a SRA.

SRAs negatively - weakly negatively influence the *learning curve* of developers (60% belief). Developers that use a SRA need to learn its features [S2]. As a consequence, '*many engineers have difficulty learning'* some SRAs [S5]. *Organizational thinking* indirectly proportionally moderates the learning curve: '*changing organizational thinking in employees is often achieved through training that takes place when introducing SRAs*' [S4].

SRAs positively - strongly positively impact the *development time* of software projects (52% belief). This benefit is also proportionally moderated by *reuse*, which can lead to shorter development cycles. However, it is not the same effect as development costs, because it refers to lower time-to-market of the constructed software [S1], [S2].

Effects of SRAs that have their belief slightly increased

Six effects have slightly increased their belief. Three of these effects have been reported by only two studies: reduced *maintenance costs* of software projects, improved *productivity* of developers, and *alignment* of applications to an organization's business needs. Since the studies agree on them, these effects have increased their value, but more research is needed to corroborate them.

However, the other three effects were reported by at least three studies. In the case of *ease of developing* and *standardization*, we can see in Table 25 that these effects are stronger for some types of SRAs (see Section 8.3.4.2). In the case of regulative SRAs that *restrict* the development on software systems, the percentage is low because the three studies reporting it gave a really low confidence value, thus, it seems that it is not seen as a very important drawback for practitioners.

Effects of SRAs that have their belief unchanged

Seven effects did not change their belief. Three of them, *dependency* of the software systems over the SRA, propagation of bad *software quality* and wrong decisions of the SRA, and *novel design solutions* are reported by two studies, with different degrees of effect intensity, which did not contribute to increase the evidence level during the aggregation. In fact, the two latter effects have a negligible conflict level of 0.002. We can conclude that the confidence level on these three effects is very low, so they rarely appear in practice, and it seems that they are not considered as fundamental benefits or drawbacks.

The use of *latest technologies*, up-front and migration SRA *investment*, *reliability* of SRA software components used in the software systems, and *reputation* of acquisition organizations have been reported by only one study. It does not mean that these effects caused by SRAs are not important, but that more investigation effort by the research community is needed in other to understand them. Still, some of the effects, as reputation, may depend on the SRA type, *e.g.*, a SRA for a market domain so that other companies may be interested in outsourcing [S5].

Effects of SRAs that have their belief decreased

Three effects have lower confidence level after the aggregation due to contradictory evidence in the single studies. These effects are: a) the *complexity* of the software construction process due to using a SRA; b) how a SRA affects the establishment of *terminology conventions*; and c) how a SRA influences the *flexibility of suppliers* or outsourcing companies that develop software systems based on the SRA. We further discuss the reasons why these effects have decreased their belief in Section 8.3.4.2.

8.3.4.2 Analysis

In this section, we respectively discuss the effects that were present in different SRA contexts, contradictory results, and the utility of the aggregation with respect to SRAs theory.

Effects of SRAs present in different contexts

The context varies among different studies (see Table 24). Still, we have seen common SRA effects reported in different contexts and application domains. This is the case of improved *interoperability*, reduced *development costs*, better *communication*, and higher *learning curve*, which have been reported in four out of five studies without contradictions. These SRA effects, described in previous section, are the strongest results of the aggregation.

Contradictory effects of SRAs from different studies

In all studies, the context was the use of SRAs for the design and construction of software systems. However, these SRAs were of different type, for instance, they had different goals (e.g., standardization and facilitation), targeted several domains (e.g., automotive software and e-government), involved different stakeholders (e.g., vendors and client organizations), and coexisted with different software and constraints (e.g., reference models). Due to their different contexts, there are some effects that are caused by some types of SRAs, but are not present in other types of SRAs. Still, even with those differences, we understood that it was possible to generalize the concept of SRA for software systems, independently from their types, in order to analyze its most prominent effects, and then, examine the conflicting results from the perspective of their contextual differences.

Next, we discuss those effects that have contradictory evidence and form hypothesis to contextualize them.

- Are SRAs *complex* or do they *ease the development* of software systems?

The study S5 reported that AUTOSAR (which is an SRA to standardize automotive software) influences negatively - strongly negatively the *complexity* of the software construction process with 27% belief. However, the study S2 showed that nine SRAs for information systems weakly negatively influence this complexity with 6% belief.

Therefore, we can see different SRAs that differently affect to the complexity of software development. For these two studies, related effects to complexity had different effect intensity and confidence level. For instance, AUTOSAR have worse results (i.e., lower intensity and confidence level) for the effects *ease of developing* and *productivity* that the other SRAs.

The reasons that we posit for this conflict is that AUTOSAR has the goal of standardization of concrete architectures (aiming at system/component interoperability), whereas the SRAs of the study S2 focus more on facilitation of the design of concrete architectures (aiming at providing guidelines and inspiration for the design of systems). Also, the study of AUTOSAR mentioned two moderators of this effect: a) the size of the concrete architecture project, e.g., large projects with many developers and highly interconnected functionality is where using AUTOSAR becomes very tough; b) the existence of a tool environment, e.g., tools that help developers while using an SRA. The first moderator, the size of the project, can be present in both contexts (i.e., standardization and facilitation SRAs). However, we can see that facilitation SRAs tend to include more guidelines and a tool environment to facilitate the development of applications, e.g., user manuals, tool prescriptions and plugins, and sample instantiations [16].

Hypothesis: Complexity depends on the type/goal of the SRA (i.e., standardization and facilitation) and on the guidelines that it delivers to facilitate the development. SRAs that aim to standardize tend to be more complex that those that aim to facilitate software systems construction. Providing a tool environment seems to reduce the complexity for both types of SRAs.

- Is it always positive to establish *terminology conventions*?

The study S1 claimed that '*an SRA can serve as a framework and lexicon of terms and naming conventions'* whereas the study S5 stated that '*AUTOSAR practitioners face problems with term confusion*'.

One reason we posit for this conflict is that although an SRA could aim to establish term conventions, they do not always reach this benefit. For the case of AUTOSAR [S5], documentation is large (more than 100,000 pages), what could discourage users to completely read these lexicons of terms.

Hypothesis: Although an SRA can define a common lexicon of vocabulary, the success of establishing term conventions depends on the design and documentation size. If documentation is not 'digestible', it may lead to terms confusion when stakeholders are not familiar with those terms.

- Do SRAs allow *flexibility of suppliers*?

In the context of organizations that outsource suppliers to develop their software systems, flexibility of suppliers refers to the capability of an organization to change these suppliers. With the effect '*flexibility of suppliers*' there was also a conflict during the aggregation. In the study S1, the authors state: '*An acquisition program backed up by a strong SRA that ensures interoperability and 'form, fit, and function' compatibility promotes flexibility in the choice of suppliers, as well as a lower risk through multi-sourcing*'. However, vendor lock-in moderator of study [S4] shows that '*customers are restricted in changing their system without the involvement of the vendor, despite the use of open standards. Customers try to reduce vendor lock-in, but this is not always possible, given the small market of software vendors in certain domains and the required expertise*'. Therefore, SRA adoption does not guarantee flexibility of suppliers, which also depends on other approaches such as the use of open source.

Hypothesis: Despite the use of open standards, mature architectures, and the construction of knowledge repository, outsourcing the construction of SRA-based software may imply vendor lock-in for organizations, jeopardizing the flexibility of suppliers.

Synthesis contribution to the theory on SRAs

Previous studies reported the effects (i.e., benefits and drawbacks) caused by the use of SRAs, as well as the percentage in which they appeared in practice [S2, S3, S5]. Other works have focused on analyzing the practices and constraints of SRA, and qualitatively reported how they moderate or imply the aforementioned effects [S1, S4]. By aggregating the results, this is the first study considering the percentages given in previous quantitative studies, and explaining how specific characteristics of SRAs moderate their effects on SRAs.

This research synthesis study contributes to the body of knowledge of SRAs bringing stronger evidence of their benefits and drawbacks. For most of the effects, the results followed the trends of previous research. However, for three effects these results were contradictory. This observation helped to see that some effects are not general to all types of SRAs, but rather to specific types and contexts. We believe that the aggregated results points to more generalized perceptions and stronger indications of its applicability. Thus, it is expected that practitioners benefit from these indications to support the decision making in practice. Moreover, the stated hypotheses or even the aggregated results themselves can be target of further studies in the future.

8.3.4.3 Validity

To mitigate the threat of missing important primary studies, we systematically searched empirical studies about the benefits and drawbacks of SRAs. We obtained a set of five studies reporting evidence on real SRAs, which is a high number in SE considering that they report the same effects (i.e., benefits and drawbacks of SRAs). During this process, we discarded studies that only reported opinions, rather than empirically-grounded evidence. Even though we found five studies, more studies are needed to reach definitive results, especially considering the relatively low confidence of the aggregated studies – according to the SSM belief estimation criterion.

We are aware that each selected study poses its own validity threats; therefore, we carefully assessed them together with the studies' context to properly interpret their results. Furthermore, while representing empirical evidence from individual studies, researchers can reflect their own opinion and, therefore, bias the representation. To mitigate these subjective issues, the definition and analysis of each individual evidence model from each selected study was first done by a researcher and validated by another one. The studies S2 and S5, conducted by some of the authors, were modeled by the other authors to avoid bias and not to include "extra" knowledge that was not reported in the papers[34]. Then, the aggregated models were assessed and discussed by the whole team. During this process, we experienced some semantic issues, meaning that different studies referred to the same concept using different terms. This would lead to a wrong aggregation. To avoid this, we created a glossary of terms that was used in the evidence models and kept track of the matching terms.

To improve the interpretation of the aggregated evidence, we used some suggested strategies of SSM. For instance, given that the SSM method does not consider the different size of sampling of different studies, whenever possible, we refined the confidence level of each effect applying the discount of participants that do not mentioned it. In addition, we recorded the diverse context of each individual study, so we could better reflect and understand the aggregated evidence. It is important to note that our aggregated results are based on what the authors reported in their papers. Hence, there is always the risk that important information might not be reported. Anyway, in case of doubts we considered the option of contacting the authors to clarify any issues, but this was not the case in this study.

[34] The authors of the publication where this synthesis was published (Martínez-Fernández *et al.*, 2015b).

Finally, one of the goals of aggregation in SE is to consolidate empirically-grounded knowledge, to increase whenever possible the generalization of the results and the understanding of the contexts that might cause any effect. Given the mixed nature (i.e., qualitative/quantitative) of the assessed studies, the aggregated resulting effects could not be statistically but analytically assessed (Wieringa, 2014). For this reason, it was highly relevant to properly define the individual contexts of the studies, so we could better understand and interpret the diverse effects of SRAs. We paid special attention to identify the mechanism that produced the studied effects (*i.e.*, moderators). This helped us to explain how the SRA characteristics influence the acquisition organizations. All these strategies have increased the confidence of our results.

Our results show that some effects got higher degrees of belief while others did not. It is important to be aware of the correct interpretation of these results. On the one hand, the effects that got higher belief are potentially those that have been further studied and agreed among the studies. On the other hand, those effects that got lower belief values (or even decreased) are those that were just partially approached by the existing evidence (or got contradictory results among the studies). Therefore, these effects are relevant topics that need to be further studied. We highly encourage the SE community to investigate the effects that do not have a high confidence value yet, in order to increase knowledge and consolidation of the benefits and drawbacks of SRA.

8.4 Discussion

Apart from the synthesis results themselves, a major goal of presenting the two studies in this thesis is to discuss the utility of SSM. In this section, we examine several aspects of SSM application in the syntheses presented in this chapter.

8.4.1 Types of evidence aggregated

The first distinctive aspect of the conducted synthesis studies is the diversity of types of evidence used in them. In the UBR synthesis, all studies have similar experiment design commonly found in inspection technique studies – *i.e.*, a comparison of inspection techniques by collecting data about number of defects and observing the inspectors efficiency and effectiveness. With a quantitative nature, it would be probably possible to use statistical meta-analysis to aggregate the studies, but still it was practical for exercising and showing SSM mechanisms and procedures. In addition, as a first application of SSM, it is arguably easier to understand and follow the procedures applied to quantitative studies, as they tend to be more explicit about variables and their effects are expressed in numbers which can be straightforwardly translated to the defined scale.

Still, even with these factors favoring the aggregation, the evidence translation to theoretical structures and their combination using D-S theory used some particular aspects SSM that deserve additional consideration. For the reasons explained in the UBR study, we had to use the SSM dismembering operation. Thus, using the dismembering operation, we were able to analyze UBR effects not comparatively with other technologies, but rather their 'absolute' results, which is what is defined as descriptive effects in SSM. The dismembered descriptive effects can be, then, aggregated normally using SSM definitions. This kind of aggregation is usually not desired in many situations, because as some researchers say 'it is like comparing apples to oranges', but as SSM uses a *Likert* scale, a general trend of effects intensity direction (i.e., positive or negative) can be useful as a first general analysis. Therefore, the dismembering operation, although its disadvantages and limitations, can be used to enlarge the possibilities of combinations for aggregation as, for instance, Table 27 shows for the UBR. Descriptive theoretical structures are obtained from the comparative studies S1, S2, S3, and S4.

Table 27 – Different combinations for aggregation using the dismembering operation

Group of studies for aggregation	Type of aggregation	Technologies involved
S1	Comparative	UBR x Ad hoc
S2, S3, S4	Comparative	UBR x CBR
S1	Descriptive	Ad hoc
S2, S3, S4	Descriptive	CBR
S1, S2, S3, S4	Descriptive	UBR

In the SRA study, we find a completely different scenario with evidence coming from different research methodologies: case study, survey and a kind of focus group. Using the strategies defined in SSM (Chapter 5), particularly regarding the identification of constructs for the theoretical structures and being aware of differences in how causality manifest in qualitative (*e.g*, case study and focus group) and quantitative (*e.g.*, survey) studies. Furthermore, even in the face of the heterogeneity found in the evidence selected for SRA we are able to conceptualize about the studies' context and to aggregate their results. In the beginning of this thesis, we argued about the diversity in evidence types in SE as one important issue to be addressed in research efforts directed to research synthesis and knowledge translation in SE. In this aspect, considering the characteristics of evidence aggregated in both studies, it seems that SSM is capable of translating evidence to a unified representation and producing an integrated result from them.

The synthesis studies were also able to show the SSM ability in aggregating multiple effects in a single aggregation. A direct consequence of this, as some effects are not reported in all studies, is that the analysis of results must consider not only the belief values found, but also the number of studies that were aggregated. We have done that in both studies by informing the number of studies for each effect in Table 22 and Table 26. Apart from this aspect in the analysis of synthesis results, other aspects are discussed in the next section.

8.4.2 Flexibility in the results analysis

SSM deliberately does not give detailed guidance on how the aggregated results should be interpreted, analyzed, and reported. We believe that this kind of direction is dependent of the research goals and context of the selected evidence for the synthesis study. Thus, it is up to the researcher to decide how the results are going to be interpreted to answer the research questions based on abstract thinking skills and analytic reasoning. For this, SSM specifies in details the semantic constructs of the representation and the mechanics of aggregation using the D-S theory. In addition, it gives basic orientation on some basic aspects that should be present in any analysis of an aggregation using SSM as how to report and interpret a belief value, how to deal with conflicts, examples of how effects can be explained, and the importance of making explicit the validity threats. Still, we also understand that any kind of further guidance for analysis could not be formulated without an in-depth comprehension of SSM application to a broad range of domains by several different researchers, which is left as future work. It is even possible to reach a conclusion that this detailed guidance is not necessary at all.

In fact, this absence of detailed guidance was precisely what allowed the analysis of both studies to be significantly different. Clearly, the primary motivation for this difference was explained in the previous section – *i.e.*, the types of evidence aggregated. However, we could also point to the utilization of SSM by an external researcher, who could bring his own perceptions and interpretations to the analysis. From the two experiences with SSM, we could observe in what aspects analysis differentiate. In general terms, the analysis is directly influenced by the input that can be extracted from individual studies. Quantitative studies usually do not present why effects occur, but only what are the effects and which are significant. Qualitative studies, on the other hand, usually have this kind of explanation. Thus, when interpreting the aggregation results, the researcher can decide if he/she wants to use that information to explain effects' mechanisms in the analysis of the synthesis results.

Interestingly enough, this is in consonance with what participants of the SSM experimental study (Chapter 5) reported. They have diverse opinions about the question asking if theoretical structures were enough to interpret aggregation results or if it was necessary to read the original papers from which the aggregation was extracted. It seems that such division is explained by the fact that participants used SSM for both quantitative and qualitative studies (two of each type).

Another aspect that can affect the analysis is associated with the number of available evidence. If the number of studies is enough high, researchers tend to be more interested in the general trend (or a 'mean') than in detailed explanations. This is one of most important aspects characterizing more qualitative analysis from more quantitative ones (Mahoney and Goertz, 2006). Therefore, when there are several studies reporting an effect, aggregation results tend to be more generalizable, *i.e.* more detached from the studies from which the aggregation was obtained. As a result, the more non-conflicting results reporting the same effect, more close to a 'law' the aggregation results are and less interested researchers are in giving details about (the assumed to be not significant) differences of effect in different contexts.

8.4.3 External researcher experiences

Although SSM was already used by other researchers (Chapter 6), the SRA synthesis study was the first time SSM was used outside controlled settings by a different researcher besides the author. It is true that the SRA study was not an independent application of SSM as the author took part in the study as well. In addition, this collaboration was not organized as a systematic observation of SSM application. Still, we understand it is important to discuss this experience from the perspective of the external researcher from ESSI group. In this section, we reference to the author as AU and to the external researcher as ER.

Before starting the synthesis activities, we first had an informal training session about SSM. In this session, with duration of about three hours, AU introduced research synthesis methods, possible advantages and limitations of SSM, evidence representation with theoretical structure, an overview of the aggregation procedure and an introduction to the Evidence Factory tool facilities. The domain of interest, in this case SRA, was also covered by ER with discussions about the importance of the research theme and a preliminary view of the evidence already found – the systematic review protocol was already concluded. In fact, this first session was an initial understanding if SSM was appropriate to the synthesis expected goals. It was done in person at UFRJ/COPPE. From that point, further sessions were carried out remotely using regular communication tools for voice and remote desktop sharing.

In the following session, we structured our activities according to the SSM steps. As the ER did not have experience with the evidence representation, we agreed that the AU should first translate one of the studies to the evidence representation while ER could read some material about SSM – Santos and Travassos (2013) and Santos *et al* (2015). After the first evidence model created, we discussed again about the representation semantics and the reasoning behind created concepts and their relationships. We also took this opportunity, given that the model was developed in the Evidence Factory tool, to have a more throughout presentation about the related functionalities present in the infrastructure. Then, the activities followed as described in Section 8.3.3, where AU and ER translated the evidence models separately and then reviewed each other's work. We had further discussion sessions to homogenize our understanding about the developed concepts. These sessions also included some attempts to aggregate the evidence, where we identified conceptual conflicts that were, in fact, caused by misunderstandings about the used concepts.

Apart from these conceptual misunderstandings, which are inherent to this kind of activity, ER was able to model three of the five studies using the Evidence Factory tool. The models produced by ER did not have almost any semantic issues – *i.e.*, a semantic construct used for another purpose. This observation is aligned with the low number of faults present in the models created in the experimental study (Chapter 6, see data in Table 12). Arguably, in this particular case, we can hypothesize that the tool formal meta-model avoided some faults. Yet, AU was able to notice ER level of understanding about the models. Particularly during the sessions in which conceptual misunderstandings were resolved, it was necessary to elaborate arguments based on the understanding of the models such as differentiating moderators form effects or distinguishing the importance of contextual aspects.

Regarding the tool, ER seemed comfortable with the range of functionalities available in the tool, especially considering, according to him, it was developed within the scope of a research. Nevertheless, ER missed some important features including support for collaborative synthesis (we had to share a user account to edit the models) and an 'undo/redo' feature (evidence aggregation had to be performed at once or, otherwise, the aggregation had to start from the beginning again). It was also interesting to observe how the tool support makes the D-S theory application transparent. Taking the complexity of calculations away, the researcher does need to understand how it works, but rather only interpret the combination results in terms of belief values. Interestingly enough, even the notion of belief values do not need to be grasped by researchers, as it can be interpreted within the general notion of evidence strength commonly used in the evidence-based practice.

Corroborating these observations, the analysis of aggregation was structured in most part following ER decisions. For instance, it was ER decision to split the aggregation results into four groups according to the confidence strengthening or weakening after combination. Moreover, the hypotheses explaining conflicting results was ER decision – even though explanation is one alternative for dealing with conflicts in SSM, the organization in questions and the reasoning developed in the answers came from the ER skills. As a matter of fact, the previous section was motivated by and examined several aspects observed in this opportunity.

8.4.4 Tool support: the Evidence Factory

A computerized infrastructure is a major element of the unified view for evidence in SE presented in the beginning of this thesis. This stems from its support for knowledge translation using SSM and the features related to scientific knowledge engineering that were incorporated into the tool. In this chapter, both studies used tool support from Evidence Factory.

From the SSM perspective, all supporting facilities were used. At the current stage of implementation, SSM support is focused in four aspects: (i) evidence selection, (ii) studies quality assessment, (iii) viewing and editing theoretical structure models, and (iv) evidence aggregation. Together, these features cover three of the five SSM steps, leaving out the first and the last steps, which have only basic support to register essential information. Apart from usability issues, we found these facilities to be useful to conduct the two syntheses studies. We can cite several reasons. The first reason we can mention is the organization of the synthesis information. For instance, keeping the trace between studies (*i.e.*, papers), quality assessment, theoretical structures, aggregation groups, and how conflicts were resolved, can be cumbersome (Appendix C is an example of this issue different sets of information including the theoretical models, concepts and relationships description, and quality assessment data). In addition, although usually a neglected aspect, viewing and editing theoretical structure models in a Web environment is particularly appealing if we think that it makes a fairly complex tool available within a browser, which can be used in almost any client hardware. And the conflict resolution mechanism is fundamental for SSM Structured Aggregation as it automatically detects any kind of incongruences between the models and let the researcher choose and apply the appropriate resolutions. Again, not only these features facilitates the execution of SSM steps, but most importantly they allow registering information associated with these activities into the computerized infrastructure for further usage or application in other syntheses.

From the Scientific Knowledge Engineering perspective, the Evidence Factory tool misses several aspects present in the computerized infrastructures reviewed in Chapter 4, but it is still able to cover important elements of SKE. As mentioned in the previous section, the knowledge representation formal model for evidence and the D-S theory implementation have important application in the process of modeling scientific knowledge (i.e., creating the theoretical structures) and making inferences about aggregation (i.e., combination of evidence effects).

8.4.5 Transparency and subjectivity

Transparency is the degree to which the activities performed in an investigation using a research method are easily seen and understood by researcher peers. Usually, the more transparent is a research method, the less likely it will be influenced by subjective decisions and reasoning.

Both analytically and empirically (including the experimental study in Chapter 6), SSM has relatively less and more transparent phases. Most subjectivity of SSM is inherited from its interpretive characteristics. Interpretive synthesis methods make subjectivity inherent to the research process as they require an extensive analysis and interpretation of raw data (usually text) to develop concepts or any other kind of organized explanation of the investigated theme. As such, they are influenced by the researcher abstraction skills and knowledge about the topic of interest to improve the quality of interpretations.

The interpretive disposition of SSM essentially manifests itself in the modeling of theoretical structures. Still, considering the degree of subjectivity that is expected from this kind of activity, we found that models created have some degree of similarity. In the experiment from Chapter 6, models created by subjects achieved 55% of similarity (considering synonyms). In our experience with an external researcher, we had some misunderstandings between the models created, but in general, our perception is that interpretations were quite similar. Furthermore, it is interesting to notice that the identification of concepts and relationships is an iterative process, and the modeling of an evidence can be a trigger to review the other models. This is important to make concepts and evidence structures more consistent, and particularly important for evidence synthesis.

The integrative disposition of SSM, on the other hand, are relatively more transparent. The identification of incompatibilities between the models for aggregation, the quality assessment of studies, belief values estimation, and the evidence combination using D-S theory can all be executed with relative degree of systematicity. Even in the results analysis, despite the flexibility tolerated by SSM, the interpretations are grounded in the aggregation results (belief values).

Thus, we believe that SSM is a sufficiently transparent research method considering the goals of research synthesis studies of making explicit how the aggregated evidence has contributions from different pieces of evidence to answer the defined research question.

8.5 Conclusion

The best scenario to experimentally evaluate something is to have a controlled environment in a real world case. This is challenging in most situations as the risk of failure is usually intolerable or the costs involved do not worth the potential results. Particularly in scientific context, research methods directly affects the results observed in an investigation. More precisely, it is the lens through which researchers try to observe and explain the world, making the research method intrinsic to the investigation that uses it. Thus, when new research methods are introduced they have to overcome a point in time where it is still unknown if the 'lens' will augment what can be 'seen'. In addition, it is necessary to determine if it can be appropriately used by researchers. These two dimensions can be referred to as utility and ease of use. The ease of use was evaluated on the experimental study presented in Chapter 6, although it was not evaluated in its full extension, since we only evaluated if participants were able to apply the method and use the theoretical structure representation for translating evidence.

The utility of research methods, as discussed earlier in the beginning of this thesis, is usually examined analytically. The scientific community usually associate utility to how transparent is a method so that the results can be traced to the activities performed in the study and to how it brings new means to analyze the problem at hand. In this chapter, we tried to address both aspects. In the first study, we focused in showing how the five steps are used to conduct a research synthesis. We used a well-known domain so that we could focus in how SSM is applied, and how the synthesis can be traced to the steps. In the second study, on the other hand, our focus was directed towards the synthesis result itself. As it could be seen, results of the synthesis are more detailed, which serve as an example of the utility of findings obtained with SSM.

9 Conclusion and Future Work

In this chapter, we present the conclusions, emphasizing the main contributions of this research. Additionally, we present some limitations and open questions not addressed in this thesis. Finally, the way ahead is outlined in order to show possible future work to come as result of the current achievements.

9.1 Final considerations

In this thesis, we addressed the knowledge translation topic in SE. This topic is central to the advancement of evidence-based practice, since it defines how the body of knowledge of an area is made available to its users (*i.e.*, researchers and practitioners). Based on the observation that SE research is subject to a high heterogeneity of primary studies along with a common perception of communication issues between practitioners and researchers when trying to leverage from the body of knowledge, we built our view focused on knowledge representation for evidence. Upon this perspective, we conceptually devised the notion of scientific knowledge engineering, proposed a research synthesis method, and developed a computational infrastructure for evidence modeling and synthesis.

As knowledge translation is a community wide concern, fully addressing this topic in a single research effort would be ambitious. Rather, we opted to propose its building blocks and indicate their utility. For that, we focused on the main aspects of knowledge translation – namely, synthesis of knowledge, appropriate forms for communicating it, and adequate support for interaction between its users. This was essentially accomplished through the Structured Synthesis Method and the Evidence Factory infrastructure. Linking these two proposals was the idea that we should define a unified view for evidence in SE. This unified view was established based on a formal knowledge representation for evidence, which was grounded on the theory notion and complemented with the D-S theory. Moreover, since formal knowledge representation played an important role in this research, we further developed this theme by proposing the scientific knowledge engineering as a research topic concerned with this issue.

Therefore, even though arguably the main outcome of this research is the proposed synthesis method, we still delimited our problem within knowledge translation. By doing so, we tried to bring focus not only to the research synthesis method, but also to evidence representation and communication issues, which are important aspects of evidence-based practice. Consequently, although SSM has unique features as a research synthesis method for SE, we put it on equal level of importance with knowledge representation formalization and the construction of a computational infrastructure (*i.e.*, SKE). Only together, they form a mean to achieve the view for knowledge translation established in the beginning of this thesis. In that regard, they define how the body of knowledge in SE can be maintained under the evidence-based practice paradigm. Both have concerns for knowledge creation and application, which are cornerstones of knowledge translation. SKE shows how knowledge representation can be useful not only for organizing scientific knowledge into formal representation, but also in how knowledge-based systems can support researchers and practitioners in making sense from the body of knowledge. SSM complements knowledge creation determining the process through which evidence is modeled using the defined knowledge representation and contributes to knowledge application systematizing how collected evidence are synthesized turning the body of knowledge more comprehensible.

9.1.1 Research question answers

As a conclusion, although scattered through the chapters of this thesis, we resume the answers for the defined research questions:

- **RQ1**: Is it possible to have a formal representation capable of translate (i.e., represent) results of primary studies in Software Engineering?

  We have used a theory diagrammatic representation as an evidence representation and defined a formal model based on its definitions. Moreover, SSM determines the set of activities that must be performed to model evidence using the proposed representation with particular emphasis on research synthesis.

  An experimental study was designed to observe if researchers are able to understand and use the representation by extracting evidence from scientific papers following a well-defined process. In addition, two worked examples were presented with the purpose of describing SSM, which in the case of this specific research question shows how a formal knowledge representation was used to represent evidence from two real SE themes.

Our conclusion is that it is not only possible to define formal representations for representing evidence in SE, but also we hypothesize that it has beneficial influence over knowledge translation.

- **RQ2**: Does such formal representation can help organizing evidence in SE by supporting the management, search and aggregation of primary studies' findings?

  We conceptually developed the notion of SKE with the particular goal of analyze this research question. It was shown that there is an increasing interest in this theme, although research efforts have only begun recently. Apart from more than 20 proposal related to the usage of formal representations for scientific knowledge in the technical literature, we designed and constructed a computational infrastructure based on the formal representation for SE evidence. Furthermore, the D-S theory was used as uncertainty formalism to allow evidence aggregation.

  We were able to use this computational support to manage and aggregate evidence in the two conducted research synthesis studies. However, we have not evaluated the search facilities, as the experimental study was conducted before the development of the computational infrastructure.

  Our conclusion is the characterization that computational support can be designed to that end based on formal representation, but we still need further investigations to show if the computational support expected benefits do indeed exist.

9.2 Contributions of this research

We enumerate the contributions of this thesis referencing the three research domains defined in the beginning of this Thesis (Figure 2):

- Methodological domain
  i. Analysis of the knowledge translation in SE from the perspective of knowledge structure instead of focusing on the activities necessary to that end, particularly considering:
     - A view of knowledge systematization as a factor for body of knowledge maintenance improvement with consequences for communication among and between researchers and practitioners (Santos and Travassos, 2015);

        - Indication of the importance of discussing this theme in SE given the heterogeneity of evidence in the area, which is usually attributed to its dual technical and social nature;
    - Indication of computational support as one mean to support knowledge translation (based on the knowledge systematization premise);

ii. Definition of a research synthesis method in SE: the Structured Synthesis Method (Santos and Travassos, 2013);
    - Characterization of a systematic process for research synthesis based on formal knowledge representation and existing research synthesis methods;
    - Drawing attention to knowledge representation as an intrinsic aspect of research synthesis. This aspect does not seem to be focused in any research synthesis method proposal; even though to our understanding, it is implicitly present in any of them. Thus, in a meta-research perspective, this thesis shows it is possible to conceptualize about research synthesis in a different order. We first worked on a knowledge structure for synthesis and then organized the set of activities to manipulate it. Indeed, although we have not presented an analysis of this issue, the fact that SSM process is mostly formed by activities from other methods (Section 5.2) is an indication that these activities can be shared among different methods;

iii. Usage of diagrammatic representation as an instrument for evidence comparison in research synthesis. There are research synthesis methods that use graphical representation to display the studies outcomes (e.g., thematic synthesis), but we are not aware of any method that uses diagrammatic representation as part of the method's procedures (in this case, evidence comparison);

iv. Usage of theory notion as epistemological and conceptual foundation for evidence representation and research synthesis in SE. Theory notion is gaining relevance in SE in the recent years (e.g., Johnson *et al*. (2013) and Ralph *et al.* (2013)). Still, discussions about research synthesis with/for theories seems to be inexistent in SE;

v. Usage of the D-S theory as uncertainty mechanism to evaluate the degree of aggregation consensus and estimate the confidence in the results. To the best of our knowledge, this is the first usage of D-S theory for research synthesis purposes;

vi. Conduction of an experimental study to evaluate the feasibility of the proposed research synthesis method. Experimental evaluation of research synthesis methods are seldom conducted and thus represent an important aspect of this research;

vii. Construction of a computational infrastructure to support evidence search, modeling, and aggregation in SE (Santos *et al.*, 2015);

- Conceptual domain

viii. Conceptualization of the scientific knowledge engineering as a research topic concerned with addressing the increasing volume of scientific findings by providing computational infrastructures to support researchers in maintaining bodies of knowledge (Santos and Travassos, 2015);

  - Comparison of different strategies for computational manipulation of scientific knowledge, namely, data-intensive approaches, computational discovery of scientific knowledge, and scientific knowledge engineering;
  - Identification of sources for characterization of scientific knowledge structure, particularly from epistemology and domain (*i.e.*, area of research);
  - Organization of a body of knowledge regarding SKE: review of existing works with SKE features and determination of main necessary steps for constructing a computational infrastructure for scientific knowledge (Section 4.4.1);

- Substantive domain

ix. A research synthesis study regarding usage-based reading inspection technique (Santos and Travassos, 2013);

x. A research synthesis study regarding software reference architecture (Martínez-Fernández *et al.*, 2015b).

9.3 Limitations

Working at the edge of epistemological interpretations, but also concerned with practical definitions to support researchers (and practitioners in the future), we are able to identify limitations from the more philosophical standpoint to the more practical research activities and decisions developed in this thesis:

- The research problem regarding knowledge translation was not fully addressed. Particularly, we cite:

  - The practitioners cycle of the knowledge translation conceptualization defined in Figure 1 was not object of study in this thesis;
  - Knowledge application was partially addressed as the focus was directed towards knowledge creation, which is noticed by the relevance of SSM in this work;
- We were not able to precisely define the knowledge representation expressivity in terms of what kinds of evidence it is able to represent. This seems to be a difficult problem as Chapter 4 shows how this issue is neglected by most proposals of knowledge representation for scientific findings. We tried to analytically examine this matter by establishing a relation with epistemological aspects of research question types from Easterbrook *et al.* (2008) and indicating how the causal notion manifest both in qualitative and quantitative research (Mahoney and Goertz, 2006);
- The belief value estimation based on study type and quality by strictly segregating the probability domain [0, 1] into four parts can lead to some distortions. Although evidence types are usually used as a determinant for evidence strength, there can exist observational studies which generate stronger evidence than controlled ones depending on their respective design and quality (Atkins *et al.*, 2004);
- The computational infrastructure implementation has several missing features from its specification (Appendix D), particularly important requirements regarding knowledge translation including social and collaborative features. Furthermore, it was not experimentally evaluated and thus there is no evidence about its expected benefits;
- The SKE delineation was based on a non-systematic literature review. Although a tentative search string returned over 13000 papers, we believe that research efforts dedicated to select and review those papers would disclose important findings in this topic. Moreover, there is not significant guidance on technical aspects of SKE such as algorithms for inferences or techniques for defining knowledge representations – only the basic steps regarding SKE projects were presented;
- The study presented in Chapter 6 has an exploratory nature and thus only give an indication about the feasibility of SSM. Several aspects were not considered in its design including the comparison with other methods and knowledge application (*i.e.*, are researchers and professionals able to apply the knowledge embedded in the evidence representation?);

- A limited number of worked examples. Only two applications of SSM were presented, which narrowed down the discussion regarding overall SSM applicability and its limitations. Still, as we discussed in the beginning of this thesis, the acceptance of the synthesis results by SE community is an indication of the method utility. We particularly highlight the software reference architecture study (Martínez-Fernández *et al.*, 2015b), since it was published within the Experimental Software Engineering community.

9.4 Open questions and future work

As a work concerned in indicating the essential elements for establishing an alternative view for knowledge translation, there are several opportunities to further investigate these elements individually or the established view as a whole. Furthermore, these opportunities are associated with the scope delimiting decisions taken in this Thesis. They also unfold future works.

Does a formal knowledge representation affects knowledge translation effectiveness? We tried to examine this issue by showing the feasibility of proposing a knowledge representation with the specific purpose of supporting knowledge translation. However, as it was discussed in several occasions in this Thesis, the answer to this question involves more than the technical aspects of designing a knowledge representation and the associated activities for creation and application of this knowledge. We believe the community will address this in a similar way systematic reviews were worked out. First, guidelines and definitions for its application in SE were proposed (e.g., Kitchenham (2004) and Biolchini *et al.* (2005)). Then, we saw a wide community adoption demonstrating an understanding that it produces useful outcomes for the area (e.g., Kitchenham *et al.* (2009) and Silva *et al.* (2011)). And later, evaluations and experiences were reported about its application (e.g., Imtiaz *et al.* (2013) and Zhang and Ali Babar (2013)).

How can practitioners be involved in the knowledge translation cycle defined in this work? Complementing the previous one, this question was purposely left as a future work due to scope restrictions. It is necessary to determine the activities to the involvement of practitioners in knowledge translation. Both knowledge creation and application should be covered. As a suggestion, our starting point for these definitions would be using continuous improvement cycles (Salo and Abrahamsson, 2007). They are usually the moment where software teams reflect about their practices and discuss about improvement decisions. Thus, they represent a convenient moment to create or apply knowledge (*i.e.,* evidence) about software practices. Another important aspect regarding this issue is defining the decision making process supported by theoretical structures. For instance, should there be a belief value threshold for determining the utility or applicability of a software technology?

Does a diagrammatic representation have benefits over other kinds of knowledge representations for knowledge translation? While this assumption was taken from other works (*e.g.*, Worren *et al.* (2002)), which discuss this theme in general terms, there is still no evidence about the use of diagrammatic representation for the specific purpose of knowledge translation or research synthesis. A translation to a textual or tabular format can be sought as well, since they could balance diagrammatic deficiencies.

How knowledge representation can be designed for knowledge translation, research synthesis, or to organize the body of knowledge? As mentioned in this thesis, the discussion about SKE focused in first indicate 'what' is necessary to design knowledge representations and construct computational infrastructures. Hence, future works should investigate and organize 'how' this can be performed, particularly in terms of the technical aspects related to identify the appropriate algorithms, technologies, and methodologies available to that end.

How does SSM compare to other research synthesis methods? The study presented in this thesis only focused in investigating the method feasibility, by observing if researchers are able to follow the SSM process and to use its diagrammatic knowledge representation. Moreover, the worked examples describe the application of SSM individually – not in comparison to other methods. Thus, there is still no indication about how SSM compare with other methods. There are several aspects that can be compared: (i) do SSM produce results comparable to other synthesis methods (*i.e.*, do the results have the same trends particularly considering the usage of the D-S theory?), (ii) when SSM should be chosen in lieu of other methods and what must be taken into account to reach this decision?, and (iii) how SSM compares in terms of ease of use and scalability? One possible study design to investigate this theme can be inspired in the work of Dieste *el al.* (2011).

Is knowledge structure a distinctive element of research synthesis methods? If this is true, what are the 'canonical' set of synthesis activities? From our experience in combining knowledge representation and research synthesis, we hypothesize if the knowledge structure is the main element for differentiating research synthesis methods. This is based on the view that activities of a research synthesis method are determined by knowledge structure. In other words, according to this hypothesis, we could say there are activities that can not be performed depending on the representation at hand. Thus, they should be defined after determining the knowledge representation. Clearly, arguing on the opposite direction, it also could be the case that knowledge representations are partitionable into smaller units, which could be shared among different research synthesis methods. However, while we could extract activities from other methods to compose SSM, we were not able to identify this possibility for knowledge representation based on the review presented in Chapter 4.

How other evidence types can be represented in SE? The knowledge representation used in this Thesis is focused in representing investigations concerned in answering causal questions (Easterbrooks *et al.*, 2008). It is still open to future works how evidence related to other kind of questions can be represented. We have particular interest in investigating and design knowledge representation for detailed explanatory mechanisms (Glennan, 2005). In our view, representations can coexist and complement each other not only in representing other types of information (e.g., contextual and methodological information), but also in different level of abstraction or granularity (e.g., explanatory and descriptive information).

Is SSM applicable to other scientific domains? Apart from the archetypes, the theoretical structure is a reasonably generic knowledge representation. It has the basic elements commonly enumerated in discussions about the theory notion (Reynolds, 1971, Bacharach, 1989, Lynham, 2002, Walker and Avant, 2004, Gregor, 2006, Sjøberg *et al.*, 2008, Pawar, 2009) and it models essential relationships among concepts (Thagard and Nowak, 1990). Thus, the investigation regarding the adaptation and application of SSM to other scientific domains is an interesting research path. The first step in this direction is evaluating the feasibility of defining archetypes for other disciplines. After that, the research methodology defined in this Thesis can be followed, by performing real worked examples and feasibility studies with researchers.

REFERENCES

ANGELOV, S.; GREFEN, P.; GREEFHORST, D. A framework for analysis and design of software reference architectures. **Information and Software Technology**, v. 54, n. 4, p. 417–431, 2012.

ATKINS, D.; BEST, D.; BRISS, P. A.; et al. Grading quality of evidence and strength of recommendations. **BMJ (Clinical research ed.)**, v. 328, n. 7454, p. 1490, 2004.

AUERBACH, C. F.; SILVERSTEIN, L. B. **Qualitative data: an introduction to coding and analysis**. New York: New York University Press, 2003.

BACHARACH, S. B. Organizational theories: Some criteria for evaluation. **Academy of Management Review**, v. 14, n. 4, p. 496–515, 1989.

BAIROCH, A. The future of annotation/biocuration. **Nature Precedings**, 2009.

BARGA, R.; GANNON, D. Scientific versus Business Workflows. In: I. J. Taylor; E. Deelman; D. B. Gannon; M. Shields (Orgs.); **Workflows for e-Science**. p.9–16, 2007. Springer London.

BARLEY, S. R.; MEYER, G. W.; GASH, D. C. Cultures of Culture: Academics, Practitioners and the Pragmatics of Normative Control. **Administrative Science Quarterly**, v. 33, n. 1, p. 24, 1988.

Special Issue on General Theories of Software Engineering - Science of Computer Programming. .Disponível em: <http://www.journals.elsevier.com/science-of-computer-programming/call-for-papers/special-issue-on-general-theories-of-software-engineering/>. Acesso em: 23/10/2015.

BASILI, V.; CALDIERA, G.; MCGARRY, F. et al. The software engineering laboratory: an operational software experience factory. **Proceedings of the 14th international conference on Software engineering**, ICSE ’92. p. 370–381. New York, NY, USA: ACM. doi: 10.1145/143062.143154, 1992.

BAUER-MEHREN, A.; FURLONG, L. I.; SANZ, F. Pathway databases and tools for their exploitation: benefits, current limitations and challenges. **Molecular Systems Biology**, v. 5, n. 1, p. n/a–n/a, 2009.

BECHHOFER, S.; BUCHAN, I.; DE ROURE, D.; et al. Why linked data is not enough for scientists. **Future Generation Computer Systems**, Special section: Recent advances in e-Science., v. 29, n. 2, p. 599–611, 2013.

BERNERS-LEE, T.; HENDLER, J. Publishing on the semantic web. **Nature**, v. 410, n. 6832, p. 1023–1024, 2001.

BEYTH-MAROM, R.; FIDLER, F.; CUMMING, G. Statistical Cognition: Towards Evidence-Based Practice in Statistics and Statistics Education. **Statistics Education Research Journal**, v. 7, n. 2, p. 20–39, 2008.

BICKLE, J. Connectionism, eliminativism, and the semantic view of theories. **Erkenntnis**, v. 39, n. 3, p. 359–382. doi: 10.1007/BF01128508, 1993.

BIOLCHINI, J.; MIAN, P.; NATALI, A.; TRAVASSOS, G. H. **Systematic Review in Software Engineering**. Federal University of Rio de Janeiro (UFRJ/COPPE), 2005.

BJØRNSON, F. O.; WANG, A. I.; ARISHOLM, E. Improving the effectiveness of root cause analysis in post mortem analysis: A controlled experiment. **Information and Software Technology**, v. 51, n. 1, p. 150-161. doi: 10.1016/j.infsof.2008.02.003, 2009.

BLOCH, I. Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account. **Pattern Recognition Letters**, v. 17, n. 8, p. 905–919, 1996.

BOOTH, A. Evidence-based practice: triumph of style over substance? **Health Information & Libraries Journal**, v. 28, n. 3, p. 237–241, 2011.

BRINBERG, D.; MCGRATH, J. **Validity and the Research Process**, SAGE Publications, 1985.

BRITTEN, N.; CAMPBELL, R.; POPE, C. et al. Using meta-ethnography to synthesise qualitative research: a worked example. **Journal of Health Services Research & Policy**, v. 7, n. 4, p. 209–215. doi: 10.1258/135581902320432732, 2002.

BUDGEN, D.; TURNER, M.; BRERETON, P.; KITCHENHAM, B. Using Mapping Studies in Software Engineering. **Proceedings of PPIG Psychology of Programming Interest Group**, p.195-204, 2008. Lancaster University, UK.

BUDGEN, D.; KITCHENHAM, B.; BRERETON, P. The Case for Knowledge Translation. **2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement**, p.263–266, 2013.

BUNGE, M. How Does It Work? The Search for Explanatory Mechanisms. **Philosophy of the Social Sciences**, v. 34, n. 2, p. 182–210. doi: 10.1177/0048393103262550, 2004.

BUNGE, M. **Scientific Research I: The Search for System**. Springer, 1967.

BYLANDER, T.; CHANDRASEKARAN, B. Generic tasks for knowledge-based reasoning: the “right” level of abstraction for knowledge acquisition. **International Journal of Man-Machine Studies**, v. 26, n. 2, p. 231–243, 1987.

CALLAHAN, A.; DUMONTIER, M.; SHAH, N. H. HyQue: evaluating hypotheses using Semantic Web technologies. **Journal of Biomedical Semantics**, v. 2, n. 2, p. 1–17, 2011.

CAMERON, D.; BODENREIDER, O.; YALAMANCHILI, H.; et al. A graph-based recovery and decomposition of Swanson’s hypothesis using semantic predications. **Journal of Biomedical Informatics**, v. 46, n. 2, p. 238–251, 2013.

CHARMAZ, K. **Constructing Grounded Theory: A Practical Guide through Qualitative Analysis**. Pine Forge Press, 2006.

CHANDRASEKARAN, B.; GLASGOW; NARAYANAN, N. H. **Diagrammatic reasoning: cognitive and computational perspectives**. Menlo Park, Calif.; Cambridge, Mass.: AAAI Press ; MIT Press, 1995.

CHARTERS, S.; BUDGEN, D.; TURNER, M.; et al. Objectivity in Research: Challenges from the Evidence-Based Paradigm. **Australian Software Engineering Conference**, p.73–80, 2009.

CHUA, C. E. H.; STOREY, V. C.; CHIANG, R. H. L. Deriving knowledge representation guidelines by analyzing knowledge engineer behavior. **Decision Support Systems**, v. 54, n. 1, p. 304–315, 2012.

CILLIERS, P. Knowledge, limits and boundaries. **Futures**, v. 37, n. 7, p. 605-613. doi: 10.1016/j.futures.2004.11.001, 2005.

CIOLKOWSKI, M. What do we know about perspective-based reading? An approach for quantitative aggregation in software engineering. **Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement**. p.133–144. Washington, DC, USA: IEEE Computer Society. doi: 10.1109/ESEM.2009.5316026, 2009.

CLARKE, K. A.; PRIMO, D. M. Modernizing Political Science: A Model-Based Approach. **Perspectives on Politics**, v. 5, n. 4, pp. 741-753, 2007.

CLOUTIER, R.; MULLER, G.; VERMA, D.; et al. The Concept of Reference Architectures. **Systems Engineering**, v. 13, n. 1, p. 14–27, 2010.

COHEN, A. M.; HERSH, W. R. A survey of current work in biomedical text mining. **Briefings in Bioinformatics**, v. 6, n. 1, p. 57–71, 2005.

COHEN, A. M.; STAVRI, P. Z.; HERSH, W. R. A categorization and analysis of the criticisms of Evidence-Based Medicine. **International Journal of Medical Informatics**, v. 73, n. 1, p. 35–43, 2004.

COHEN, B. P. **Developing Sociological Knowledge: Theory and Method**. Prentice Hall, 1980.

COOPER, H. M.; HEDGES, L. V.; VALENTINE, J. C. **The Handbook of Research Synthesis and Meta-Analysis**. Russell Sage Foundation, 2009.

CRAVER, C. F.; DARDEN, L. Introduction. **Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences**, v. 36, n. 2, p. 233–244, 2005.

CRUZES, D. S.; DYBÅ, T. Recommended Steps for Thematic Synthesis in Software Engineering. **International Symposium on Empirical Software Engineering and Measurement**, p.275 –284, 2011a.

CRUZES, D. S.; DYBÅ, T. Research synthesis in software engineering: A tertiary study. **Information and Software Technology**, v. 53, n. 5, p. 440–455, 2011b.

CAUSEVIC, A.; SUNDMARK, D.; PUNNEKKAT, S. Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review. **IEEE Fourth International Conference on Software Testing, Verification and Validation (ICST)**. p.337–346, 2011.

DA CRUZ, S. M. S.; CAMPOS, M. L. M.; MATTOSO, M. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. **World Conference on Services - I**. p.259–266, 2009.

DAVIS, D.; DAVIS, M. E.; JADAD, A.; et al. The case for knowledge translation: shortening the journey from evidence to effect. **BMJ**, v. 327, n. 7405, p. 33–35, 2003.

DEELMAN, E.; GANNON, D.; SHIELDS, M.; TAYLOR, I. Workflows and e-Science: An overview of workflow system features and capabilities. **Future Generation Computer Systems**, v. 25, n. 5, p. 528–540, 2009.

DENŒUX, T. Reasoning with imprecise belief structures. **International Journal of Approximate Reasoning**, v. 20, n. 1, p. 79–111, 1999.

DENNIS, C. Biology databases: Information overload. **Nature**, v. 417, n. 6884, p. 14–14, 2002.

DIBBLE, D.; BOSTROM, R. P. Managing expert systems projects: factors critical for successful implementation. **Proceedings of the conference on The 1987 ACM SIGBDP-SIGCPR Conference**. p.96–128, 1987. New York, NY, USA: ACM.

DIESTE, O.; FERNANDEZ, E.; GARCIA-MARTINEZ, R.; JURISTO, N. The Risk of Using the Q Heterogeneity Estimator for Software Engineering Experiments. **International Symposium on Empirical Software Engineering and Measurement**. p.68–76, 2011.

DIESTE, O.; JURISTO, N. Systematic review and aggregation of empirical studies on elicitation techniques. **IEEE Transactions on Software Engineering**, v. 37, n. 2, p. 283 –304, 2011.

DIXON-WOODS, M.; AGARWAL, S.; JONES, D.; YOUNG, B.; SUTTON, A. Synthesising qualitative and quantitative evidence: a review of possible methods. **Journal of Health Services Research & Policy**, v. 10, n. 1, p. 45–53B, 2005.

DRUZDZEL, M. J.; VAN DER GAAG, L. C. Elicitation of probabilities for belief networks: combining qualitative and quantitative information. **Proceedings of the Eleventh conference on Uncertainty in artificial intelligence**. p.141–148, 1995. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

DUBIN, R. **Theory building**. New York, Free Press, 1969.

DUBIN, R. Theory building in applied areas. In: M.D. Dunnette (Ed.), **Handbook of industrial and organizational psychology**, p. 17-39, Chicago, Rand McNally, 1976.

DUNG, P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. **Artificial Intelligence**, v. 77, n. 2, p. 321–357, 1995.

DYBÅ, T.; DINGSØYR, T. Strength of evidence in systematic reviews in software engineering. **Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement**, p.178–187. New York, NY, USA: ACM. doi: 10.1145/1414004.1414034, 2008a.

DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. **Information and Software Technology**, v. 50, n. 9–10, p. 833-859. doi: 10.1016/j.infsof.2008.01.006, 2008b.

DYBÅ, T.; DINGSØYR, T.; HANSSEN, G. K. Applying Systematic Reviews to Diverse Study Types: An Experience Report. **First International Symposium on Empirical Software Engineering and Measurement**, p. 225–234, 2007.

DŽEROSKI, S.; LANGLEY, P.; TODOROVSKI, L. Computational Discovery of Scientific Knowledge. In: S. Džeroski; L. Todorovski (Orgs.); **Computational Discovery of Scientific Knowledge**, Lecture Notes in Computer Science. p.1–14, 2007. Springer Berlin Heidelberg.

EASTERBROOK, S.; SINGER, J.; STOREY, M.-A.; DAMIAN, D. Selecting Empirical Methods for Software Engineering Research. In: F. Shull; J. Singer; D. I. K. Sjøberg (Orgs.); **Guide to Advanced Empirical Software Engineering**. p.285-311. Springer London, 2008.

ERDOGMUS, H.; MORISIO, M.; TORCHIANO, M. On the effectiveness of the test-first approach to programming. **IEEE Transactions on Software Engineering**, v. 31, n. 3, p. 226–237, 2005.

ERIKSSON, H. A survey of knowledge acquisition techniques and tools and their relationship to software engineering. **Journal of Systems and Software**, v. 19, n. 1, p. 97–107, 1992.

EVANS, D.; FITZGERALD, M. Reasons for physically restraining patients and residents: a systematic review and content analysis. **International Journal of Nursing Studies**, v. 39, n. 7, p. 735–743. doi: 10.1016/S0020-7489(02)00015-9, 2002.

FALBO, R.A.; MENEZES, C.S.; ROCHA, A.R. Using Ontologies to Improve Knowledge Integration in Software Engineering Environments. **Proceedings of the World Multiconference on Systemic, Cybernetics and Informatics / 4th International Conference on Information Systems Analysis and Synthesis**, Orlando, USA, July, 1998.

FAYYAD, U.; STOLORZ, P. Data mining and KDD: Promise and challenges. **Future Generation Computer Systems**, v. 13, n. 2–3, p. 99–115, 1997.

FALKENHAINER, B. A Unified Approach to Explanation and Theory Formation. In: J. Shrager; P. Langley (Orgs.); **Computational Models of Scientific Discovery and Theory Formation**, Morgan Kaufman Series in Machine Learning. p.157–196, 1990. Morgan Kaufmann Pub.

FELDMAN, R.; CONEE, E. Evidentialism. **Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition**, v. 48, n. 1, p. 15–34, 1985.

FELLERS, J. Key factors in knowledge acquisition. **SIGCPR Comput. Pers.**, v. 11, n. 1, p. 10–24, 1987.

FERGUSON, E. S. **Engineering and the Mind’s Eye**. Cambridge, Mass.: The MIT Press, 1994.

FIORE, S.; ALOISIO, G. Special section: Data management for eScience. **Future Generation Computer Systems**, v. 27, n. 3, p. 290–291, 2011.

FORBUS, K. D. Qualitative process theory. **Artificial Intelligence**, v. 24, n. 1–3, p. 85–168, 1984.

FORBUS, K. D.; DEKLEER, J. **Building problem solvers**. Cambridge, Mass. [u.a.: MIT Press, 1993.

FORD, K. M. **Knowledge acquisition as modeling**. New York, NY: Wiley, 1993.

FOREMAN, J., GROSS, J., ROSENSTEIN, R., FISHER, D., & BRUNE, K. **C4 Software Technology Reference Guide: A Prototype** (Technical Report CMU/SEI-97-HB-001). Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1997.

FRANÇA, B. B. N.; TRAVASSOS, G. H. Experimentation with dynamic simulation models in software engineering: planning and reporting guidelines. **Empirical Software Engineering**, p. 1–44, 2015.

FREILING, M.; ALEXANDE, J.; MESSICK, S.; REHFUSS, S.; SHULMAN, S. Starting a Knowledge Engineering Project: A Step-By-Step Approach. **AI Magazine**, v. 6, n. 3, p. 150, 1985.

GALSTER, M.; AVGERIOU, P.; TOFAN, D. Constraints for the design of variability-intensive service-oriented reference architectures – An industrial case study. **Information and Software Technology**, Special Section: Component-Based Software Engineering (CBSE), v. 55, n. 2, p. 428–441, 2013.

GEORGE, B.; WILLIAMS, L. A structured experiment of test-driven development. **Information and Software Technology**, v. 46, n. 5, p. 337–342, 2004.

GERBER, A.; RAYMOND, K. MOF to EMF: There and Back Again. **Proceedings of the 2003 OOPSLA Workshop on Eclipse Technology eXchange**. New York, NY, USA: ACM, p.60–64, 2003.

GLASS, G. V. Primary, Secondary, and Meta-Analysis of Research. **Educational Researcher**, v. 5, n. 10, p. 3–8, 1976.

GLASS, R. L. Intuition’s Role in Decision Making. **IEEE Software**, v. 25, n. 1, p. 96, 95, 2008.

GLENNAN, S. Mechanisms and the nature of causation. **Erkenntnis**, v. 44, n. 1, p. 49–71. doi: 10.1007/BF00172853, 1996.

GLENNAN, S. Modeling mechanisms. **Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences**, v. 36, n. 2, p. 443–464. doi: 10.1016/j.shpsc.2005.03.011, 2005.

GIOIA, D. A.; PITRE, E. Multiparadigm Perspectives on Theory Building. **The Academy of Management Review**, v. 15, n. 4, p. 584. doi: 10.2307/258683, 1990.

GORDON, J.; SHORTLIFFE, E. H. A method for managing evidential reasoning in a hierarchical hypothesis space. **Artificial Intelligence**, v. 26, n. 3, p. 323–357. doi: 10.1016/0004-3702(85)90064-5, 1985.

GRAHAM, I. D.; LOGAN, J.; HARRISON, M. B.; et al. Lost in knowledge translation: Time for a map? **Journal of Continuing Education in the Health Professions**, v. 26, n. 1, p. 13–24, 2006.

GREGOR, S. The nature of theory in Information Systems. **MIS Quarterly: Management Information Systems**, v. 30, n. 3, p. 611–642, 2006.

GUERRA, C.; CAPITELLI, M.; LONGO, S. The Role of Paradigms in Science: A Historical Perspective. In: L. L'Abate (Org.); **Paradigms in Theory Construction**. p.19–30. Springer New York, 2012.

GUYATT, G.; RENNIE, D.; MEADE, M.; COOK, D. **Users' Guides to the Medical Literature: A Manual for Evidence-Based Clinical Practice, Second Edition**. 2º ed. McGraw-Hill Professional, 2008.

JACOBSON, I.; NG, P.-W.; MCMAHON, P. E.; SPENCE, I.; LIDMAN, S. **The Essence of Software Engineering: Applying the SEMAT Kernel**. 1 edition ed. Upper Saddle River, NJ: Addison-Wesley Professional, 2013.

JACOX, A. Theory construction in nursing: An overview. **Nursing Research**, v. 23, n. 1, p. 4-13, 1974.

JAMES, M. Applying meta-analytical procedures to software engineering experiments. **Journal of Systems and Software**, v. 54, n. 1, p. 29-39. doi: 10.1016/S0164-1212(00)00024-8, 2000.

JAMES, W. What pragmatism means. In **Pragmatism: A New Name for Some Old Ways of Thinking**, James W (ed.). Longmans, Green: New York; 43–81, 1907.

JOHNSON, P.; GOEDICKE, M.; EKSTEDT, M.; JACOBSON, I. Special Issue on General Theories of Software Engineering - Science of Computer Programming.

Available in: http://www.journals.elsevier.com/science-of-computer-programming/call-for-papers/special-issue-on-general-theories-of-software-engineering/. Accessed in: 8/23/2015.

KEARNEY, M. H. Enduring love: A grounded formal theory of women's experience of domestic violence. Research in Nursing & Health, v. 24, n. 4, p. 270–282, 2001.

KERLINGER, F. N. **Foundations of Behavioral Research**. New York, Hold, Rinehart and Winston Inc., 1988.

KHATRI, P.; SIROTA, M.; BUTTE, A. J. Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges. **PLoS Comput Biol**, v. 8, n. 2, p. e1002375, 2012.

KIRITCHENKO, S.; BRUIJN, B. DE; CARINI, S.; MARTIN, J.; SIM, I. ExaCT: automatic extraction of clinical trial characteristics from journal publications. **BMC Medical Informatics and Decision Making**, v. 10, n. 1, p. 56, 2010.

KITCHENHAM, B., **Procedures for performing systematic reviews**. In: TR/SE – 0401/NICTA Technical Report 0400011T, v. 1, p. 33, Keele University, 2004.

KITCHENHAM, B.; PEARL BRERETON, O.; BUDGEN, D.; et al. Systematic literature reviews in software engineering – A systematic literature review. **Information and Software Technology**, Special Section - Most Cited Articles in 2002 and Regular Research Papers., v. 51, n. 1, p. 7–15, 2009.

KITCHENHAM, B.; BUDGEN, D.; BRERETON, P. et al. Large-scale software engineering questions - expert opinion or empirical evidence? **IET Software**, v. 1, n. 5, p. 161-171. doi: 10.1049/iet-sen:20060052, 2007.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Keele University and Durham University Joint Report, 2007.

KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In: **26th International Conference on Software Engineering**, p. 273-281. doi: 10.1109/ICSE.2004.1317449, 2004.

KITCHENHAM, B.; PFLEEGER, S. L. Software quality: the elusive target [special issues section]. **IEEE Software**, v. 13, n. 1, p. 12 –21. doi: 10.1109/52.476281, 1996.

KITCHENHAM, B. A.; SJØBERG, D. I. K.; DYBÅ, T.; et al. Three empirical studies on the agreement of reviewers about the quality of software engineering experiments. **Information and Software Technology**, v. 54, n. 8, p. 804–819, 2012.

KITCHER, P. 1953 and all That. A Tale of Two Sciences. **The Philosophical Review**, v. 93, n. 3, p. 335–373. doi: 10.2307/2184541, 1984.

HAENNI, R. Shedding new light on Zadeh's criticism of Dempster's rule of combination. **8th International Conference on Information Fusion**, v. 2, p.879–884, 2005.

HANNAY, J. E.; DYBÅ, T.; ARISHOLM, E.; SJØBERG, D. I. K. The effectiveness of pair programming: A meta-analysis. **Information and Software Technology**, v. 51, n. 7, p. 1110–1122, 2009.

HANNAY, J. E.; SJOBERG, D. I. K.; DYBA, T. A Systematic Review of Theory Use in Software Engineering Experiments. **IEEE Transactions on Software Engineering**, v. 33, n. 2, p. 87 –107, 2007.

HARRISON, R.; BADOO, N.; BARRY, E. et al. Directions and Methodologies for Empirical Software Engineering Research. **Empirical Software Engineering**, v. 4, n. 4, p. 405-410. doi: 10.1023/A:1009877923978, 1999.

HARS, A. Designing Scientific Knowledge Infrastructures: The Contribution of Epistemology. **Information Systems Frontiers**, v. 3, n. 1, p. 63–73, 2001.

HEY, T.; TREFETHEN, A. The Data Deluge: An e-Science Perspective. In: F. Berman; G. Fox; T. Hey (Orgs.); **Grid Computing**. p.809–824, 2003. John Wiley & Sons, Ltd.

HIGGINS, P. A.; SHIRLEY, M. M. Levels of theoretical thinking in nursing. **Nursing Outlook**, v. 48, n. 4, p. 179–183. doi: 10.1067/mno.2000.105248, 2000.

HUNTER, A.; LIU, W. A survey of formalisms for representing and reasoning with scientific knowledge. **The Knowledge Engineering Review**, v. 25, n. 02, p. 199–222, 2010.

HUNTER, J. Scientific Publication Packages – A Selective Approach to the Communication and Archival of Scientific Output. **International Journal of Digital Curation**, v. 1, n. 1, p. 33–52, 2008.

ICHIKAWA, J. J.; STEUP, M. The Analysis of Knowledge. In: E. N. Zalta (Org.); **The Stanford Encyclopedia of Philosophy**. Spring 2014 ed., 2014. Available in: <http://plato.stanford.edu/archives/spr2014/entries/knowledge-analysis/>. Accessed in 1/4/2016.

IMTIAZ, S.; BANO, M.; IKRAM, N.; NIAZI, M. A Tertiary Study: Experiences of Conducting Systematic Literature Reviews in Software Engineering. **Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering**. p.177–182, 2013. New York, NY, USA: ACM.

IVARSSON, M.; GORSCHEK, T. Tool support for disseminating and improving development practices. **Software Quality Journal**, v. 20, n. 1, p. 173–199, 2012.

L'ABATE, L. **Paradigms in Theory Construction**. Springer New York, 2012.

LAGOZE, C.; PATZKE, K. A research agenda for data curation cyberinfrastructure. **Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries**. p.373–382, 2011. New York, NY, USA: ACM.

LANGLEY, P. **Scientific discovery: computational explorations of the creative processes**. Cambridge, Mass.: MIT Press, 1987.

LANGLEY, P.; ZYTKOW, J. M.; BRADSHAW, G. L.; SIMON, H. A. Three facets of scientific discovery. **Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 1**. p.465–468, 1983. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

LARSSON, R. Case Survey Methodology: Quantitative Analysis of Patterns Across Case Studies. **Academy of Management Journal**, v. 36, n. 6, p. 1515–1546, 1993.

LENAT, D. B.; FEIGENBAUM, E. A. On the thresholds of knowledge. **Artificial Intelligence**, v. 47, n. 1–3, p. 185–250, 1991.

LENZ, E.; SUPPE, F.; GIFT, A.; Pugh, L.; MILLIGAN, R. Collaborative development of middle-range theories: toward a theory of unpleasant symptoms. **Advances in Nursing Science**, v. 17, n. 3, p. 1-13, 1995.

LIN, C.; LU, S.; FEI, X.; et al. A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution. **IEEE Transactions on Services Computing**, v. 2, n. 1, p. 79–92, 2009.

LORD, P.; MACDONALD, A.; LYON, L.; GIARETTA, D. From Data Deluge to Data Curation. **In Proc 3th UK e-Science All Hands Meeting**. p.371–375, 2004.

LOWRANCE, J. D.; GARVEY, T. D.; STRAT, T. M. A Framework for Evidential-Reasoning Systems. In: R. R. Yager; L. Liu (Orgs.); **Classic Works of the Dempster-Shafer Theory of Belief Functions**, Studies in Fuzziness and Soft Computing. p.419–434, 2008. Springer Berlin Heidelberg.

LYNHAM, S. A. The General Method of Theory-Building Research in Applied Disciplines. **Advances in Developing Human Resources**, v. 4, n. 3, p. 221-241. doi: 10.1177/1523422302043002, 2002.

LOPES, V. P.; TRAVASSOS, G. H. Knowledge Repository Structure of an Experimental Software Engineering Environment. **XXIII Brazilian Symposium on Software Engineering**. p.32–42, 2009.

MACCAGNAN, A.; RIVA, M.; FELTRIN, E.; et al. Combining ontologies and workflows to design formal protocols for biological laboratories. **Automated Experimentation**, v. 2, n. 1, p. 1–14, 2010.

MAGNUS, P.D. **What species can teach us about theory**. Unpublished manuscript. Available at: http://www.fecundity.com/job/paper.php?item=speciesanalogy. Accessed in November 19, 2014.

MAHONEY, J.; GOERTZ, G. A Tale of Two Cultures: Contrasting Quantitative and Qualitative Research. **Political Analysis**, v. 14, n. 3, p. 227–249, 2006.

MARCHENKO, A.; ABRAHAMSSON, P.; IHME, T. Long-Term Effects of Test-Driven Development A Case Study. In: P. Abrahamsson; M. Marchesi; F. Maurer (Orgs.); **Agile Processes in Software Engineering and Extreme Programming**, Lecture Notes in Business Information Processing. p.13–22, 2009. Springer Berlin Heidelberg.

MARKMAN, A. B. **Knowledge representation**. Mahwah, NJ: L. Erlbaum, 1999.

MARSHALL, C.; BRERETON, P. Tools to Support Systematic Literature Reviews in Software Engineering: A Mapping Study. **ACM / IEEE International Symposium on Empirical Software Engineering and Measurement**. p.296–299, 2013.

MARTÍNEZ-FERNÁNDEZ, S.; AYALA, C. P.; FRANCH, X.; MARQUES, H. M. Benefits and Drawbacks of Reference Architectures. In: K. Drira (Org.); **Software Architecture**, Lecture Notes in Computer Science. Springer Berlin Heidelberg, p.307–310, 2013.

MARTÍNEZ-FERNÁNDEZ, S.; AYALA, C. P.; FRANCH, X.; NAKAGAWA, E. Y. A Survey on the Benefits and Drawbacks of AUTOSAR. **Proceedings of the First International Workshop on Automotive Software Architecture**. New York, NY, USA: ACM, p.19–26, 2015a.

MARTÍNEZ-FERNÁNDEZ, S.; SANTOS, P.S.M.; AYALA, C. P.; FRANCH, X.; TRAVASSOS, G. H. Aggregating Empirical Evidence about the Benefits and Drawbacks of Software Reference Architectures. **ACM / IEEE International**

Symposium on Empirical Software Engineering and Measurement. 2015b, in press.

MCDERMOTT, J. Preliminary Steps Toward a Taxonomy of Problem-Solving Methods. In: S. Marcus (Org.); **Automating Knowledge Acquisition for Expert Systems**, The Kluwer International Series in Engineering and Computer Science. p.225–256, 1988. Springer US.

MCKELVEY, B. Complexity Theory in Organization Science: Seizing the Promise or Becoming a Fad? **Emergence**, v. 1, n. 1, p. 5–32. doi: 10.1207/s15327000em0101_2, 1999.

MELLO, R. M.; DA SILVA, P. C.; RUNESON, P.; TRAVASSOS, G. H. Towards a Framework to Support Large Scale Sampling in Software Engineering Surveys. **Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement**. p.48:1–48:4, New York, NY, USA: ACM, 2014.

MOHAMED, M.; ROMDHANI, M.; GHEDIRA, K. MOF-EMF Alignment. **International Conference on Autonomic and Autonomous Systems**. Los Alamitos, CA, USA: IEEE Computer Society**,** v. 0, p.1, 2007.

MOODY, D. L. The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. **IEEE Transactions on Software Engineering**, v. 35, n. 6, p. 756–779, 2009.

MONS, B. Which gene did you mean? **BMC Bioinformatics**, v. 6, n. 1, p. 142, 2005.

MONS, B.; VELTEROP, J. Nano-Publication in the e-science era. **Workshop on Semantic Web Applications in Scientific Discourse**, 2009.

MOTTA, E.; RAJAN, T.; EISENSTADT, M. Knowledge acquisition as a process of model refinement. **Knowledge Acquisition**, v. 2, n. 1, p. 21–49, 1990.

NAKAGAWA, E. Y.; ANTONINO, P. O.; BECKER, M. Reference Architecture and Product Line Architecture: A Subtle But Critical Difference. In: I. Crnkovic; V. Gruhn; M. Book (Orgs.); **Software Architecture**, Lecture Notes in Computer Science. p.207–211, 2011. Springer Berlin Heidelberg.

NEWMAN, H. B.; ELLISMAN, M. H.; ORCUTT, J. A. Data-intensive e-Science Frontier Research. **Commun. ACM**, v. 46, n. 11, p. 68–77, 2003.

NOBLIT, G. W.; HARE, R. D. **Meta-Ethnography: Synthesizing Qualitative Studies**. SAGE, 1988.

NOVÈRE, N. L.; HUCKA, M.; MI, H.; et al. The Systems Biology Graphical Notation. **Nature Biotechnology**, v. 27, n. 8, p. 735–741, 2009.

O'GRADY, L. What is knowledge and when should it be implemented? **Journal of Evaluation in Clinical Practice**, v. 18, n. 5, p. 951–953, 2012.

OMG. **Essence – Kernel and Language for Software Engineering Methods**. Object Management Group (OMG), OMG Document ad/2013-02-01. Available at: http://www.omg.org/cgi-bin/doc?ad/13-02-01. Accessed in November 22, 2014.

OVERTON, W. F. The Structure of Developmental Theory. In: P. Geert; L. P. Mos (Orgs.); **Annals of Theoretical Psychology**, v. 7, p.191-235 Springer US, 1991.

PAI, M.; MCCULLOCH, M.; GORMAN, JD. et al. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. **The National medical journal of India**, v. 17, n. 2, p. 86, 2004.

PARSONS, S. Some qualitative approaches to applying the Dempster-Shafer theory. **Information and decision technologies**, v. 19, n. 4, p. 321–337, 1994.

PAWAR, B. S. **Theory Building for Hypothesis Specification in Organizational Studies**. Sage Publications Pvt. Ltd, 2009.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. **Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering**, p.68–77, 2008. Swinton, UK, UK: British Computer Society.

PLANT, R. T. Rigorous approach to the development of knowledge-based systems. **Knowledge-Based Systems**, v. 4, n. 4, p. 186–196, 1991.

PREECE, A. D. Towards a methodology for evaluating expert systems. **Expert Systems**, v. 7, n. 4, p. 215–223, 1990.

RAFIQUE, Y.; MISIC, V. B. The Effects of Test-Driven Development on External Quality and Productivity: A Meta-Analysis. **IEEE Transactions on Software Engineering**, v. 39, n. 6, p. 835–856, 2013.

RAINER, A.; JAGIELSKA, D.; HALL, T. Software engineering practice versus evidence-based software engineering research. **Proceedings of the 2005 workshop on Realising evidence-based software engineering**, p.1–5, New York, NY, USA: ACM. doi: 10.1145/1082983.1083177, 2005.

RAKOWSKY, U. K. Fundamentals of the dempster-shafer theory and its applications to reliability modeling. **International Journal of Reliability, Quality and Safety Engineering**, v. 14, n. 06, p. 579–601, 2007.

RALPH, P.; JOHNSON, P.; JORDAN, H. Report on the First SEMAT Workshop on General Theory of Software Engineering (GTSE 2012). **SIGSOFT Softw. Eng. Notes**, v. 38, n. 2, p. 26–28, 2013.

REYNOLDS, P. D. **A Primer in Theory Construction**. Bobbs-Merrill Co, 1971.

RIHOUX, B. Qualitative Comparative Analysis (QCA) and Related Systematic Comparative Methods Recent Advances and Remaining Challenges for Social Science Research. **International Sociology**, v. 21, n. 5, p. 679–706, 2006.

ROOK, F. W.; CROGHAN, J. W. The knowledge acquisition activity matrix: a systems engineering conceptual framework. **IEEE Transactions on Systems, Man and Cybernetics**, v. 19, n. 3, p. 586–597, 1989.

ROSEN, R. On the limitations of scientific knowledge. In **Boundaries and Barriers: On the Limits of Scientific Knowledge**. Casti J. L. and Karlqvist A. (Eds.). Addison Wesley, Reading, MA, 199-214, 1996.

ROSENTHAL, A. The History of Calculus. **The American Mathematical Monthly**, v. 58, n. 2, p. 75–86, 1951.

RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. **Empirical Software Engineering**, v. 14, n. 2, p. 131-164. doi: 10.1007/s10664-008-9102-8, 2009.

RUSPINI, E. H.; LOWRANCE, J. D.; STRAT, T. M. Understanding evidential reasoning. **International Journal of Approximate Reasoning**, v. 6, n. 3, p. 401–424, 1992.

RYCROFT-MALONE, J.; SEERS, K.; TITCHEN, A.; et al. What counts as evidence in evidence-based practice? **Journal of Advanced Nursing**, v. 47, n. 1, p. 81–90, 2004.

RZHETSKY, A.; IOSSIFOV, I.; KOIKE, T.; et al. GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. **Journal of biomedical informatics**, v. 37, n. 1, p. 43–53, 2004.

SACKETT, D. L.; ROSENBERG, W. M. C.; GRAY, J. A. M.; HAYNES, R. B.; RICHARDSON, W. S. Evidence based medicine: what it is and what it isn’t. **BMJ**, v. 312, n. 7023, p. 71–72, 1996.

SANDERS, T. J. M.; SPOOREN, W. P. M.; NOORDMAN, L. G. M. Coherence relations in a cognitive theory of discourse representation. **Cognitive Linguistics**, v. 4, n. 2, p. 93–134, 1993.

SALO, O.; ABRAHAMSSON, P. An iterative improvement process for agile software development. **Software Process: Improvement and Practice**, v. 12, n. 1, p. 81-100. doi: 10.1002/spip.305, 2007.

SANTOS, P. S. M.; NASCIMENTO, I. E.; TRAVASSOS, G. H. A Computational Infrastructure for Research Synthesis in Software Engineering. **XVIII Ibero-American Conference on Software Engineering, Track: XVII Workshop on Experimental Software Engineering**, p. 309-322, Lima, Peru, 2015.

SANTOS, P. S. M.; TRAVASSOS, G. H. Action Research Can Swing the Balance in Experimental Software Engineering. **Advances in Computers**. v. 83, p. 205-276. Elsevier, 2011.

SANTOS, P. S. M.; TRAVASSOS, G. H. On the Representation and Aggregation of Evidence in Software Engineering: A Theory and Belief-based Perspective. **Electron. Notes Theor. Comput. Sci.**, v. 292, p. 95–118, 2013.

SANTOS, P. S. M.; TRAVASSOS, G. H. Scientific Knowledge Engineering: a Conceptual Delineation and Overview of the State of the Art. **Knowledge Engineering Review**, *accepted for publication*.

SCANNIELLO, G.; GRAVINO, C.; RISI, M.; TORTORA, G. A Controlled Experiment for Assessing the Contribution of Design Pattern Documentation on Software Maintenance. **Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement**. p.52:1–52:4. New York, NY, USA: ACM, 2010.

SCHREIBER, G. **Knowledge engineering and management: the CommonKADS methodology**. Cambridge, Mass.: MIT Press, 2000.

SCOTT-FINDLAY, S.; POLLOCK, C. Evidence, Research, Knowledge: A Call for Conceptual Clarity. **Worldviews on Evidence-Based Nursing**, v. 1, n. 2, p. 92–97, 2004.

SFETSOS, P.; STAMELOS, I.; ANGELIS, L.; DELIGIANNIS, I. An experimental investigation of personality types impact on pair effectiveness in pair programming. **Empirical Software Engineering**, v. 14, n. 2, p. 187–226, 2008.

SHAFER, G. **A Mathematical Theory of Evidence**. Princeton University Press, 1976.

SHORTLIFFE, E. H.; BUCHANAN, B. G.; FEIGENBAUM, E. A. Knowledge engineering for medical decision making: A review of computer-based clinical decision aids. **Proceedings of the IEEE**, v. 67, n. 9, p. 1207–1224, 1979.

SHOTTON, D. Semantic publishing: the coming revolution in scientific journal publishing. **Learned Publishing**, v. 22, n. 2, p. 85–94, 2009.

SHULL, F.; FELDMANN, R.; SHAW, M. Building decision support in an imperfect world. **Proceedings of International Symposium on Empirical Software Engineering (ISESE)**, Rio de Janeiro, Brazil, pp. 33–35, 2006.

SHULL, F.; SINGER, J.; SJØBERG, D. I. K. **Guide to Advanced Empirical Software Engineering**. 2008 edition ed. London: Springer, 2007.

SHRAGER, J. **Computational Models of Scientific Discovery and Theory Formation**. Morgan Kaufmann Pub, 1990.

SIEBRA, C. S. A.; TONIN, G. S.; SILVA, F. Q. B.; et al. Managing Technical Debt in Practice: An Industrial Report. **Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement**. p.247–250, New York, NY, USA: ACM, 2012.

SILVA, F. Q. B.; CRUZ, S. S. J. O.; GOUVEIA, T. B.; CAPRETZ, L. F. Using Meta-ethnography to Synthesize Research: A Worked Example of the Relations between Personality and Software Team Processes. **ACM / IEEE International Symposium on Empirical Software Engineering and Measurement**, p.153–162, 2013.

SILVA, F. Q. B.; SANTOS, A. L. M.; SOARES, S.; et al. Six years of systematic literature reviews in software engineering: An updated tertiary study. **Information and Software Technology**, v. 53, n. 9, p. 899–913, 2011.

SIMON, H. A. Scientific Discovery and the Psychology of Problem Solving. **Models of Discovery**, Boston Studies in the Philosophy of Science. p.286–303, 1977. Springer Netherlands.

SJØBERG, D. I. K.; DYBÅ, T.; ANDA, B. C. D.; HANNAY, J. E. Building Theories in Software Engineering. In: F. Shull; J. Singer; D. I. K. Sjøberg (Orgs.); **Guide to Advanced Empirical Software Engineering**. p.312–336. London: Springer London, 2008.

SLATER, T.; BOUTON, C.; HUANG, E. S. Beyond data integration. **Drug Discovery Today**, v. 13, n. 13–14, p. 584–589, 2008.

SLYNGSTAD, O. P. N.; LI, J.; CONRADI, R.; et al. The Impact of Test Driven Development on the Evolution of a Reusable Framework of Components: An Industrial Case Study. **The Third International Conference on Software Engineering Advances**. ICSEA '08. p.214–223, 2008.

SMETS, P.; KENNES, R. The transferable belief model. **Artificial Intelligence**, v. 66, n. 2, p. 191–234, 1994.

SRIVASTAVA, R. P.; MOCK, T. J. Introduction to Belief Functions. In: P. R. P. Srivastava; P. T. J. Mock (Orgs.); **Belief Functions in Business Decisions**, Studies in Fuzziness and Soft Computing. p.1–16, 2002. Physica-Verlag HD.

STOL, K.-J.; FITZGERALD, B. Uncovering theories in software engineering. **2nd SEMAT Workshop on a General Theory of Software Engineering (GTSE)**. p.5–14, 2013.

STOL, K.-J.; FITZGERALD, B. Theory-oriented software engineering. **Science of Computer Programming**, Towards general theories of software engineering., v. 101, p. 79–98, 2015.

STRAUS, S.; TETROE, J.; GRAHAM, I. D. (ORGS.). **Knowledge Translation in Health Care: Moving from Evidence to Practice**. 2 ed. Chichester, West Sussex ; Hoboken, NJ: BMJ Books, 2013.

STRIGINI, L. Limiting the Dangers of Intuitive Decision Making. **IEEE Software**, v. 13, n. 1, p. 101–103, 1996.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: Principles and methods. **Data & Knowledge Engineering**, v. 25, n. 1–2, p. 161–197, 1998.

SUPPE, F. (ORG.). **The Structure of Scientific Theories**. 2nd ed. University of Illinois Press, 1977.

SUPPE, F. Understanding Scientific Theories: An Assessment of Developments, 1969-1998. **Philosophy of Science**, v. 67, pp. S102-S115, 2000.

SUPPES, P. **Representation and Invariance of Scientific Structures**. 1º ed. Center for the Study of Language and Inf, 2001.

SUTTON, A. J.; ABRAMS, K. R. Bayesian methods in meta-analysis and evidence synthesis. **Statistical Methods in Medical Research**, v. 10, n. 4, p. 277–303, 2001.

TARLIER, D. Mediating the meaning of evidence through epistemological diversity. **Nursing Inquiry**, v. 12, n. 2, p. 126–134, 2005.

THAGARD. P.; NOWAK. G. The conceptual structure of the geological revolution. In J. Shrager & P. Langley (Eds.), **Computational models of discovery and theory formation**. San Mateo, CA: Morgan Kaufman, 1990.

THELIN, T.; ANDERSSON, C.; RUNESON, P.; DZAMASHVILI-FOGELSTROM, N. A replicated experiment of usage-based and checklist-based reading. **Proceedings of the 10th International Symposium on Software Metrics**, p.246– 256. IEEE. doi: 10.1109/METRIC.2004.1357907, 2004.

THELIN, T.; RUNESON, P.; REGNELL, B. Usage-based reading—an experiment to guide reviewers with use cases. **Information and Software Technology**, v. 43, n. 15, p. 925–938. doi: 10.1016/S0950-5849(01)00201-4, 2001.

THELIN, T.; RUNESON, P.; WOHLIN, C. An experimental comparison of usage-based and checklist-based reading. **IEEE Transactions on Software Engineering**, v. 29, n. 8, p. 687– 704. doi: 10.1109/TSE.2003.1223644, 2003.

THOMAS, J.; HARDEN, A. Methods for the thematic synthesis of qualitative research in systematic reviews. **BMC Medical Research Methodology**, v. 8, n. 1, p. 45, 2008.

THOMSON-JONES, M. Models and the Semantic View. **Philosophy of Science**, v. 73, n. 5, p. 524–535. doi: 10.1086/518322, 2006.

TRAVASSOS, G. H.; DOS SANTOS, P. S. M.; DIAS NETO, A. C.; Mian, P. G. M.; BIOLCHINI, J. An Environment to Support Large Scale Experimentation in Software Engineering. **Proceeding of the 13th IEEE International Conference on Engineering of Complex Computer Systems**. p.193–202, 2008.

UPSHUR, R. E. G. Seven characteristics of medical evidence. **Journal of Evaluation in Clinical Practice**, v. 6, n. 2, p. 93–97, 2000.

UPSHUR, R. E. G.; VANDENKERKHOF, E. G.; GOEL, V. Meaning and measurement: an inclusive model of evidence in health care. **Journal of Evaluation in Clinical Practice**, v. 7, n. 2, p. 91–96, 2001.

VALDÉS-PÉREZ, R. E. Computer science research on scientific discovery. **The Knowledge Engineering Review**, v. 11, n. 01, p. 57–66, 1996.

VORMS, M. Representing with imaginary models: Formats matter. **Studies in History and Philosophy of Science Part A**, v. 42, n. 2, p. 287–295, 2011.

YAMASAKI, S. AND RIHOUX, B. A commented review of applications. In RIHOUX, B. AND RAGIN, C. C. (Eds.), **Configurational Comparative Methods**. Los Angeles: Sage, pp. 123–145, 2009.

YANG, B. Meta-Analysis Research and Theory Building. **Advances in Developing Human Resources**, v. 4, n. 3, p. 296–316. doi: 10.1177/1523422302043005, 2002.

YIN, R. K.; HEALD, K. A. Using the Case Survey Method to Analyze Policy Studies. **Administrative Science Quarterly**, v. 20, n. 3, p. 371. doi: 10.2307/2391997, 1975.

WALKER, L. O.; AVANT, K. C. **Strategies for Theory Construction in Nursing**. 4o ed. Prentice Hall, 2004.

WALLACE, D. R.; FUJII, R. U. Software verification and validation: an overview. **IEEE Software**, v. 6, n. 3, p. 10–17, 1989.

WANG, S. C. 8th FY Khoo Memorial Lecture 2012--Why radiologists need philosophy. **Annals of the Academy of Medicine, Singapore**, v. 41, n. 7, p. 315–322, 2012.

WATERS, D.; RYCHETNIK, L.; CRISP, J.; BARRATT, A. Views on evidence from nursing and midwifery opinion leaders. **Nurse Education Today**, v. 29, n. 8, p. 829–834, 2009.

WIELINGA, B. J.; SCHREIBER, A. T.; BREUKER, J. A. KADS: a modelling approach to knowledge engineering. **Knowledge Acquisition**, v. 4, n. 1, p. 5–53, 1992.

WIERINGA, R.; DANEVA, M.; CONDORI-FERNANDEZ, N. The Structure of Design Theories, and an Analysis of their Use in Software Engineering Experiments. **International Symposium on Empirical Software Engineering and Measurement**. p.295 -304. doi: 10.1109/ESEM.2011.38, 2011.

WIERINGA, R. J. **Design Science Methodology for Information Systems and Software Engineering**. 2014 edition New York, NY: Springer, 2014.

WINKLER, D.; HALLING, M.; BIFFL, S. Investigating the effect of expert ranking of use cases for design inspection. **Proceedings of 30th Euromicro Conference**, p.362 – 371. doi: 10.1109/EURMIC.2004.1333391, 2004.

WHETTEN, D. A. What Constitutes a Theoretical Contribution? **Academy of Management Review**, v. 14, n. 4, p. 490–495. doi: 10.5465/AMR.1989.4308371, 1989.

WOHLIN, C.; HÖST, M.; HENNINGSSON, K. Empirical Research Methods in Software Engineering. In: R. Conradi; A. Wang (Orgs.); **Empirical Methods and Studies in**

Software Engineering, Lecture Notes in Computer Science. v. 2765, p.7-23. Springer Berlin / Heidelberg, 2003.

WORREN, N. A.; MOORE, K.; ELLIOTT, R. When theories become tools: Toward a framework for pragmatic validity. **Human Relations**, v. 55, n. 10, p. 1227 –1250, 2002.

ZELKOWITZ, M. Techniques for Empirical Validation. In: V. Basili; D. Rombach; K. Schneider; et al. (Orgs.); **Empirical Software Engineering Issues. Critical Assessment and Future Directions**, Lecture Notes in Computer Science. v. 4336, p.4-9. Springer Berlin / Heidelberg, 2007.

ZHANG, H.; ALI BABAR, M. Systematic reviews in software engineering: An empirical investigation. **Information and Software Technology**, v. 55, n. 7, p. 1341–1354, 2013.

REFERENCES FOR CHAPTER 4 LITERATURE REVIEW

BÖLLING, C.; WEIDLICH, M.; HOLZHÜTTER, H.-G. SEE: structured representation of scientific evidence in the biomedical domain using Semantic Web techniques. **Journal of Biomedical Semantics**, v. 5, n. Suppl 1, p. S1, 2014.

BOYCE, R. D.; COLLINS, C.; HORN, J.; KALET, I. Modeling Drug Mechanism Knowledge Using Evidence and Truth Maintenance. **IEEE Transactions on Information Technology in Biomedicine**, v. 11, n. 4, p. 386–397, 2007.

BRODARIC, B.; REITSMA, F.; QIANG, Y. SKIing with DOLCE: Toward an e-Science Knowledge Infrastructure. **Proceedings of the Fifth International Conference on Formal Ontology in Information System**, p.208–219, 2008. Amsterdam, The Netherlands, The Netherlands: IOS Press.

CICCARESE, P.; WU, E.; WONG, G.; et al. The SWAN biomedical discourse ontology. **Journal of biomedical informatics**, v. 41, n. 5, p. 739–751, 2008.

CLARE, A.; CROSET, S.; GRABMUELLER, C.; et al. Exploring the generation and integration of publishable scientic facts using the concept of nano-publications. **Workshop on Semantic Publishing at ESWC2011**, p.13-17, 2011.

CROFT, D.; O'KELLY, G.; WU, G.; et al. Reactome: a database of reactions, pathways and biological processes. **Nucleic Acids Research**, v. 39, n. suppl 1, p. D691–D697, 2011.

DE WAARD, A.; BUCKINGHAM SHUM, S.; CARUSI, A.; et al. Hypotheses, evidence and relationships: The HypER approach for representing scientific knowledge claims, 2009, Washington DC.

DE WAARD, A.; SCHNEIDER, J. Formalising Uncertainty: An Ontology of Reasoning, Certainty and Attribution (ORCA). **Workshop on Semantic Technologies Applied to Biomedical Informatics and Individualized Medicine (SATBI+SWIM)**, 2012.

DINAKARPANDIAN, D.; LEE, Y.; VISHWANATH, K.; LINGAMBHOTLA, R. MachineProse: An Ontological Framework for Scientific Assertions. **Journal of the American Medical Informatics Association**, v. 13, n. 2, p. 220–232, 2006.

EKAPUTRA, F.; SABOU, M.; SERRAL, E.; BIFFL, S. Supporting Information Sharing for Reuse and Analysis of Scientific Research Publication Data. **Proceedings of the 4th Workshop on Semantic Publishing**, 2014. Anissaras, Greece.

GROTH, P.; GIBSON, A.; VELTEROP, J. The anatomy of a nanopublication. **Information Services and Use**, v. 30, n. 1, p. 51–56, 2010.

GROZA, T.; MÖLLER, K.; HANDSCHUH, S.; TRIF, D.; DECKER, S. SALT: Weaving the Claim Web. In: K. Aberer; K.-S. Choi; N. Noy; et al. (Orgs.); **The Semantic Web**, Lecture Notes in Computer Science. p.197–210, 2007. Springer Berlin Heidelberg.

HUNTER, A.; WILLIAMS, M. Aggregating evidence about the positive and negative effects of treatments. **Artificial Intelligence in Medicine**, v. 56, n. 3, p. 173–190, 2012.

KRAINES, S.; GUO, W. A System for Ontology-Based Sharing of Expert Knowledge in Sustainability Science. **Data Science Journal**, v. 9, p. 107–123, 2011.

KUHN, T.; BARBANO, P. E.; NAGY, M. L.; KRAUTHAMMER, M. Broadening the Scope of Nanopublications. In: P. Cimiano; O. Corcho; V. Presutti; L. Hollink; S. Rudolph (Orgs.); **The Semantic Web: Semantics and Big Data**, Lecture Notes in Computer Science. p.487–501, 2013. Springer Berlin Heidelberg.

MANCINI, C.; BUCKINGHAM SHUM, S. J. Modelling discourse in contested domains: A semiotic and cognitive framework. **International Journal of Human-Computer Studies**, v. 64, n. 11, p. 1154–1171, 2006.

MARCONDES, C. H. Knowledge network of scientific claims derived from a semantic publication system. **Information Services and Use**, v. 31, n. 3, p. 167–176, 2011.

PIKE, W.; GAHEGAN, M. Beyond ontologies: Toward situated representations of scientific knowledge. **International Journal of Human-Computer Studies**, v. 65, n. 7, p. 674–688, 2007.

RUSS, T. A.; RAMAKRISHNAN, C.; HOVY, E. H.; BOTA, M.; BURNS, G. A. Knowledge engineering tools for reasoning with scientific observations and interpretations: a neural connectivity use case. **BMC Bioinformatics**, v. 12, n. 1, p. 351, 2011.

SANTOS, P. S. M.; TRAVASSOS, G. H. On the Representation and Aggregation of Evidence in Software Engineering: A Theory and Belief-based Perspective. **Electron. Notes Theor. Comput. Sci.**, v. 292, p. 95–118, 2013.

SHARMA, R.; POOLE, D.; SMYTH, C. A framework for ontologically-grounded probabilistic matching. **International Journal of Approximate Reasoning**, Bayesian Model Views., v. 51, n. 2, p. 240–262, 2010.

VAN VALKENHOEF, G.; TERVONEN, T.; ZWINKELS, T.; DE BROCK, B.; HILLEGE, H. ADDIS: A decision support system for evidence-based medicine. **Decision Support Systems**, v. 55, n. 2, p. 459–475, 2013.

Appendix A. Primary study quality assessment checklists and scoring schema

A.1. Quality assessment of experimental studies

As the checklist can sum up to 0.25, where each question can be rated up to 0.025. Evaluation rules are defined based on the number of 'things to consider' that are taken into account in each question:

- 0.0000 = <u>None</u> of the 'things to consider' for that question is answered with 'yes';
- 0.0083 = <u>Some</u> of the 'things to consider' for that question is answered with 'yes';
- 0.0166 = <u>Most</u> of the 'things to consider' for that question is answered with 'yes';
- 0.0250 = <u>All</u> of the 'things to consider' for that question is answered with 'yes'.

Two changes were applied to the checklist. One item ('was there random allocation to treatments?') was removed since it is already considered in the evidence hierarchy adopted to estimate belief values. Furthermore, one question was added (question 3) from the checklist proposed in Dybå and Dingsøyr (2008b), which relates to the experiment context.

#	Question	Things to consider	Ok?	Total
Category: Questions on Aims				
1.	Do the authors clearly state the aims of the research?	*Do the authors state research questions, e.g., related to time-to-market, cost, product quality, process quality, developer productivity, and developer skills?*		
		Do the authors state hypotheses and their underlying theories?		
Category: Questions on Design, Data Collection, and Data Analysis				
2.	Is there an adequate description of the context in which the research was carried out?	*The industry in which products are used (e.g. banking, telecommunications, consumer goods, travel, etc)*		
		If applicable, the nature of the software development organization (e.g. in-house department or independent software supplier)		
		The skills and experience of the subjects (e.g. with a language, a method, a tool, an application domain)		
		The type of software products used (e.g. a design tool, a compiler)		

		If applicable, the software processes being used (e.g. a company standard process, the quality assurance procedures, the configuration management process)		
3.	Do the authors describe the sample and experimental units (=experimental materials and participants as individuals or teams)?	*Do the authors explain how experimental units were defined and selected?*		
		Do the authors state to what degree the experimental units are representative?		
		Do the authors explain why the experimental units they selected were the most appropriate for providing insight into the type of knowledge sought by the experiment?		
		Do the authors report the sample size?		
4.	Do the authors describe the design of the experiment?	*Do the authors clearly describe the chosen design (blocking, within or between subject design, do treatments have levels)?*		
		Do the authors define/describe all treatments and all controls?		
5.	Do the authors describe the data collection procedures and define the measures?	*Are all measures clearly defined (e.g., scale, unit, counting rules)?*		
		Is the form of the data clear (e.g., tape recording, video material, notes, etc.)?		
		Are quality control methods used to ensure consistency, completeness and accuracy of collected data?		
		Do the authors report drop-outs?		
6.	Do the authors define the data analysis procedures?	*Do authors justify their choice / describe the procedures / provide references to descriptions of the procedures?*		
		Do the authors report significance levels and effect sizes?		
		If outliers are mentioned and excluded from the analysis, is this justified?		
		Do the authors report or give references to raw data and/or descriptive statistics?		
7.	Do the authors discuss potential experimenter bias?	*Were the authors the developers of some or all of the treatments? If yes, do the authors discuss the implications anywhere in the paper? (If the authors developed the treatments (or parts of them) without discussing the implications, the answer to question 7 is "not at all".)*		
		Was training and conduct equivalent for all treatment groups?		
		Was there allocation concealment, i.e., did the researchers know to what treatment each subject was assigned?		

8.	Do the authors discuss the limitations of their study?	*Do the authors discuss external validity with respect to subjects, materials, and tasks?*		
		If the study was a quasi-experiment, do the authors discuss the design components that were used to address any study weaknesses?		
		If the study used novel measures, is the construct validity of the measures discussed?		
Category: Questions on Study Outcome				
9.	Do the authors state the findings clearly?	*Do the authors present results clearly?*		
		Do the authors present conclusions clearly?		
		Are the conclusions warranted by the results and are the connections between the results and conclusions presented clearly?		
		Do the authors discuss their conclusions in relation to the original research questions?		
		Are limitations of the study discussed explicitly?		
10.	Is there evidence that the E/QE can be used by other researchers / practitioners?	*Do the authors discuss whether or how the findings can be transferred to other populations, or consider other ways in which the research can be used?*		
		To what extent do authors interpret results in the context of other studies / the existing body of knowledge?		

A.2. Quality assessment of non-systematic studies

Pts	Possible Answers
Question 1: How the practice was applied (weight = 35%)	
0,0	If it is unclear how the practice was applied.
0,0375	For one experiment (i.e., a trial of the practice out of context).
0,0500	For a series of experiments, or for one pilot project (i.e., a trial of the practice in context).
0,0625	For a series of pilot projects, or for one application in a production project.
0,0875	For application in a series of production projects.
Question 2: How the results were measured (weight = 25%)	
0,000	If it is unclear how the results were measured.
0,0250	For a subjective opinion such as a lesson learned.
0,0500	For a case study (i.e., the results were obtained within a project lifetime), or a loose comparison between practices.
0,0625	For a rigorous comparison with another practice, across projects, or with a baseline.
Question 3: How the evidence was reported (weight = 25%)	
0,0125	If the evidence was gained via an interview, or if the evidence was web-published as a report (e.g., white paper), expert opinion, or lesson learned.
0,0250	For publication as a technical report within a university or organization, or as part of workshop proceedings.
0,0375	For publication in a trade journal (e.g., CrossTalk).
0,0500	For publication in a conference or most academic journals.
0,0625	For publication in the highest-quality archival journals.
Question 4: Who reported the evidence (weight = 15%)	
0,0	If it is unclear who reported the results.
0,0250	For someone external to the team that applied the practice (i.e., a person who got the evidence second hand or after the fact).
0,0375	For someone who was part of the group that applied the practice.
Max summed: 0,25.	

Appendix B. SSM experimental study instruments and additional materials

NOTE: ALL INSTRUMENTS AND FORMS OF THIS APPENDIX WERE TRANSLATED FROM PORTUGUESE. THE STUDY USED THE ORIGINAL INSTRUMENTS, IN PORTUGUESE.

B.1. Researcher characterization form

Name ____ Level (M.Sc./D.Sc.): ____

General background

Please, evaluate your skills in using study instruments in English language:

__ I have fluency speaking, reading and writing in English.

__ I consider English a language where:

My skills in reading and comprehending texts:

__ could be better

__ are moderate

__ are high

__ are very high

My skills in working and following instructions:

__ could be better

__ are moderate

__ are high

__ are very high

What is your experience with software development practice? (mark those which apply)

__ I have never developed software

__ I am used to develop software for my own use

__ I am used to develop software as part of a software team in a course

__ I am used to develop software as part of a software team in industry

Please, explain your answer. If possible, mention how many semesters or years of relevant experience do you have in software development (*e.g.*, ‘I have worked for 10 years as software programmer in industry’).

Research experience

In questions of this section, please indicate your experience level using the following scale:

1 = no experience

2 = I have had courses or studied in books/articles

3 = I have practiced once in a graduate discipline

4 = I used it once in a research conducted by me

5 = I used it multiple times in a research conducted by me or collaborating with peers

Experience in data analysis

• Data analysis in general	1	2	3	4	5
• Descriptive statistics	1	2	3	4	5
• Inferential statistics (*e.g.*, statistical tests)	1	2	3	4	5
• Qualitative data analysis	1	2	3	4	5
• Text and data coding	1	2	3	4	5

Experience in research methods

• Planning and executing controlled studies	1	2	3	4	5
• Planning and executing case studies	1	2	3	4	5
• Planning and executing other primary study types	1	2	3	4	5
• Planning and executing systematic reviews	1	2	3	4	5

Experience in research synthesis

• Meta-analysis	1	2	3	4	5
• Other research synthesis methods	1	2	3	4	5

Software Engineering experience

In this section, we try to assess how familiar you are with the software engineering domains used for research synthesis. In questions of this section, please indicate your experience level using the following scale:

1 = I am not familiar with this area. I have never done this before.

3 = I have done that few times, but I’m not a specialist.

5 = I am very familiar with this area. I feel comfortable doing this.

• How much do you know about VV&T?	1	3	5
• How much do you know about?	1	3	5
• How much do you know about agile processes?	1	3	5
• How much do you know about prescriptive processes?	1	3	5

B.2. Study procedures script

Study procedures

The study goal is to evaluate the Structured Synthesis Method (SSM) in a set of primary studies regarding *Test-Driven Development*.

Study procedures are basically associated with SSM process steps, which are described below. Still, all training material is available for reference and can be used in any stage of the study. It should be noticed that some of SSM process steps will not be performed as the primary studies for aggregation are defined in advance. Some instruments are also provided with this script. They should be filled and must be returned back by the end of the study.

Important: any additional reference about the research synthesis theme (Test-Driven Development) should be read or searched during study execution.

SSM STEPS

1) Planning and definition: the study objectives are defined, including the research question, and the inclusion/exclusion criteria are formalized.

In this study, the research question for the synthesis study is defined as: ‘What are the expected effects of Test-Driven Development in software development projects?’

2) Selection: primary studies are collected in systematic manner considering the defined criteria. In this study, the following primary studies are defined for aggregation:

Paper 1: MARCHENKO, A.; ABRAHAMSSON, P.; IHME, T. Long-Term Effects of Test-Driven Development A Case Study. In: P. Abrahamsson; M. Marchesi; F. Maurer (Orgs.); **Agile Processes in Software Engineering and Extreme Programming**, Lecture Notes in Business Information Processing. p.13–22, 2009. Springer Berlin Heidelberg.

Paper 2: GEORGE, B.; WILLIAMS, L. A structured experiment of test-driven development. **Information and Software Technology**, v. 46, n. 5, p. 337–342, 2004.

Paper 3: SLYNGSTAD, O. P. N.; LI, J.; CONRADI, R.; et al. The Impact of Test Driven Development on the Evolution of a Reusable Framework of Components: An Industrial Case Study. **The Third International Conference on Software Engineering Advances**. ICSEA ’08. p.214–223, 2008.

Paper 4: ERDOGMUS, H.; MORISIO, M.; TORCHIANO, M. On the effectiveness of the test-first approach to programming. **IEEE Transactions on Software Engineering**, v. 31, n. 3, p. 226–237, 2005.

3) Quality assessment: the quality of primary studies is assessed using quality assessment checklists. The quality assessment is used as an input to estimate the confidence (*i.e.*, belief values) on the study results' causal and moderation relationships. Checklists are provided and should be filled as part of the study procedures.

4) Extraction and translation: for each primary study a theoretical structure must be modeled. Knowledge extraction is performed in terms of concepts and relationships identification, following the restrictions of the adopted diagrammatic model.

Theoretical structures can be sketched using Microsoft Visio tool (a trial version can be obtained in https://products.office.com/pt-br/Visio/). One example of a theoretical structure created with MS Visio is attached. All concepts, relationships and explanations should be textually described in specific forms that are provided as well.

5) Aggregation and analysis: based on the extracted theoretical structures compatible evidence is aggregated by pooling their effects and moderators. The results are, then, analyzed together.

In this study, it is not necessary to manually compute combined belief values using Dempster's rule of combination. Only the indication of evidence compatible evidence is sufficient for this step in the context of this study. There is a form for registering which evidence are compatible.

ADDITIONAL ITEMS

- The approximated time spent in steps 3, 4 and 5 should be registred.
- Participants are encouraged to take notes about misunderstandings or any other problem either with the study instruments or regarding SSM itself.

B.3. Belief value estimation form

Name: ____

Paper 1 – Long-Term Effects of Test-Driven Development A Case Study			
Study type	**Base belief value based on study type**	**Belief value increment based on quality assessment (up to 0.25)**	**Belief value**

Paper 2 – A structured experiment of test-driven development			
Study type	**Base belief value based on study type**	**Belief value increment based on quality assessment (up to 0.25)**	**Belief value**

Paper 3 – The Impact of Test Driven Development on the Evolution of a Reusable Framework of Components: An Industrial Case Study			
Study type	**Base belief value based on study type**	**Belief value increment based on quality assessment (up to 0.25)**	**Belief value**

Paper 4 – On the effectiveness of the test-first approach to programming			
Study type	**Base belief value based on study type**	**Belief value increment based on quality assessment (up to 0.25)**	**Belief value**

B.4. Theoretical structure textual description form

Name: ____

Paper: ____

Concepts	
C_1	
C_2	
C_3	
C_4	
C_5	
C_6	
C_7	
C_8	
C_9	
C_{10}	
Relationships	
R_1	
R_2	
R_3	
R_4	
R_5	
R_6	
R_7	
R_8	
R_9	
R_{10}	

Explanation	
E_1	
E_2	
E_3	
E_4	
E_5	
E_6	
E_7	
E_8	
E_9	
E_{10}	

B.5. Quantitative to semi-quantitative conversion form

Name: ____

Variable (e.g., #defects)	**Quantitative interval**	**Likert scale value**

B.6. Aggregation compatibility form

Name: ____

Primary studies aggregation compatibility

Studies	**Can be aggregated?**	**Please explain your answer in case evidence are not compatible in your understanding (you can cite incompatible concepts or any other reason for determining the incompatibility)**
Study 1 + Study 2		
Study 1 + Study 3		
Study 1 + Study 4		
Study 2 + Study 3		
Study 2 + Study 4		
Study 3 + Study 4		

B.7. Time spent form

Name: ____

Time spent (in minutes)
Papers Reading and steps 3 and 4 of SSM are performed individually for each paper. Hence, the time spent should be registered for each paper. Step 5 involves the analysis of all papers together and, thus, time spent is registered for the whole step. As SSM is an iterative process, it should be noticed that time can be approximated when registered in this form.

Paper 1 - Long-Term Effects of Test-Driven Development A Case Study		
Reading	**Step 3**	**Step 4**

Paper 2 - A structured experiment of test-driven development		
Reading	**Step 3**	**Step 4**

Paper 3 - The Impact of Test Driven Development on the Evolution of a Reusable Framework of Components; An Industrial Case Study		
Reading	**Step 3**	**Step 4**

Paper 4 - On the effectiveness of the test-first approach to programming		
Reading	**Step 3**	**Step 4**

Time spent in step 5: ________

Problems and suggestions

Please, describe any problem and suggestion in applying SSM for the four studies.

B.8. Post-study questionnaire

Post-study questionnaire

Nome: ____

Please, answer the questions below based on the procedures conducted within the study.

Structured Synthesis Method evaluation

1) Do the theoretical structures are able to capture the main results present in the primary studies?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

____

____

____

____

____

2) Do the archetypes support identifying which aspects of primary studies should be made explicit in the theoretical structure graphical representation?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

____

____

____

____

____

3) Are the archetypes equally important? Are they sufficient or other archetypes are needed?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer (cite archetypes that could be added or removed):

____

____

____

____

4) Do the theoretical structure could represent the aspects that you (the subject) considered most important?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer (if you disagree, cite what aspects you could not represent):

5) Would you modify any aspect of the theoretical structure diagrammatic representation?

6) Could the conversion from the quantitative scale to the semi-quantitative (*Likert*) scale used in the theoretical structures be done appropriately?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

7) Could the conversion from the qualitative scale to the semi-quantitative (*Likert*) scale used in the theoretical structures be done appropriately?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

8) Theoretical structures are useful to determine if the primary studies could be aggregated.

☐ Yes, it is possible to determine studies compatibility solely based on theoretical structures (including textual description of concepts, relationships and explanations).

☐ Yes, but reading papers from which theoretical structures were extracted improves the aggregation reliability (*i.e.*, less susceptible to validity threads).

☐ No, it is not possible to aggregate studies solely based on theoretical structures. It is indispensible to read the papers.

Justify your answer:

____

____

____

____

____

9) Is the belief level estimation (step 3) based on the evidence hierarchy and study quality checklist appropriate for the aggregation purposes?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

____

____

____

____

____

10) Is the belief level estimative (step 3), based on the evidence hierarchy and study quality checklist, appropriate for the aggregation purposes?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

____

____

____

____

____

11) Could the extraction step (step 4) be executed without difficulties for quantitative studies?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

____

____

____

____

____

12) Could the extraction step (step 4) be executed without difficulties for qualitative studies?

☐ completely disagree ☐ somewhat disagree ☐ somewhat agree ☐ completely agree

Justify your answer:

13) In your opinion, any step should be removed?

14) In your opinion, any step should be added?

Study procedures evaluation

15) How you evaluate the training that you received?

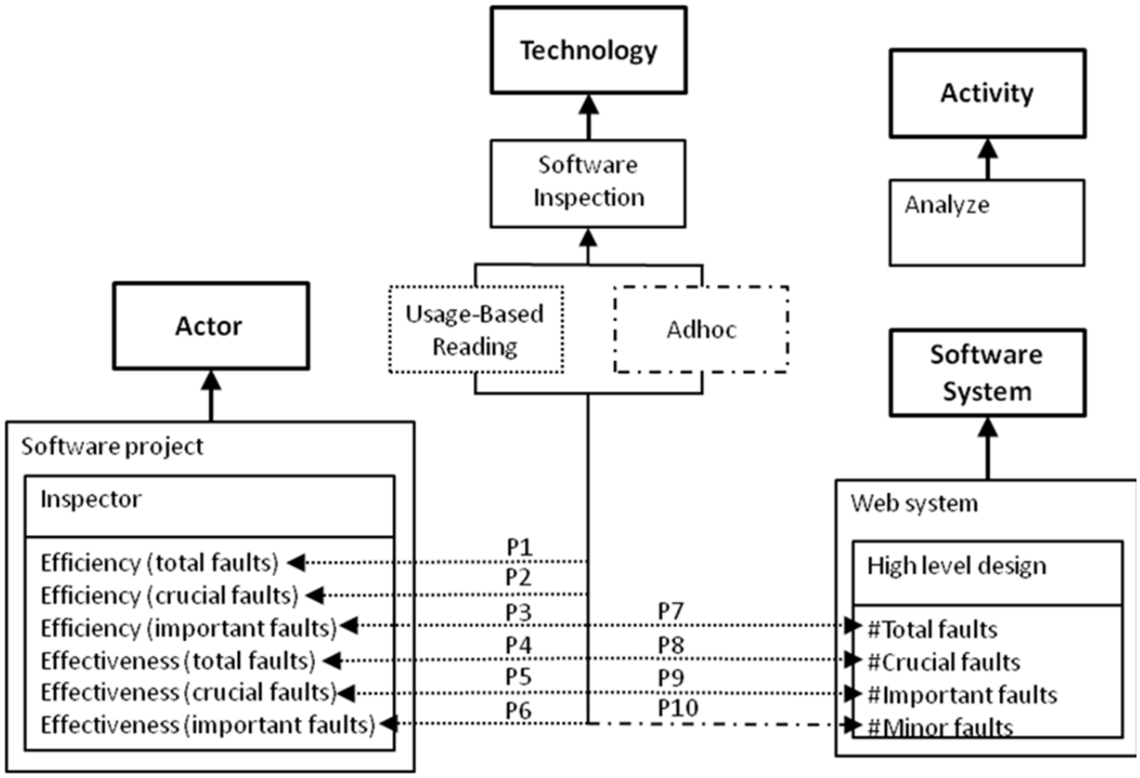
☐ insufficient ☐ neutral ☐ appropriate

If you have considered the training insufficient or neutral, what you would you change in training procedures?

16) Any other comment about study procedures, including training and execution itself?

Appendix C. Comparative theoretical structures for the research synthesis of Usage-Based Reading primary studies

C.1. Study S1 (Thellin *et al.*, 2001)



Constructs

C1 *Usage-Based Reading* (inspection technique that focus the reading effort on the most critical faults, from the user perspective, by using a set of use cases as a guide to orientate the inspection)

C2 *Efficiency* (the number of faults found per hour)

C3 *Effectiveness* (the percentage of the total number of faults found)

C4 *Total faults* (total number of faults found on the system)

C5 *Crucial faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are important for a user and often used)

C6 *Important faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are either important and rarely used or not as important but often used)

C7 *Minor faults* (the system will work although these faults are present)

C8 *Software project* (a typical software project team with no special characteristics)

C9 *Inspector* (a person who examines a software artifact to identify possible faults)

C10 *Web system* (a system that uses the internet infrastructure to operate)

C11 *High level design* (a design document which consists of an overview of the software modules and communication signals that are sent to and from the modules. It uses the specification description language.)

C12 *Ad hoc inspection* (a non-systematic inspection technique where no support or specific guidance is given to the reviewer)

Propositions

P1 Usage-Based Reading performs weakly better than ad hoc inspection in relation to efficiency associated with total faults (UBR=SP and Ad hoc=PO).

P2 Usage-Based Reading performs better than ad hoc inspection in relation to efficiency associated with crucial faults (UBR=PO-SP and Ad hoc=WP).

P3 Usage-Based Reading performs weakly better than ad hoc inspection in relation to efficiency associated with important faults (UBR=PO and Ad hoc=WP).

P4 Usage-Based Reading performs weakly better than ad hoc inspection in relation to the effectiveness associated with total faults (UBR=WP-PO and Ad hoc=WP).

P5 Usage-Based Reading performs better than ad hoc inspection in relation to effectiveness associated with crucial faults (UBR=PO-SP and Ad hoc=WP).

P6 Usage-Based Reading performs weakly better than ad hoc inspection in relation to effectiveness associated with important faults (UBR=PO and Ad hoc=WP).

P7 Usage-Based Reading performs weakly better than ad hoc inspection in relation to the identification of total faults (UBR=SP and Ad hoc=PO).

P8 Usage-Based Reading performs better than ad hoc inspection in relation to the identification of crucial faults (UBR=PO-SP and Ad hoc=WP).

P9 Usage-Based Reading performs weakly better than ad hoc inspection in relation to the identification of important faults (UBR=WP-PO and Ad hoc=WP).

P10 Usage-Based Reading performs weakly worse than ad hoc inspection in relation to the identification of minor faults (UBR=WP and Ad hoc=WP-PO).

Explanations

E1 The inspector identifies more faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.044
- The author does not present explicitly the efficiency, but since the experiment was time boxed (2.5h) the efficiency is directly associated with the number of faults identified (see E7). Using boxplot in Fig. 8 to calculate the means according to Hozo *et al.* (2005):
  - $\bar{X}_{ubr}$ = (2.9 + 2*5 + 8.8)/4 + (2.9 - 2*5 + 8.8)/ (4*27) = 5.425 + 0.016 = 5,441 (faults found per hour).
  - $\bar{X}_{adhoc}$ = (1.9 + 2*4.2 + 5.5)/4 + (1.9 - 2*4.2 + 5.5)/108 =3.95 - 0,009 = 3.941 (faults found per hour).
  - So according to above UBR faults 38% more defects than ad hoc by unit of time.

E2 The inspector identifies more crucial faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.

- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.0004
- The author does not present explicitly the efficiency, but since the experiment was time boxed (2.5h) the efficiency is directly associated with the number of faults identified (see E8). Using boxplot in Fig. 8 to calculate the means according to Hozo *et al.* (2005):
    - $\bar{X}_{ubr}$ = (1.3 + 2*2.3 + 4.2)/4 + (1.3 - 2*2.3 + 4.2)/108 = 2.525 + 0.008 = 2.533 (faults found per hour).
    - $\bar{X}_{adhoc}$ = (0 + 2*1.5 + 2.2)/4 + (0 - 2*1.5 + 2.2)/108 = 1.3 – 0,007 = 1.293 (faults found per hour).
    - So according to above UBR faults 95% more defects than ad hoc by unit of time.

E3 The inspector identifies more important faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.0049
- The author does not present explicitly the efficiency, but since the experiment was time boxed (2.5h) the efficiency is directly associated with the number of faults identified (see E9).

E4 The inspector identifies more unique faults considering the total number of faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (46%). The means calculated using Hozo *et al.* (2005) based on the data of the boxplot presented in Fig. 9:
        - $\bar{X}_{ubr}$ = (0.15 + 2*0.27 + 0.49)/4 + (0.15 - 2*0.27 + 0.49)/108 = 0.295 + 0.001 = 0.296 (faults found/total).
        - $\bar{X}_{adhoc}$ = (0.1 + 2*0.23 + 0.28)/4 + (0.1 - 2*0.23 + 0.28)/108 = 0.21 – 0.001 = 0.209 (faults found/total).
        - So according to above UBR faults 42% more unique defects than ad hoc.
- P = 0.0652

E5 The inspector identifies more unique faults considering the total number of crucial faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.

- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (63%). The means calculated using Hozo *et al.* (2005) based on the data of the boxplot presented in Fig. 9:
        - $\bar{X}_{ubr}$ = (0.23 + 2*0.38 + 0.69)/4 + (0.23 - 2*0.38 + 0.69)/108 = 0.42 + 0.001 = 0.421 (faults found/total).
        - $\bar{X}_{adhoc}$ = (0.0 + 2*0.23 + 0.38)/4 + (0.0 - 2*0.23 + 0.38)/108 = 0.21 - 0.001 = 0.209 (faults found/total).
        - So according to above UBR faults 100% more unique defects than ad hoc.
- P = 0.0017

E6 The inspector identifies more unique faults considering the total number of important faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (important + crucial = 42% → important < 42%).
- P = 0.0045

E7 An inspector identifies a larger number of faults on average

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- UBR detects 35% more faults than ad hoc on average. From Fig. 3 cumulative graph:
    - $\bar{X}_{ubr}\approx$ 11.5 faults.
    - $\bar{X}_{adhoc}\approx$ 8.5 faults.

E8 An inspector identifies a larger number of crucial faults on average

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- UBR detects 88% more crucial faults than ad hoc on average. From Fig. 4 cumulative graph:
    - $\bar{X}_{ubr}\approx$ 5.7 faults.
    - $\bar{X}_{adhoc}\approx$ 3.0 faults.

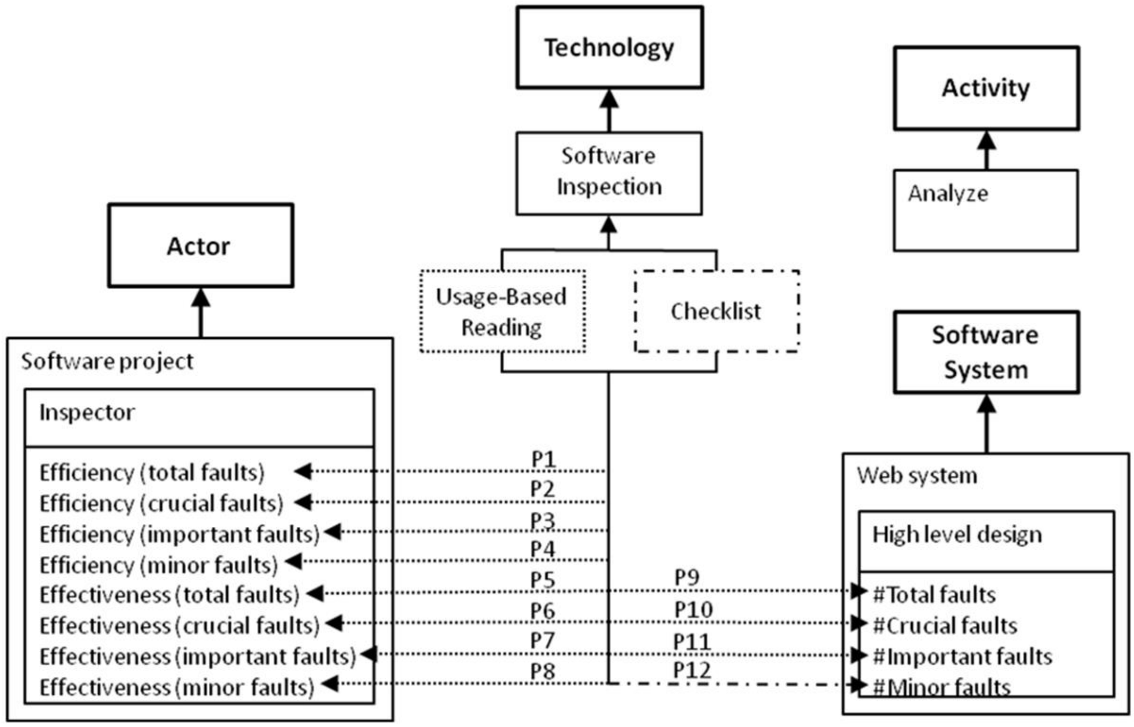
E9 An inspector identifies a larger number of important faults on average

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- UBR detects 51% more crucial + important faults than ad hoc on average. From Fig. 5 cumulative graph:
    - $\bar{X}_{ubr}\approx$ 9.5 – 5.7 = 3.8 faults.
    - $\bar{X}_{adhoc}\approx$ 5.8 – 3.0 = 2.8 faults.
    - (3.8 - 2.8)/2.8 = 36%

E10 An inspector does not focus on identifying minor faults on average

- When reviewer is not focused by what is understood as the most important functionality of the system, the inspector does not know what is critical to inspect and, thus, tend to identify a larger number of less significant faults.
- UBR detects 16% less minor faults than ad hoc on average

C.2. Study S2 (Thellin et al., 2003)



Constructs

C1 *Usage-Based Reading* (inspection technique that focus the reading effort on the most critical faults, from the user perspective, by using a set of use cases as a guide to orientate the inspection)

C2 *Efficiency* (the number of faults found per hour)

C3 *Effectiveness* (the percentage of the total number of faults found)

C4 *Total faults* (total number of faults found on the system)

C5 *Crucial faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are important for a user and often used)

C6 *Important faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are either important and rarely used or not as important but often used)

C7 *Minor faults* (the system will work although these faults are present)

C8 *Software project* (a typical software project team with no special characteristics)

C9 *Inspector* (a person who examines a software artifact to identify possible faults)

C10 *Web system* (a system that uses the internet infrastructure to operate)

C11 *High level design* (a design document which consists of an overview of the software modules and communication signals that are sent to and from the modules. It uses the specification description language.)

C12 *Checklist based inspection* (a semi-systematic inspection technique where the reviewer is guided by a list of characteristics which have to be revised)

Propositions

P1 Usage-Based Reading performs weakly better than checklist inspection in relation to efficiency associated with total faults (UBR=SP and Ad hoc=PO).

P2 Usage-Based Reading performs better than checklist inspection in relation to efficiency associated with crucial faults (UBR=PO-SP and Ad hoc=WP).

P3 Usage-Based Reading performs weakly better than checklist inspection in relation to efficiency associated with important faults (UBR=WP-PO and Ad hoc=WP).

P4 Usage-Based Reading performs weakly worse than checklist inspection in relation to efficiency associated with minor faults (UBR=WP and Ad hoc=PO).

P5 Usage-Based Reading performs weakly better than checklist inspection in relation to the effectiveness associated with total faults (UBR=PO and Ad hoc=WP-PO).

P6 Usage-Based Reading performs better than checklist inspection in relation to effectiveness associated with crucial faults (UBR=PO-SP and Ad hoc=WP).

P7 Usage-Based Reading performs weakly better than checklist inspection in relation to effectiveness associated with important faults (UBR=WP-PO and Ad hoc=WP).

P8 Usage-Based Reading performs worse than checklist inspection in relation to effectiveness associated with minor faults (UBR=IF-WP and Ad hoc=WP-PO).

P9 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of total faults (UBR=PO and Ad hoc=WP-PO).

P10 Usage-Based Reading performs better than checklist inspection in relation to the identification of crucial faults (UBR=PO-SP and Ad hoc=WP).

P11 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of important faults (UBR=WP-PO and Ad hoc=WP).

P12 Usage-Based Reading performs worse than checklist inspection in relation to the identification of minor faults (UBR=IF-WP and Ad hoc=WP-PO).

Explanations

E1 The inspector identifies more faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.042 (35% more)
  - $\bar{X}_{ubr}\approx$ 5.6 faults/hour.
  - $\bar{X}_{cbr}\approx$ 4.1 faults/hour.

E2 The inspector identifies more crucial faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.

- P = 0.013 (95% more)
    - $\bar{X}_{ubr}$≈ 2.6 faults/hour.
    - $\bar{X}_{cbr}$≈ 1.3 faults/hour.

E3 The inspector identifies more important faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.148 (46% more)
    - $\bar{X}_{ubr}$≈ 2.1 faults/hour.
    - $\bar{X}_{cbr}$≈ 1.4 faults/hour.

E4 The inspector identifies less minor faults by unit of time (hour)

- When reviewer is not focused by what is understood as the most important functionality of the system, the inspector does not know what is critical to inspect and, thus, tend to identify a larger number of less significant faults.
- P = 0.268 (49% less)
    - $\bar{X}_{ubr}$≈ 0.9 faults/hour.
    - $\bar{X}_{cbr}$≈ 1.4 faults/hour.

E5 The inspector identifies more unique faults considering the total number of faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (21%).
    - $\bar{X}_{ubr}$≈ 0.31 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.25 faults found/total.
- P = 0.103

E6 The inspector identifies more unique faults considering the total number of crucial faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (75%).
    - $\bar{X}_{ubr}$≈ 0.43 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.24 faults found/total.
- P = 0.036

E7 The inspector identifies more unique faults considering the total number of important faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.

- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (28%).
    - $\bar{X}_{ubr}$≈ 0.31 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.24 faults found/total.
- P = 0.175

E8 The inspector identifies less unique faults considering the total number of minor faults

- When reviewer is not focused by what is understood as the most important functionality of the system, the inspector does not know what is critical to inspect and, thus, tend to identify a larger number of less significant faults.
    - So less unique faults are identified (63%).
    - $\bar{X}_{ubr}$≈ 0.18 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.30 faults found/total.
- P = 0.148

E9 An inspector identifies a larger number of faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E5.

E10 An inspector identifies a larger number of crucial faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E6.

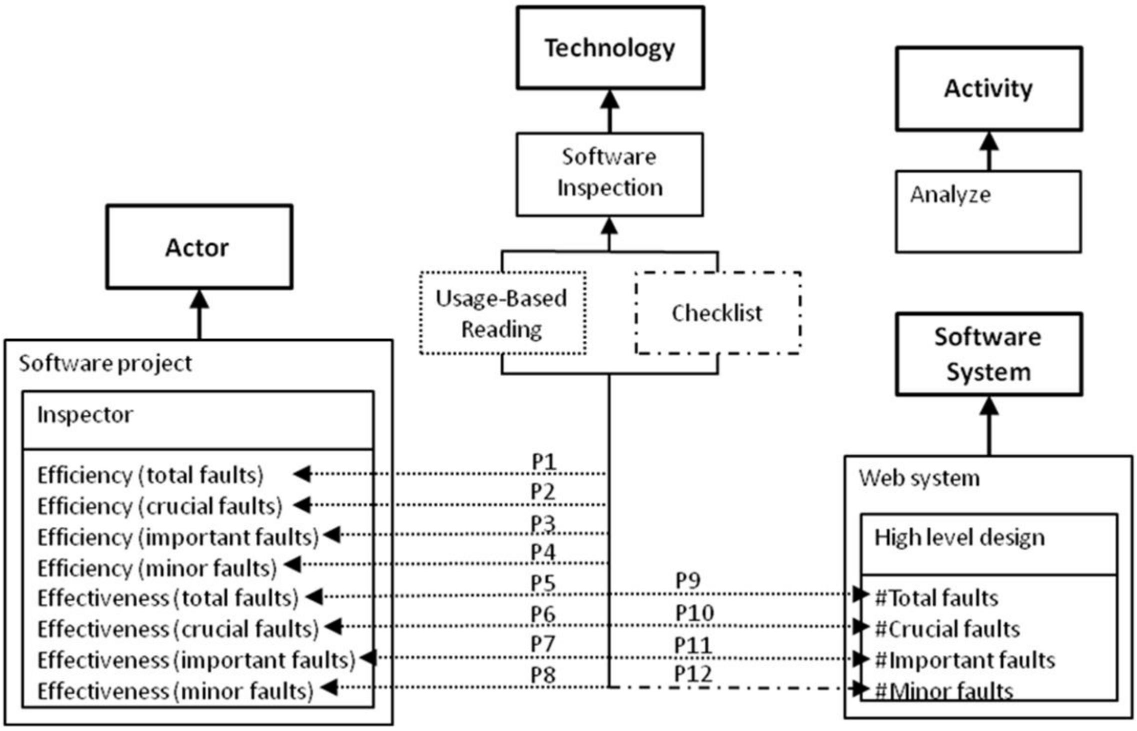
E11 An inspector identifies a larger number of important faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E7.

E12 An inspector does not focus on identifying minor faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E8.

C.3. Study S3 (Thellin et al., 2004)



Constructs

C1 *Usage-Based Reading* (inspection technique that focus the reading effort on the most critical faults, from the user perspective, by using a set of use cases as a guide to orientate the inspection)

C2 *Efficiency* (the number of faults found per hour)

C3 *Effectiveness* (the percentage of the total number of faults found)

C4 *Total faults* (total number of faults found on the system)

C5 *Crucial faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are important for a user and often used)

C6 *Important faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are either important and rarely used or not as important but often used)

C7 *Minor faults* (the system will work although these faults are present)

C8 *Software project* (a typical software project team with no special characteristics)

C9 *Inspector* (a person who examines a software artifact to identify possible faults)

C10 *Web system* (a system that uses the internet infrastructure to operate)

C11 *High level design* (a design document which consists of an overview of the software modules and communication signals that are sent to and from the modules. It uses the specification description language.)

C12 *Checklist based inspection* (a semi-systematic inspection technique where the reviewer is guided by a list of characteristics which have to be revised)

Propositions

P1 Usage-Based Reading performs weakly better than checklist inspection in relation to efficiency associated with total faults (UBR=WP-PO and Ad hoc=WP).

P2 Usage-Based Reading performs better than checklist inspection in relation to efficiency associated with crucial faults (UBR=WP-PO and Ad hoc=IF-WP).

P3 Usage-Based Reading performs weakly better than checklist inspection in relation to efficiency associated with important faults (UBR=WP and Ad hoc=IF-WP).

P4 Usage-Based Reading performs indifferently than checklist inspection in relation to efficiency associated with minor faults (UBR=WP and Ad hoc=WP).

P5 Usage-Based Reading performs weakly better than checklist inspection in relation to the effectiveness associated with total faults (UBR=PO and Ad hoc=WP).

P6 Usage-Based Reading performs better than checklist inspection in relation to effectiveness associated with crucial faults (UBR=PO-SP and Ad hoc=WP).

P7 Usage-Based Reading performs weakly better than checklist inspection in relation to effectiveness associated with important faults (UBR=PO and Ad hoc=WP).

P8 Usage-Based Reading performs indifferently than checklist inspection in relation to effectiveness associated with minor faults (UBR=WP and Ad hoc=WP).

P9 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of total faults (UBR=PO and Ad hoc=WP).

P10 Usage-Based Reading performs better than checklist inspection in relation to the identification of crucial faults (UBR=PO-SP and Ad hoc=WP).

P11 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of important faults (UBR=PO and Ad hoc=WP).

P12 Usage-Based Reading performs indifferently than checklist inspection in relation to the identification of minor faults (UBR=WP and Ad hoc=WP).

Explanations

E1 The inspector identifies more faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.037 (41% more)
  - $\bar{X}_{ubr}$≈ 4.8 faults/hour.
  - $\bar{X}_{cbr}$≈ 3.4 faults/hour.

E2 The inspector identifies more crucial faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.001 (91% more)
  - $\bar{X}_{ubr}$≈ 2.1 faults/hour.
  - $\bar{X}_{cbr}$≈ 1.1 faults/hour.

E3 The inspector identifies more important faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.

- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- No statistical test was performed (29% more)
    - $\bar{X}_{ubr}$≈ 1.8 faults/hour.
    - $\bar{X}_{cbr}$≈ 1.4 faults/hour.

E4 No difference on the identification of minor faults by unit of time (hour)

- 0% difference, no statistical test was performed.
    - $\bar{X}_{ubr}$≈ 0.9 faults/hour.
    - $\bar{X}_{cbr}$≈ 0.9 faults/hour.

E5 The inspector identifies more unique faults considering the total number of faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (45%).
    - $\bar{X}_{ubr}$≈ 0.29 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.2 faults found/total.
- P = 0.006

E6 The inspector identifies more unique faults considering the total number of crucial faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (85%).
    - $\bar{X}_{ubr}$≈ 0.37 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.2 faults found/total.
- P = 0.001

E7 The inspector identifies more unique faults considering the total number of important faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
    - So more unique faults are identified (50%).
    - $\bar{X}_{ubr}$≈ 0.30 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.2 faults found/total.
- no statistical test was performed.

E8 No difference on the identification of unique faults considering the total number of minor faults

- 0% difference, no statistical test was performed.

    - $\bar{X}_{ubr}$≈ 0.2 faults found/total.
    - $\bar{X}_{cbr}$≈ 0.2 faults found/total.

E9 An inspector identifies a larger number of faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E5.

E10 An inspector identifies a larger number of crucial faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E6.

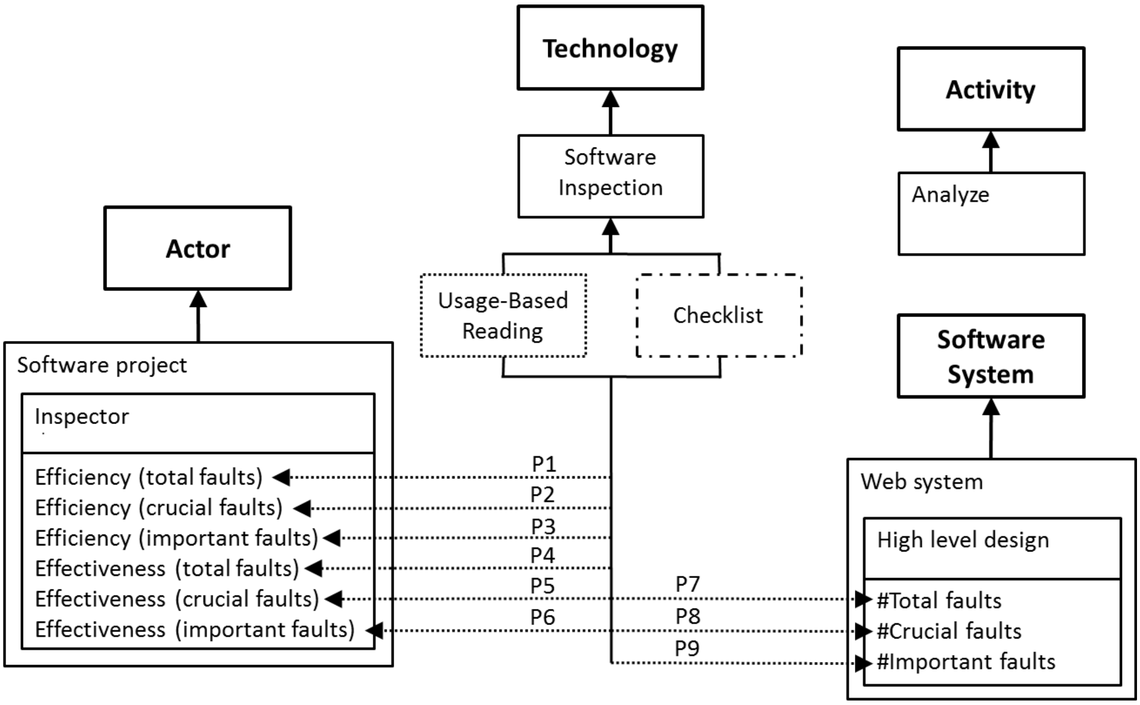
E11 An inspector identifies a larger number of important faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E7.

E12 An inspector does not focus on identifying minor faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E8.

C.4. Study S4 (Winkler et al., 2004)



Constructs

C1 *Usage-Based Reading* (inspection technique that focus the reading effort on the most critical faults, from the user perspective, by using a set of use cases as a guide to orientate the inspection)

C2 *Efficiency* (the number of faults found per hour)

C3 *Effectiveness* (the percentage of the total number of faults found)

C4 *Total faults* (total number of faults found on the system)

C5 *Crucial faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are important for a user and often used)

C6 *Important faults* (functions affected by these faults are crucial in the user perspective, i.e. the functions affected are either important and rarely used or not as important but often used)

C7 *Software project* (a typical software project team with no special characteristics)

C8 *Inspector* (a person who examines a software artifact to identify possible faults)

C9 *Web system* (a system that uses the internet infrastructure to operate)

C10 *High level design* (a design document which consists of an overview of the software modules and communication signals that are sent to and from the modules. It uses the specification description language.)

C11 *Checklist based inspection* (a semi-systematic inspection technique where the reviewer is guided by a list of characteristics which have to be revised)

Propositions

P1 Usage-Based Reading performs <u>weakly better</u> than checklist inspection in relation to efficiency associated with total faults (UBR=PO and Ad hoc=WP).

P2 Usage-Based Reading performs <u>better</u> than checklist inspection in relation to efficiency associated with crucial faults (UBR=WP-PO and Ad hoc=IF-WP).

P3 Usage-Based Reading performs <u>indifferent</u> than checklist inspection in relation to efficiency associated with important faults (UBR= IF-WP and Ad hoc=IF-WP).

P4 Usage-Based Reading performs weakly better than checklist inspection in relation to the effectiveness associated with total faults (UBR=SP and Ad hoc=PO).

P5 Usage-Based Reading performs weakly better than checklist inspection in relation to effectiveness associated with crucial faults (UBR=SP and Ad hoc=PO).

P6 Usage-Based Reading performs indifferent than checklist inspection in relation to effectiveness associated with important faults (UBR= IF-WP and Ad hoc=IF-WP).

P7 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of total faults (UBR=SP and Ad hoc=PO).

P8 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of crucial faults (UBR=SP and Ad hoc=PO).

P9 Usage-Based Reading performs weakly better than checklist inspection in relation to the identification of important faults (UBR= IF-WP and Ad hoc=IF-WP).

Explanations

E1 The inspector identifies more faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.053 (25% more)
  - $\bar{X}_{ubr}$≈ 4.5 faults/hour.
  - $\bar{X}_{cbr}$≈ 3.6 faults/hour.

E2 The inspector identifies more crucial faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- P = 0.002 (63% more)
  - $\bar{X}_{ubr}$≈ 1.8 faults/hour.
  - $\bar{X}_{cbr}$≈ 1.1 faults/hour.

E3 The inspector identifies more important faults by unit of time (hour)

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- In this case, they detect the more important faults ranked from a user's point of view.
- More faults are found during the first part of an inspection and the longer an inspection last the less faults are found due to lack of concentration.
- No statistical test was performed (13% more)
  - $\bar{X}_{ubr}$≈ 1.8 faults/hour.
  - $\bar{X}_{cbr}$≈ 1.6 faults/hour.

E4 The inspector identifies more unique faults considering the total number of faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
  - So more unique faults are identified (19%).
  - $\bar{X}_{ubr}$≈ 0.76 faults found/total.
  - $\bar{X}_{cbr}$≈ 0.64 faults found/total.
- P = 0.018

E5 The inspector identifies more unique faults considering the total number of crucial faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
  - So more unique faults are identified (46%).
  - $\bar{X}_{ubr}$≈ 0.92 faults found/total.
  - $\bar{X}_{cbr}$≈ 0.63 faults found/total.
- P = 0.001

E6 The inspector identifies more unique faults considering the total number of important faults

- The result of the experiment shows that it is possible to control reviewers in order to make them focus on important parts of a software artifact.
- The prioritization of use cases forces the covering an adequate part of the software artifact inspected and thus it is easier to find faults guided by these use cases.
  - So more unique faults are identified (40%).
  - $\bar{X}_{ubr}$≈ 0.07 faults found/total.
  - $\bar{X}_{cbr}$≈ 0.05 faults found/total.
- no statistical test was performed.

E7 An inspector identifies a larger number of faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E4.

E8 An inspector identifies a larger number of crucial faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.
- See E5.

E9 An inspector identifies a larger number of important faults on average

- The author do not present explicitly the number of defects found, but since the number of faults present in the artifact reviewed is the same, we are considering the number of faults identified enough close to the average inspector effectiveness.

- See E6.

Appendix D. Evidence Factory requirements

To illustrate the main concerns addressed by Evidence Factory we organize the them in Figure 48. In general terms, four main groups are identified: storage and processing (including search mechanisms); facilities for researchers; facilities for practitioners; and visualization, information provision and social network. This Appendix enumerates the requirements for the Evidence Factory tool according to these groups.

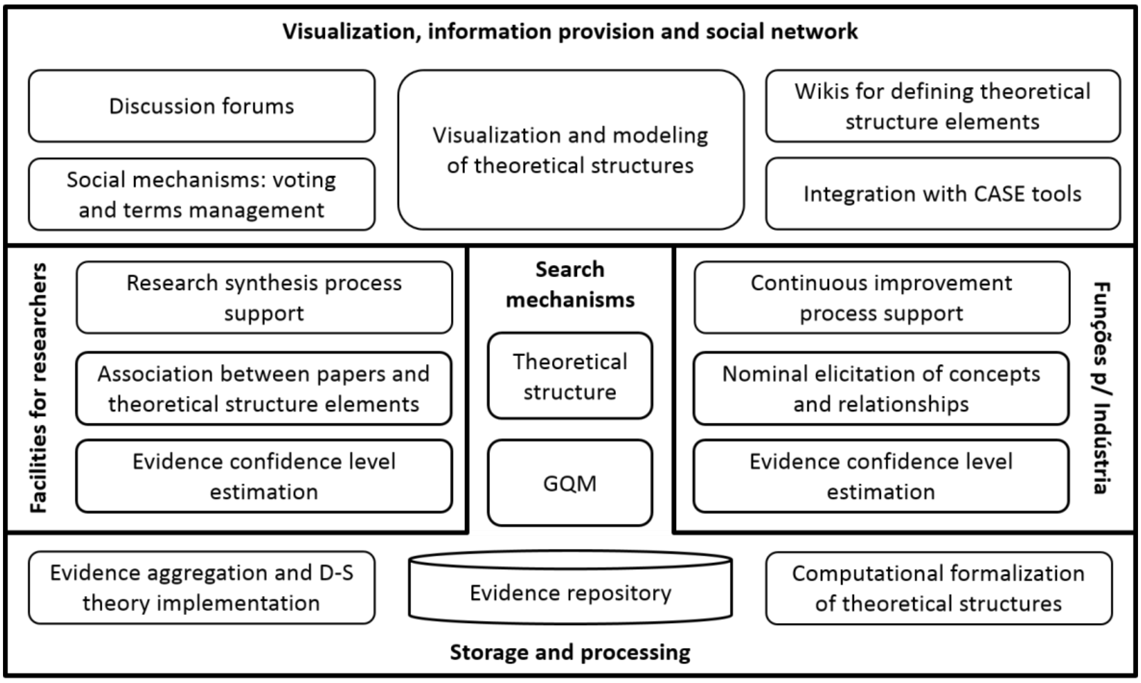

Visualization, information provision and social network
Discussion forums
Social mechanisms: voting and terms management
Visualization and modeling of theoretical structures
Wikis for defining theoretical structure elements
Integration with CASE tools
Facilities for researchers
Research synthesis process support
Association between papers and theoretical structure elements
Evidence confidence level estimation
Search mechanisms
Theoretical structure
GQM
Continuous improvement process support
Nominal elicitation of concepts and relationships
Evidence confidence level estimation
Funções p/ Indústria
Evidence aggregation and D-S theory implementation
Evidence repository
Computational formalization of theoretical structures
Storage and processing

Figure 48 – Conceptual organization of the computerized infrastructure

D.1. Functional requirements

D.1.1. Storage and processing

- D-S theory algorithms – implementation of the D-S theory conceptualization: frame of discernment, basic probability assignment function, belief function and Dempster's rule of combination.
- Formalization of theoretical structures – the infrastructure must have a formal definition of a theoretical structure, defining all restrictions associated with the usage of each type of concept and relationship.
- Causal relationship quantification rule – the infrastructure must allow the definition of conversion rules from quantitative data to the *Likert* scale used in effect intensity. The rules should be available to be reused in the effects of

different theoretical structures in order to keep consistent the usage of the *Likert* scale.

- GQM associations – the infrastructure must allow the creation of GQMs and associate them to theoretical structures. Example: the theoretical structure TS_1 was used as part of the goal defined in the GQM_1. Moreover, there must be a consistence between the metrics defined in the GQM and the variable concepts defined in the theoretical structure.
- Search mechanisms – the infrastructure must support three different ways for searching evidence:
    - By using a GQM as a search criterion. In this case, the returned theoretical structures should have part of the metrics defined in the GQM as its variable concepts.
    - By defining a theoretical structure fragment, which must be present in the returned results.
    - By keyword present in the any part of the theoretical structure model, from the concepts to the textual explanations associated with causal relationships.
- Search results – for each concept of the theoretical structures returned in a search, the infrastructure must present in how many other theoretical structures each of them are used.
- Indicators for evaluation criteria – the infrastructure must provide the following indicators for the following evaluation criteria:
    - Experimental support ([0,1]): defined by the belief values mean of the causal and moderation relationships.
    - Explanatory power ([0,1]): provide an indication for generalization level of a theoretical structure. Explanatory power = [#compatible theoretical structures] / [#compatible theoretical structures related to the same cause].
    - Parsimony ([1,∞], lower values indicate greater level of parsimony): it is computed based on the number of used concepts and the explanatory power. Parsimony = [#concepts in the theoretical structure] / [(#compatible theoretical structures related to the same cause])$_{\text{max explanatory power}}$.
- Terms glossary – the infrastructure must keep terms used in all theoretical structures created. Moreover, it must be possible to define synonyms between them. There should also be a possibility to have alternative definitions for the same term. Therefore, when terms are used in theoretical structures the user must inform the definition that will be associated to it.

D.1.2. Facilities for researchers

- Support for SSM – the infrastructure must support the instantiation of a synthesis study so that all information of the study is kept in that context. Following SSM phases, the requirements to support a synthesis study are enumerated:
  - Step 1 – registration of information related to the research question, inclusion and exclusion criteria, search string and general keywords.
  - Step 2 – the infrastructure must allow the researcher to inform the papers returned for the search string and define those which were included and excluded. Each paper excluded should be associated with the inclusion criteria that it does not meet or the exclusion criteria that caused its exclusion. It should be possible to upload a set of papers in batch using the BibTex format.
  - Step 3 – the infrastructure must provide a closed-ended questionnaire based on the quality assessment checklists. A questionnaire must be answered for each paper considering the study type. Therefore, the infrastructure must allow the determination of the study type for each paper.
  - Step 4 – the infrastructure must have a graphical editor for theoretical structures.
  - Step 5 – the infrastructure must support the theoretical structure aggregation:
    - The infrastructure must be able to determine the compatibility between two theoretical structures according to their concepts and relationships. For those partially compatible, the infrastructure should make explicit what make them incompatible. These differences form the model conflicts.
    - The infrastructure must implement the following conflicting resolution rules: ignore, explain and incorporate.
    - The infrastructure must compute the moderation and causal relationships aggregated belief values according to the Dempster's rule of combination.
- Traceability among synthesis main elements – traces can be of three kinds:
  - Between research synthesis and papers.
  - Between papers and theoretical structures – the infrastructure must keep the link between papers and theoretical structures so that multiple theoretical structures can be created for the same paper.
  - Between concepts/relationships and text (excerpt of the paper from which the concept/relationship came).

- Collaborative synthesis – the infrastructure must support that more than one research conduct a research synthesis.
- Belief value determination – the infrastructure must compute the belief value according to the SSM definitions based on the study type and quality assessment.

D.1.3. Facilities for practitioners

- Support for nominal elicitation activities in continuous improvement cycle – in a nominal (Bjørnson *et al.*, 2009) experience collection phase (Salo and Abrahamsson, 2007) of the continuous improvement cycle the infrastructure must allow that each developer individually registers your experience regarding the goal set (i.e., GQM).
- Context delimitation – contextual delimitation should be available for the characterization scenario, where there is no theoretical structure compatible with the goals set (i.e., GQM). In these cases, the infrastructure must support the software team to model a new theoretical structure:
    - Initial project characterization – this characterization should be performed once, at the beginning of the project. The idea is to identify the value concepts deriving from Actor, System and Software. In this way, any time that a new theoretical structure is modeled in the context of this project there is no need to define these concepts multiple times.
    - Additional value constructs – the value constructs identified in the initial characterization can be insufficient in the cases where the object under observation requires additional contextual information to define variable concepts (with *property of* relationships). To associate variable concepts to value concepts the following algorithm is defined:
        - The infrastructure shows the first level value concepts.
        - User associate the appropriate variable concepts to the value concepts at that level.
        - Repeat for the next levels or until there is variable concept to be associated.
        - If the last level is reached and there is still variable concepts not associated then the infrastructure must allow the user to create new value concepts.
- Support for continuous improvement process (steps from Salo and Abrahamsson (2007) not cited here do not need tool support as, for instance, is the preparation phase) –

- Experience collection – after nominally eliciting positive and negative experiences, the infrastructure must allow registering the negative ones and choosing among them which is going to be addressed in the next improvement cycle.
- Planning of improvement actions –
  - Search for registered goal for reuse – in this case, the team works iteratively and, after defining an improvement goal, can search for improvement goals used by other software projects. From these goals, the infrastructure must indicate what theoretical structures were generated so that it can be set as the observation focus for an improvement cycle.
  - Metrics and indicators definition – two possible situations here: (i) all variable concepts of the selected theoretical structure are mapped to the GQM defined for this improvement cycle and (ii) user want to add/remove variable concepts from an partially compatible theoretical structure.
  - Creation of theoretical structures – if there is no goal and theoretical structure aligned with the improvement needs, the infrastructure must support the software engineer defining a new one. The idea here is to support the user in defining a GQM and associating it to a new theoretical structure, supporting in the definition of value and variable concepts.
- Piloting –
  - Weekly meeting – the infrastructure must allow software engineers to register partial evaluations regarding the current focus of the improvement cycle. This information will be used to the follow-up and validation phase.
  - Visualization – the infrastructure must make available a visualization with information regarding the current improvement actions.
- Follow-up and validation – the infrastructure must provide support for these nominal tasks:
  - Definition of effects and moderators intensities.
  - Definition of belief value for effects and moderators according to follow-up questionnaire (as an unsystematic observation, the max value is 0.25 summing all three questions below).

	Questions and answers (closed-ended)	**Add to belief**
1)	**What percentage of time or situations the *practice* under observation was in fact applied over the course of time or situations in which it was considered applicable?**	
1.1	Between 80% and 100% of time or situations	0.083
1.2	Between 60% and 80% of time or situations	0.062
1.3	Between 40% and 60% of time or situations	0.041
1.4	Between 20% and 40% of time or situations	0.021
1.5	Between 0% and 20% of time or situations	0.0
2)	**What is conformance degree to the *practice* specifications? What percentage of the instructions, activities and procedures related to the *practice* were followed?**	
2.1	Between 80% and 100% of instructions, activities and procedures related to the practice	0.083
2.2	Between 60% and 80% of instructions, activities and procedures related to the practice	0.062
2.3	Between 40% and 60% of instructions, activities and procedures related to the practice	0.041
2.4	Between 20% and 40% of instructions, activities and procedures related to the practice	0.021
2.5	Between 0% and 20% of instructions, activities and procedures related to the practice	0.0
3)	**What is the similarity level among the software process activities before and during the improvement cycle over which the *practice* was used?**	
3.1	The software development process remained stable	0.083
3.2	Few activities were changed, included or removed. Just some needed adjustments to the process.	0.055
3.3	Several activities were changed, included or removed. However, the changes were focused in one process phase (e.g., codification or test).	0.028
3.4	Several activities were changed, included or removed over different process phases.	0.0

D.1.4. Visualization, information provision and social network

- Graphical modeling support – the infrastructure must support the creation of theoretical structures in a diagrammatic format.
- Forums and wikis – the infrastructure must provide means for content creation offering discussion forums related to theoretical structures and community edition of text regarding the aggregation findings.
- Social network mechanisms – the infrastructures users must be able to vote up or down a theoretical structure in a general way of making explicit the community approval or disapproval regarding its quality or importance.

- Integration with software development tools – the infrastructure must be extensible in such a way that rules for effects and moderator intensities can be defined using information from software development tools.

D.2. Non-functional requirements

- The infrastructure must be constructed as a Web application.
- The infrastructure must provide access to internal information through an open API using Web services.
- The infrastructure must be available on mobile devices (for nominal activities of facilities for practitioners).