The Knowledge Engineering Review, Vol. 31:2, 167–199.
doi:10.1017/S0269888916000011

Scientific Knowledge Engineering: a conceptual delineation and overview of the state of the art

PAULO SÉRGIO M. DOS SANTOS and GUILHERME H. TRAVASSOS

Ilha do Fundão, Centro de Tecnologia, Bloco H, Sala 317, Rio de Janeiro, RJ, Brazil
e-mail: pasemes@cos.ufrj.br, ght@cos.ufrj.br

Abstract

As a community work, 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. In this paper, we will 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. This view is also comparatively built, contrasting it with alternatives for manipulating scientific knowledge, namely data-intensive approaches and the computational discovery of scientific knowledge. The current state of the art is presented with 22 knowledge representations and computational infrastructure implementations, with their main relevant properties analyzed and compared. Based on this review and on the theoretical foundations of Knowledge Engineering, a high level step-by-step approach for specifying and constructing scientific computational environments is described. The paper concludes by indicating paths for further development of the view initiated here, especially related to the technical specificities that originates from applying Knowledge Engineering to scientific knowledge.

1 Introduction

The production of scientific knowledge is reaching amazing heights in the last decades. Therefore, 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 & 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 use have been done 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 paper, we suggest that the discipline of Knowledge Engineering (KE) offers a solid foundation upon which this vision can be built. 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 & 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 others were also directly translated from the business to the scientific domain.

Thus, the goal of this paper is to present a conceptual construction of SKE based in a theoretical correspondence with KE 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 2. Then, although not yet recognized as an area, the essence of SKE is implicitly present in many recent works, which were used to characterize the state of the art in Section 3 literature survey. Based on that survey, Section 4 details the fundamental aspects of SKE and presents a step-by-step approach to plan SKE projects. Next, Section 5 presents a real case application of the approach as a preliminary evaluation. Finally, in Section 6 the final considerations are discussed, including work limitations and future work paths.

2 Motivation 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[1] 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 & 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[2] (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 & Bostrom, 1987): knowledge acquisition and development of knowledge-based

[1] 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 this kind of system does not intend to replace the scientist expertise but to boost it. It is also consonant with Hars (2001) which uses the term Scientific Knowledge Infrastructure.

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

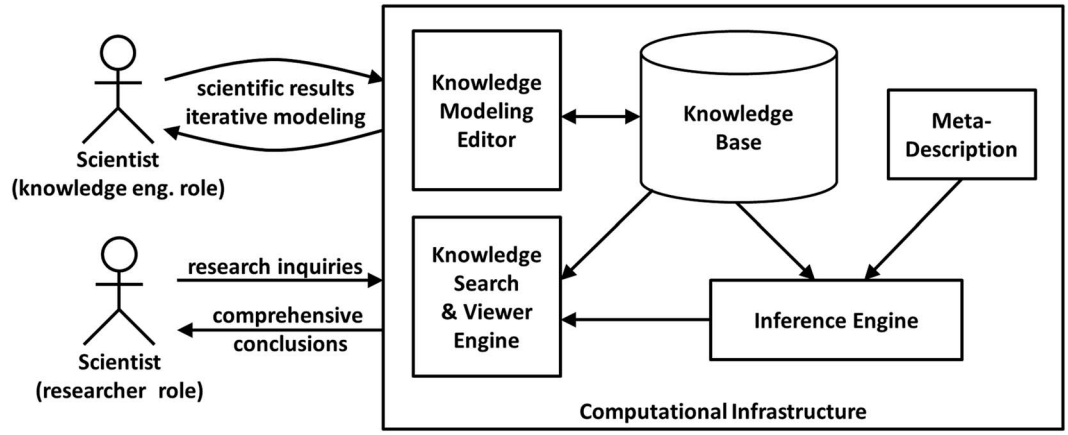

Figure 1 A typical organization in a Scientific Knowledge Engineering computational infrastructure

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 with 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 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) prescribing what domain knowledge is required to enable the knowledge-based system to perform its functions (Wielinga *et al*., 1992).

In SKE, KE is focussed 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 1 shows a typical organization of SKE computational infrastructures.

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; Dibble & Bostrom, 1987; Fellers, 1987; Rook & 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 increasing 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 can not 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 with other models?

Arguably, all of these questions can be investigated by searching digital libraries using keywords and performing 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 which 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 & 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. Literature reviews (systematic (Sackett *et al*., 1996) or not) are a commonly used tool to synthesize and condense this body of knowledge by answering research questions, exposing research gaps or guiding to new hypotheses, but they tend to get outdated in a relatively short time in most of the scientific domains and, not uncommonly, have a generalist nature which requires careful examination before their results can be applied to specific settings. SKE can support and automate the many steps involved in fully exploiting the scientific body of knowledge, defining how to develop computational infrastructures answering questions as specific as necessary (or representing knowledge capable of formalizing it) based on the current knowledge state.

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, the knowledge base 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. They 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 focusses 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 focussed 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 focusses 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). Thus, even when works seem to have SKE features (e.g. Stock *et al*., 2009) we focussed on characterizing them according to these specific aspects.

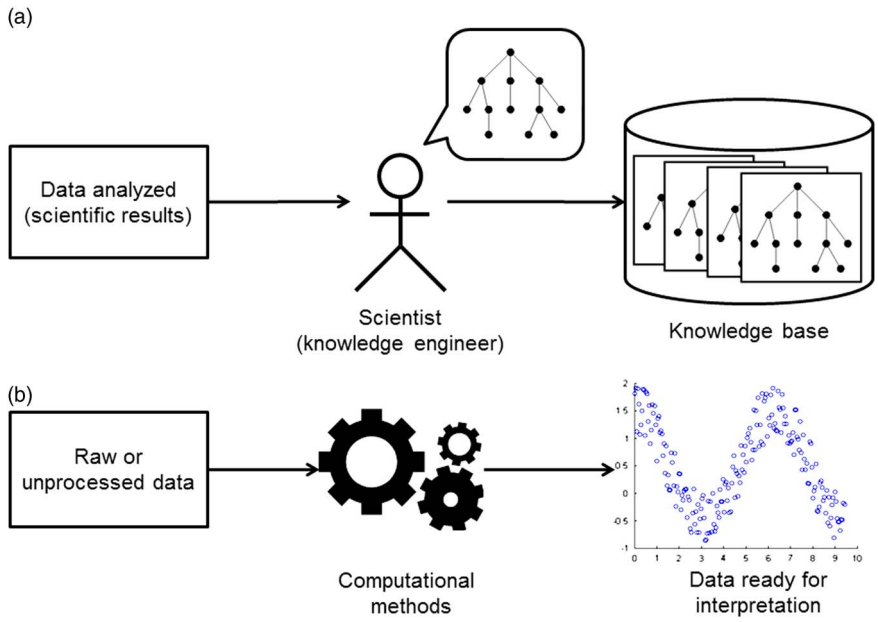

Figure 2 Comparison between (a) scientific knowledge engineering and (b) data-intensive approaches

2.1 Data-intensive approaches ('big data')

One of the alternatives that has been quite explored to support scientific investigations can be named as 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 & Aloisio, 2011). In data-intensive approaches, useful data is commonly found in digital format, usually originating from sensors, satellites, or scientific data repositories (e.g. gene and protein archives) (Hey & Trefethen, 2003). From these sources, data is computationally processed to support scientific investigation through the use of, for instance, data mining techniques (Fayyad & 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 2). Data-intensive approaches can be characterized by the use of relatively more raw data and usually aims at identifying patterns in a huge mass of data. On the other hand, SKE focusses 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.

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. 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. After computational techniques and algorithms are applied, processed data can be subjected to researchers' examination in answering the investigated research questions. 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, for example, 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) add 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 (Cohen & Hersh, 2005; Mons, 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'.

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 1970s (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 focusses on devising automated reasoning mechanisms that can emulate the whole creative mental processes involved in the generation of such scientific results (Figure 3). 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 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 emulate the researchers' discovery process. Comparatively, by keeping the human factor in scientific creative reasoning, SKE is less influenced to do that—even though, as we will discuss in Section 3, scientific methodologies also affect how scientific results

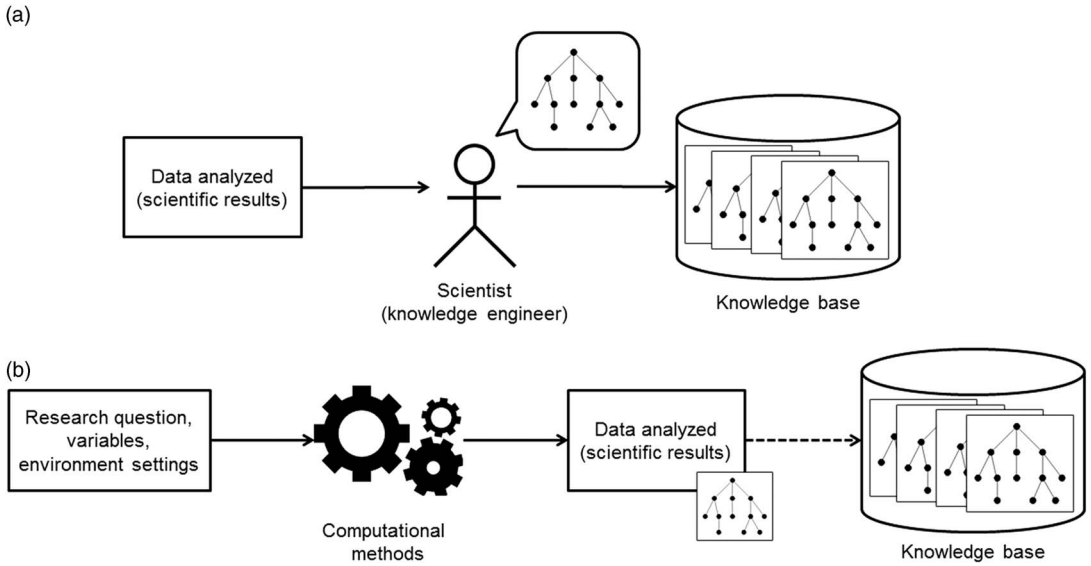

Figure 3 Comparison between (a) scientific knowledge engineering and (b) computational discovery of scientific knowledge

are described and organized, and thus 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.

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 also include some kind of suppositions and analogies.

Some common aspects between 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 3(b) 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).

3 The state of the art on Scientific Knowledge Engineering: a literature review

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 as follows: (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. As a consequence 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 ‘KE’, ‘knowledge representation’ and ‘ontology’, and ‘evidence’ and ‘science’[3]. 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

[3] To have an impression of the precision at hand with a more structured search string, we compiled one *a posteriori*, that is, after we identified the papers in this section. The string captures two dimensions, namely ‘KE’ 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 (‘KE’ 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, >13 000 papers were returned which clearly shows an absence of consensus in terminology, but also indicates the diversity of SKE applications.

search—using article, abstract and keyword fields. Supplementing Scopus, the Google Scholar search engine was also used, but only for snowballing.

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 focussed 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 focussed 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 papers' quality.

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 follows:

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.

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 4), but it can at least suggest that SKE is emerging as a research area. From the 22 articles, three

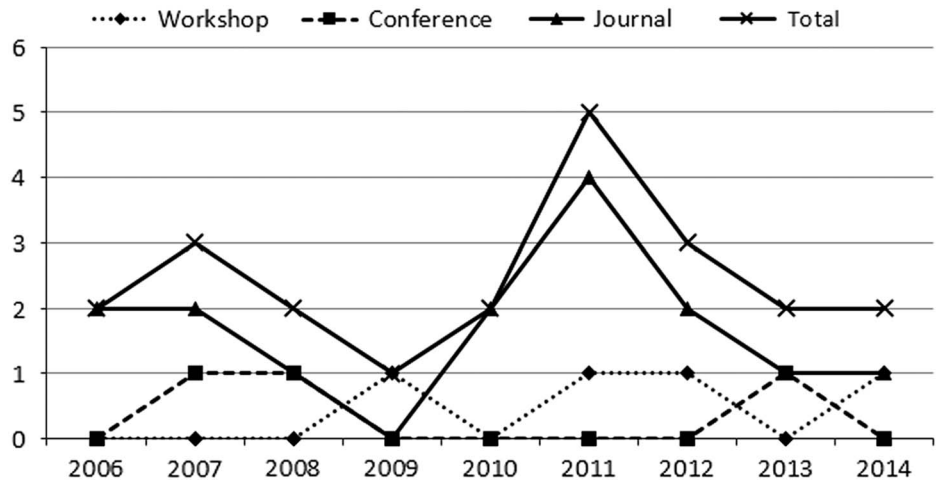

Figure 4 Paper publication types and distribution over time

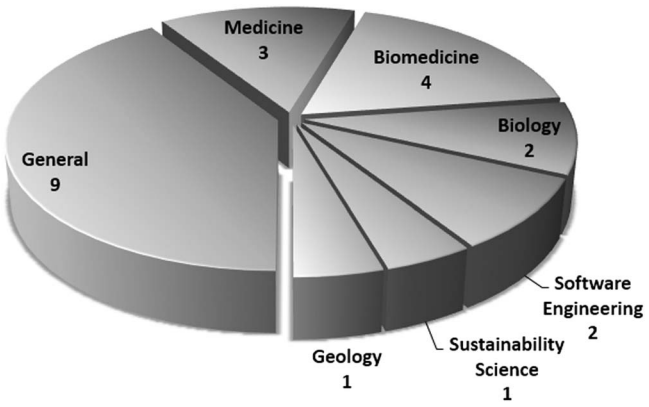

Figure 5 Paper distribution over domains

(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 4 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.

Considering the paper distribution in the scientific domains (Figure 5), 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.

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 & 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) regarding the drugs, allow the determination of interactions and the belief levels associated to their real existence. 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 domain concepts and is designed in such a way that it must be constantly extended and revised in order to accommodate new scientific results of different Software Engineering domains. 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 focussing 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. 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, an 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.

In another kind of proposals knowledge models are used 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 analysing 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. In another proposal, 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; Clare *et al.*, 2011; Marcondes, 2011; de Waard & schneider, 2012; Kuhn *et al.*, 2013) which models scientific results as assertions described using triples of the type <antecedent> <relationship> <consequent> (also described as concept-relationship-concept and subject-predicate-object). The other group (Mancini & Buckingham Shum, 2006; Groza *et al.*, 2007; Ciccarese *et al.*, 2008; de Waard *et al.*, 2009), on the other hand, is more focussed 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.

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

3.4.1 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 & 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 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 & Gahegan, 2007; Brodaric *et al.*, 2008; Groth *et al.*, 2010; Sharma *et al.*, 2010; Clare *et al.*, 2011; Croft *et al.*, 2011; Kraines & Guo, 2011; Marcondes, 2011; Russ *et al.*, 2011; Hunter & Williams, 2012; Santos & Travassos, 2013; van Valkenhoef *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.

3.4.2 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'[4]. Among the triple-based formalizations, 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; Clare *et al.*, 2011; Kraines & Guo, 2011; Marcondes, 2011; de Waard *et al.*, 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 & 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. calledAtomic, 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 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 & Buckingham Shum, 2006; Pike & Gahegan, 2007; Santos & Travassos, 2013) which, although 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) an Unified Modelling Language (UML) based representation is used to model theories concerned with evidence generated in primary studies. The other papers (Mancini & Buckingham Shum, 2006, Pike & Gahegan, 2007), based on specific 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 Cognitive Coherence Relations (Sanders *et al.*, 1993).

The remainder works propose 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 are used. As regards the aggregation of quantitative data originated from controlled experiments we can cite Hunter and Williams (2012) and van Valkenhoef *et al.* (2013). To that end, the Hunter and Williams (2012) work 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 of 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). With 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 focussing 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 Knowledge Engineering from Experimental Design that 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 upon the data. In the paper example, logical rules (e.g. *part of* and *overlap*) are defined to map cerebral regions.

[4] 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.

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 & 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 these 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 selected 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. There are also discussions about whether the triple-based representation should be widely adopted for all Science domains (Slater *et al*., 2008). 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.

3.4.3 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 & Williams, 2012; Santos & Travassos, 2013) mentioned that this aspect was not being addressed at that time or, 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.

Still, several papers discussed this theme thoroughly including some of which reserved whole sections to the theme. In this group, it is possible to identify two strategies: based on data model and based on the semantic inference. The searches based on data model only explore the syntactic structure in 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 informations 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 in making 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 & Buckingham Shum, 2006; Pike & Gahegan, 2007; Sharma *et al*., 2010; Kraines & Guo, 2011; de Waard & 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 over 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 with 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.

3.4.4 Aggregation of scientific results

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 which 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 1 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

Table 1 Aggregation strategies supported by proposals

Integrative	Boyce *et al*. (2007), Russ *et al*. (2011), Hunter and Williiams (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)

Table 2 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)

UML = Unified Modelling Language.

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-focussed 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 1 are those that explicitly discussed the aggregation possibilities.

3.4.5 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 focussed 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 are included as there are many tools to visually manipulate these representations. Still, to maintain the analysis consistency, we have chosen to consider only the papers which developed or explicitly cited forms of graphical representations.

Among the considered proposals (Table 2), 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 also a particular kind of graph.

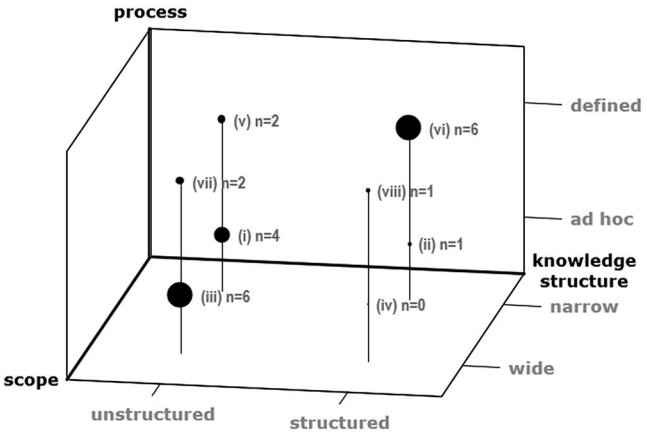

Figure 6 Paper classification in three dimensions: the definition of modelling process for representing scientific knowledge, and its mapping to scientific disciplines' knowledge structure and scope. *n* is the number of papers in each of the classification eight points[5]

3.4.6 Scientific knowledge modelling 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. Still, despite the importance of a such 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 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 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 6, 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 focussed 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

[5] 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 & schneider (2012), Santos & Travassos (2013); (vi) Boyce *et al*. (2007), Croft *et al*. (2011), Hunter and Williams (2012), van Valkenhoef *et al*. (2013); (vii) Mancini & Buckingham Shum (2006), Kuhn *et al*. (2013); (viii) Russ *et al*. (2011).

representation utilization is associated only to the understanding of its syntactical and semantic description).

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 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 5), have a narrow scope of application.

To better illustrate this classification, we present our reasoning for the narrow scope and structured knowledge' (items (ii) and (vi)) classification with two examples. The first example is the modelling of drug–drug interactions evidence from Boyce *et al*. (2007). In this work, the 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. Together these elements form the necessary definitions 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 for 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).

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 focusses 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 that should be considered in the design and construction of the knowledge model and computational infrastructure—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 knowledge models, 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[6] for the design of the knowledge representation and knowledge bases available in KE as a modelling process paradigm. Just as an example,

[6] It is important to recall 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 two main modelling approaches according to Studer *et al*. (1998) are the role-limiting methods (Mcdermott, 1988) and generic tasks (Bylander & 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, KE 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 favoured the rise of languages for the specification of knowledge-based systems (e.g. 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.

The lack of a better organization of the operational level for SKE could be seen in the reviewed papers, 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; Dibble & Bostrom, 1987; Fellers, 1987; Rook & Croghan, 1989; Motta *et al*., 1990; Plant, 1991; Studer *et al*., 1998; Schreiber, 2000). It should be noticed 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.

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

It is possible to identify three main phases in SKE projects (Figure 7; Table 3): 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

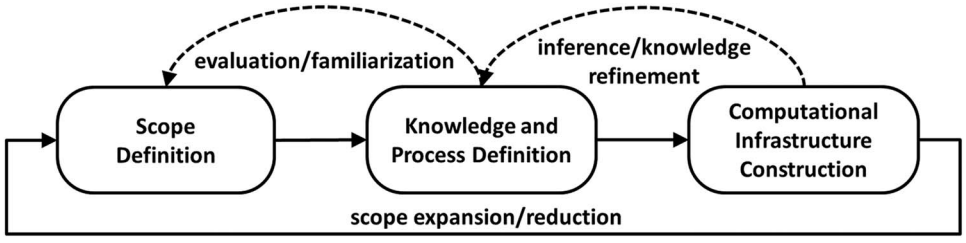

Figure 7 Main Scientific Knowledge Engineering phases

Table 3 Resume of Scientific Knowledge Engineering steps and their mapping to Knowledge Engineering (KE) process proposals

Steps	Description	Mapping to KE papers referenced at bottom
1. Familiarization	The goal is to become acquainted with how scientific knowledge is presented and what are its content elements. Scientific papers, research methods descriptions and discussions about technology transfer are examples of sources for this initial familiarization	1, 2, 3, 4, 5, 6, and 8
2. Knowledge initial organization	To be able to deal with huge amount of new information about the domain, it is important to start seeking regularities in the way scientific results are presented. This usually is found in the scientific discourse, epistemological views or major theories of the area	1, 2, 3, 5, 6, and 8
3. Domain conceptualization	Based on the on organization of the previous step, the goal is to develop an abstract model of the problem describing its main concepts and their relations	2, 4, 5, 6, 7, and 8
4. Knowledge representation definition	Definition of computational concrete formalization representing the domain knowledge. This determines how knowledge is instantiated and maintained within a computerized infrastructure	1, 2, 3, 4, 5, 6, 7, and 8
5. Knowledge utilization process definition	Complement the previous step with orientations regarding the process involved in translating scientific results into an instance of the defined knowledge representation	1, and 5
6. Inference strategy specification	Limited by the knowledge representation defined, the goal here is to determine inferences that will support scientists and practitioners in exploiting the scientific knowledge	1, 3, 4, 7, and 8
7. Computerized infrastructure implementation	It regards to the software engineering concerns involved in developing a computerized infrastructure	3, 4, 5, and 8

(1) Freiling *et al*. (1985) (2) Fellers (1987) (3) Dibble and Bostrom (1987) (4) Rook and Croghan (1989) (5) Motta *et al*. (1990) (6) Plant (1991) (7) Studer *et al*. (1998) (8) Schreiber (2000).

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.

It is interesting to notice in Table 3 that most KE discussions do not address the process for using the knowledge representation. As will be discussed in this section, the knowledge utilization process is particularly important in SKE since scientific results require special attention in how it can be translated to a knowledge representation. As a matter of comparison, as might be expected all eight KE works address the knowledge representation definition.

The roles involved in these stages are the domain specialist the knowledge engineer, and the software engineer. The first two roles were mentioned previously and are frequently cited in the technical literature (Fellers, 1987; Studer *et al*., 1998), whereas the third is seldom mentioned (Rook & 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 that is target of 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 2(a)). 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 focusses 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 modelling the scientific results into the knowledge representation—which can be done by researchers.

4.1.1 Scope definition phase

The first stage in the process has a more exploratory nature and aims at determining 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 other words this means it would be quite unusual that 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. Therefore, scientific papers can be used as the source for specialist knowledge—still, this does not exclude the possibility to use domain specialists (i.e. researchers) as source of knowledge as well. By using these sources, the objective is to become familiarized 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 that allows the knowledge engineer to grasp how the results of scientific investigations are described and what their main features are. As the universe of scientific papers can be 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 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 exploring semantic and conceptual relationships among the observed entities or on investigating phenomena 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 is 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, which are 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.

4.1.2 Knowledge and process definition stage

The second stage's main outcomes are the formalization of the knowledge representation and the 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 adopted research methods 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 to conduct some kind of validation of concepts and relationships structures commonly found 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 shared and external representations in Fellers (1987), micro- and

macro-knowledge in Rook and Croghan (1989), and primary representations and domain specifications in Plant (1991). As a result of this incremental specification and given the relative lower commitment of this step compared with 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 the 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 inference alternatives, 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 rigour used them the more direct the representation definition using a specific representation scheme will be. For instance, the use of a knowledge acquisition grammar can help specifying 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 & DeKleer, 1993).

At this point, as a strategy to select among these alternatives for knowledge definition, it should be considered the possibility of their evaluation using real scientific results extracted from research papers. In fact, most papers found in the literature review used real data to present proof of concept for 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 should be kept in mind if the knowledge engineer 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 a 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.
- Model instantiation: the instantiation process should assist the researcher in identifying the constructs, hypothesis, analogies, models commonly used in the domain and in mapping them to defined knowledge representation. For instance, the indication of how to start from raw text and identify the concepts using text codification can help in this regard. 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 the 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 since 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 consequence, it is possible to suggest some stop 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 representation semantic features 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).

4.1.3 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 & 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 defined scope and semantic richness of the knowledge representation. 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. Several were cited in the literature review presented in this text and were also examined by Hunter and Liu (2010). Amongst them, we can mention deduction and abduction strategies used to produce or abduct (assume) new knowledge from the knowledge base, integrity verification to identify possible disagreement/consensus in the available knowledge, and knowledge aggregation based on the combination is obtained from the reduction of conflicts and redundancies identified through 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 indeed 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, it is also necessary 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. This is consistent with previous steps, which consider these concerns at different points (steps 4 and 6). Therefore, the separation of knowledge formalization and inference strategy concerns is manifested both in process and implementation levels. On the process level the idea of defining two different steps is 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 & 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 requirements, or be initiated as a result of new iterations of the SKE process as shown in Figure 6.

5 Real case example: decisions and experiences

As referenced before (Santos & Travassos, 2013), the authors of this paper have proposed a knowledge representation for evidence in Software Engineering. Thus, one of the authors' particular motivation for

this present work was a search for guidelines to inform our needs in terms of how we could model a knowledge representation and how we could design a computational infrastructure for evidence modelling and aggregation in Software Engineering (Santos *et al*., 2015). The construction of the computational infrastructure and this present work were developed simultaneously. Therefore, it is not possible to achieve any reasonable level of neutrality between the account given here and the step-by-step approach described on the previous section. In fact, many of the considerations described in the approach were certainly influenced exactly by the experiences we had. Thus, the goal here is only to offer some practical guidance on scientific KE project steps and to supplement the justification for its need.

In this section, we briefly describe our own experiences in building a computational infrastructure for research synthesis in Software Engineering using the perspective of the step-by-step approach presented in the previous section.

Familiarization (step 1)

This step was hugely facilitated by the authors' experience and expertise in both Software Engineering and research methods as researchers in the Experimental Software Engineering area. For this reason, we did not need to collect and read a representative set of papers to find out how scientific investigation is described in the area and what its main characteristics are. Since the beginning, given the contradictory focus on meta-analysis research synthesis and a high heterogeneity of evidence types (i.e. qualitative and quantitative) in the area, our main goal was to develop an instrument that could represent all evidence types on the same perspective so that they could be aggregated. A secondary, but also important objective, was that this instrument could be relatively easily used by both scientists and practitioners. 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 type of study is common in Software Engineering and also that it is simpler to structure as it was expected that fewer knowledge formalization constructs would be needed to relate causal concepts in comparison to more complex and detailed knowledge from specific Software Engineering domains.

Knowledge initial organization (step 2)

Since our scope entailed 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 were described in study reports affected the development of the Software Engineering body of knowledge (e.g. Rainer *et al*., 2005; Shull *et al*., 2006; Dyba *et al*., 2007). This review 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, the need became apparent for a more structured representation for this knowledge 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 Goertz and Mahoney (2012).

Domain conceptualization (step 3) and knowledge representation definition (step 4)

We describe steps 3 and 4 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 Software Engineering scientific knowledge, especially from an epistemological standpoint. The idea was that even if papers focussed 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 software practices used in an organization. And Sjøberg

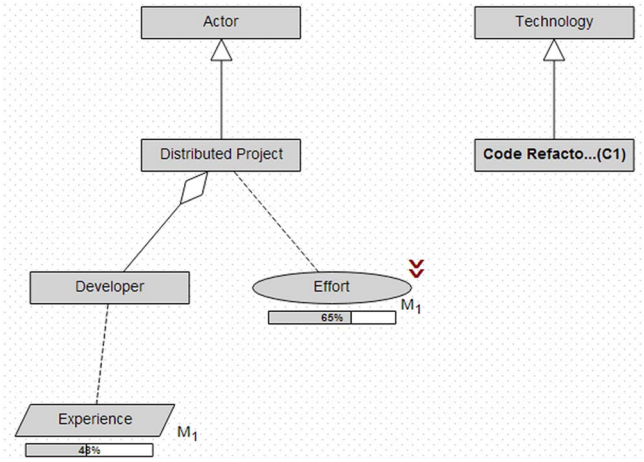

Figure 8 Knowledge representation notation—fragment with two archetypes for an evidence presented in Santos and Travassos (2013)

et al. (2008) propose a framework for describing Software Engineering theories. 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 establish 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 the proposed framework for theories, the authors defined a notation (partly based on UML) to graphically describe theories. The framework was based on the domain conceptualization that a typical Software Engineering situation is given by an *actor* applying *technologies* to perform certain *activities* on a *software system*.

Thus, this UML lineage was almost sufficient from the representational needs perspective. The notation defines all possible relationships (theory propositions) between concepts (theory constructs), and determines that each theory should have its constructs derived from four archetypes (actor, technology, activity, and software system). Five relationships are described: *type of*, *part of*, *property of*, *cause of* and *moderation of*; and two concepts are distinguished: *value concepts* used for characterizing contextual information, and *variable concepts* representing moderators or effects that assume alternative values ('more' or 'less' for a given property) in each instance.

Figure 8 presents an example of the adopted notation. Value constructs are represented with rectangles and variable constructs are represented with ellipses and parallelogram for effects and moderators, respectively. Solid lines are used for *type of* (with an arrow in the generic end) and *part of* (with a diamond in the whole end) relations. Dashed lines are used for *property of* elements. *Cause of* and *moderation of* are implicit in the diagram notation. There is no line linking these elements. Cause–effect relations are determined by reading the causal value constructs (in this case, Code Refactoring—C1 means 'cause 1', as it can be more than one cause) and effects (in this case, the Effort variable). Moderation relations can be identified from the textual hints beside effects and moderators (as M_1 next to Effort and Experience). In the figure, it should be read as 'Experience moderates the Effort effect of Code Re-factoring'. Finally, the bars under the variable constructs indicate the belief value for each effect or moderator (discussed on step 6).

Although not meant to formalize Software Engineering evidence, we could identify almost all the elements needed to translate evidence knowledge into the structured theory notation. Besides, it was a straightforward process to map the cause–effect notion into the cause–effect and moderation relationship types of the notation. Thus, for an initial iteration, we were convinced that the notation would be applicable to most of the situations that we could conceive. We just had to add a minor extension that was not defined in Sjøberg *et al*. (2008), which was an ordinal seven-point Likert scale for the cause–effect relationships from strongly negative to strongly positive (in Figure 8, the two facing down arrows on the top right of the

Effort construct means 'negative'—a value between strongly negative and weakly negative). At that point, after all the analyses and decisions, and to get a real grasp of the applicability of the theory notation for evidence representation, we carried out a proof of concept showing how software inspection studies could be translated to the representation. This resulted in the work of Santos and Travassos (2013).

After this preliminary evaluation based on the representation conceptual definitions, we built a formal model describing it. The evidence meta-model, defining the representation abstract syntax, was formalized using Eclipse Modeling Framework (www.eclipse.org/emf). For details describing the meta-model we refer to Santos *et al.* (2015).

Knowledge utilization process definition (step 5)

Once again we followed our initial determination of backing our definitions and decisions on epistemological foundations. Given our primary goal of enabling the aggregation of evidence to have a pooled result from which new hypotheses or practical decisions could be made, we concluded 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 the case, for instance, in thematic synthesis with text codes and cognitive maps, in meta-ethnography with translations and even in meta-analysis with effect sizes used to estimate and compare the order of magnitude of the outcomes of quantitative studies—a review of research synthesis methods can be found in Dixon-Woods *et al.* (2005).

The process we have defined for knowledge representation combines procedures and techniques from five research synthesis methods: *thematic synthesis*, *meta-ethnography*, *case survey*, *qualitative comparative analysis* and *theory building with meta-analysis*. The major orientation in translating evidence to the graphical theory notation comes from the thematic synthesis and its increasing abstraction levels (text, codes, concepts and relations, and evidence representation). Recommendations from meta-ethnography include how the text should be coded and papers translated from one to another, to identify concepts and relations. The inductive approach from qualitative comparative analysis, where concepts are identified inductively from the collection of studies, supplements these recommendations. To improve synthesis reliability, the participation of more than one researcher is recommended as is in case survey and many other qualitative methods. At last, instructions for identifying cause–effect relationships are also included, as this is what puts qualitative and quantitative evidence in the same perspective. The instructions are based on the differentiation from Goertz and Mahoney (2012): 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'.

Inference strategy specification (step 6)

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 inspection techniques (*ad hoc* and checklist). But if we say that both techniques are *type of* inspection, then we can represent both evidence under the same representation, generalizing software methodology (technology archetype) as inspections. When representing the results from more than one study using one instance of the graphical representation we say that the evidence was *aggregated*. We use description logics to detect if they are *compatible*.

Determining the compatibility of evidence was only one part of the aggregation problem. Once the *value constructs* associated with the description of contextual aspects are made compatible, it is necessary to analyze the relations associated with the *variable* aspects, that is, the cause–effect and moderation relations. To do that we extended the knowledge representation with uncertainty formalisms to represent the strength on the observed outcome for each study. Using an uncertainty formalism, it is possible to increase confidence on the outcome when the evidence converges or identify the possibly weaker resultant when it diverges. The mathematical theory of evidence was used as the uncertainty formalism. With its

Dempster's Rule of Combination (Shafer, 1976), the framework takes two pieces of evidence and produces new evidence representing the consensus of the two original pieces. To that end, each piece of evidence is expressed in terms of belief values—using the *basic probability assignment function*—assigned to subsets of propositions of distinct, exhaustive possibilities—called the *frame of discernment*. The connection between the UML-based notation from step 4 and the mathematical theory of evidence materialize in defining the cause–effect seven-point Likert scale as the *frame of discernment*.

As a last note, this step exemplifies why iterativity is expected in the process shown in Figure 7. At this stage, as a result of the inference specification we had to extend the knowledge representation with uncertainty formalisms. Moreover, we added quality evaluation and study type identification on the studies translation on step 5 as a mean to provide an estimate for belief value. This was based on the GRADE working group evidence classification (grading quality of evidence and strength of recommendations; Atkins *et al.*, 2004).

Computational infrastructure implementation (step 7)

We executed usual software engineering activities to specify, design, construct, and test the computational infrastructure. The specification started with a conceptual organization of the main elements expected to form the infrastructure created as a specialization of Figure 1 organization. Five high-level aspects were identified in this conceptual organization: information visualization and editing, facilities for Academia, facilities for industry, search mechanisms, and inference and knowledge base aspects. These conceptual boundaries facilitated the identification of functional and non-functional requirements, and guided subsequent stages, especially in architecture design. For instance, requirements associated with the facilities for Academia were mainly distilled from step 5, but its boundary with the search mechanisms was important to to design the interface navigational aspects. Furthermore, the search mechanisms directly influenced the design of the knowledge base as graph database[7]. The object oriented paradigm was used to model the knowledge representation defined in step 4, which also gave flexibility to link it with the uncertainty formalism described in step 6. Following the directions given on step 7 from Section 4.1.3, the inference engine—mainly the mathematical theory of evidence implementation—was implemented as a separate component.

In association with the information visualization and editing aspects, we identified many non-functional requirements related to usability and visualization. Based on Moody (2009), a complete reformulation of the original visual notation syntax was implemented, to try and improve the cognitive effectiveness of how easily and accurately the diagram could be processed. Following his recommendations, major changes include the absence of explicit cause–effect and moderation connections to reduce diagram complexity and to achieve a better use of retinal variables (e.g. visual shapes) to improve the perceptual discrimination. The representation in Figure 8 already uses this new notation and is an actual rendering from the computational infrastructure. The original representation from Sjøberg *et al*. (2008) for the same evidence can be seen in Santos and Travassos (2013).

For detailed information regarding the computational infrastructure implementation we refer to Santos *et al*. (2015). An example of the research synthesis method supported by the computational infrastructure can be obtained in Martinez-Fernandez *et al*. (2015).

6 Final considerations

In this paper, SKE was conceptualized as a research area concerned with the design and construction of infrastructures for the computational manipulation of scientific knowledge. Based on the KE 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

[7] Neo4j database: `http://www.neo4j.org/`

researchers to represent their inquiry results. In total, 22 papers were found over 8 years, showing a solid 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 focussed on the representational aspects of their formalizations without giving proper justification as to why they were chosen or what the design decisions were 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.

Still, only the initial view on the theme was sought to be established in this paper. There are several opportunities to unfold the perspective introduced here which can also be linked to some limitations of this work. The main limitation is related to the 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 paper 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.

Another topic that can be further explored, is the combination of SKE with other approaches discussed in Section 2. And a third line of investigation is a better characterization of the modelling process as performed by the researchers and the different properties of scientific knowledge, analysing their associations with the necessary facilities that computational infrastructures should offer. We believe that there are two main issues to be further investigated in this regard. One is relatively independent of domain, but tied to epistemology, which is mapping the epistemological elements, as found in the different scientific methodologies (e.g. theories, concepts, facts, statements)—a work which has already been started by Hars (2001). The other is to try to identify conceptual regularities in each domain (e.g. biological pathway) to further expand knowledge representation in terms of expressiveness and reasoning. Cognitive Science and its theories on how scientists think about problems (e.g. induction, deduction, and abduction) can also help in this regard, and is to some extent what Computational Discovery of Scientific Knowledge tried to implement—even though with a much greater ambition to automate most of the reasoning in the scientific process.

Acknowledgements

This research is supported by CNPq (Brazilian Research Council) under the grant 305929/2014-3. Prof. Travassos is a CNPq researcher.

References

Atkins, D., Best, D., Briss, P. A., Eccles, M., Falck-Ytter, Y., Flottorp, S., Guyatt, G. H., Harbour, R. T., Haugh, M. C., Henry, D., Hill, S., Jaeschke, R., Leng, G., Liberati, A., Magrini, N., Mason, J., Middleton, P., Mrukowicz, J., O'Connell, D., Oxman, A. D., Phillips, B., Schünemann, H. J., Edejer, T. T.-T., Varonen, H., Vist, G. E., Williams, J. W. Jr & Zaza, S., GRADE Working Group 2004. Grading quality of evidence and strength of recommendations. *BMJ* **328**(7454), 1490.

Bairoch, A. 2009. The future of annotation/biocuration, *Nature Precedings*.

Barga, R. & Gannon, D. 2007. Scientific versus business workows. In *Workows for e-Science*, Taylor I. J., Deelman E., Gannon D. B. & Shields M. (eds). Springer, 9–16.

Bauer-Mehren, A., Furlong, L. I. & Sanz, F. 2009. Pathway databases and tools for their exploitation: benefits, current limitations and challenges. *Molecular Systems Biology* **5**(290).

Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D., Owen, S., Newman, D., Sufi, S. & Goble, C. 2013. Why linked data is not enough for scientists. *Future Generation Computer Systems* **29**(2), 599–611.

Biolchini, J., Mian, P., Natali, A. & Travassos, G. H. 2005. *Systematic review in software engineering*. Technical report No. RT-ES 679/05, Federal University of Rio de Janeiro (UFRJ/COPPE).

Booth, A. 2011. Evidence-based practice: triumph of style over substance? *Health Information & Libraries Journal* **28**(3), 237–241.

Budgen, D., Turner, M., Brereton, P. & Kitchenham, B. 2008. Using mapping studies in software engineering. *In* Proceedings of PPIG Psychology of Programming Interest Group, 195–204. Lancaster University.

Bunge, M. 2004. How does it work? The search for explanatory mechanisms. *Philosophy of the Social Sciences* **34**(2), 182–210.

Bylander, T. & Chandrasekaran, B. 1987. Generic tasks for knowledge-based reasoning: the 'right' level of abstraction for knowledge acquisition. *International Journal of Man-Machine Studies* **26**(2), 231–243.

Callahan, A., Dumontier, M. & Shah, N. H. 2011. HyQue: evaluating hypotheses using semantic web technologies. *Journal of Biomedical Semantics* **2**(2), 1–17.

Chua, C. E. H., Storey, V. C. & Chiang, R. H. 2012. Deriving knowledge representation guidelines by analyzing knowledge engineer behavior. *Decision Support Systems* **54**(1), 304–315.

Cohen, A. M. & Hersh, W. R. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics* **6**(1), 57–71.

Cooper, H. M., Hedges, L. V. & Valentine, J. C. 2009. *The Handbook of Research Synthesis and Meta-Analysis*. Russell Sage Foundation.

Craver, C. F. & Darden, L. 2005. Introduction. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences* **36**(2), 233–244.

da Cruz, S., Campos, M. & Mattoso, M. 2009. Towards a taxonomy of provenance in scientific workow management systems. In *2009 World Conference on Services – I*, 259–266.

Deelman, E., Gannon, D., Shields, M. & Taylor, I. 2009. Workows and e-science: an overview of workow system features and capabilities. *Future Generation Computer Systems* **25**(5), 528–540.

Dennis, C. 2002. Biology databases: information overload. *Nature* **417**(6884), 14.

Dibble, D. & Bostrom, R. P. 1987. Managing expert systems projects: factors critical for successful implementation. In *Proceedings of the Conference on the 1987 ACM SIGBDP-SIGCPR Conference*, *SIGCPR' 87*, 96–128. ACM.

Dinakarpandian, D., Lee, Y., Vishwanath, K. & Lingambhotla, R. 2006. MachineProse: an ontological framework for scientific assertions. *Journal of the American Medical Informatics Association* **13**(2), 220–232.

Dixon-Woods, M., Agarwal, S., Jones, D., Young, B. & Sutton, A. 2005. Synthesising qualitative and quantitative evidence: a review of possible methods. *Journal of Health Services Research & Policy* **10**(1), 45–53.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* **77**(2), 321–357.

Dyba, T., Dingsoyr, T. & Hanssen, G. 2007. Applying systematic reviews to diverse study types: an experience report. In International Symposium on Empirical Software Engineering and Measurement, 225–234.

Džeroski, S., Langley, P. & Todorovski, L. 2007. Computational discovery of scientific knowledge. In *Computational Discovery of Scientific Knowledge*, Džeroski S. & Todorovski L. (eds), Lecture Notes in Computer Science **4660**, 1–14. Springer.

Easterbrook, S., Singer, J., Storey, M.-A. & Damian, D. 2008. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*, Shull F., Singer J. & Sjøberg D. I. K. (eds). Springer, 285–311.

Eriksson, H. 1992. A survey of knowledge acquisition techniques and tools and their relationship to software engineering. *Journal of Systems and Software* **19**(1), 97–107.

Fayyad, U. & Stolorz, P. 1997. Data mining and KDD: promise and challenges. *Future Generation Computer Systems* **13**(2–3), 99–115.

Fellers, J. 1987. Key factors in knowledge acquisition. *SIGCPR Computer Personnel* **11**(1), 10–24.

Fiore, S. & Aloisio, G. 2011. Special section: data management for eScience. *Future Generation Computer Systems* **27**(3), 290–291.

Forbus, K. D. & DeKleer, J. 1993. *Building Problem Solvers*. MIT Press.

Ford, K. M. 1993. *Knowledge Acquisition as Modeling*. Wiley.

Freiling, M., Alexande, J., Messick, S., Rehfuss, S. & Shulman, S. 1985. Starting a knowledge engineering project: a step-by-step approach. *AI Magazine* **6**(3), 150.

Goertz, G. & Mahoney, J. 2012. *A Tale of Two Cultures: Qualitative and Quantitative Research in the Social Sciences*. Princeton University Press.

Hars, A. 2001. Designing scientific knowledge infrastructures: the contribution of epistemology. *Information Systems Frontiers* **3**(1), 63–73.

Hey, T. & Trefethen, A. 2003. The data deluge: an e-science perspective. In *Grid Computing*, Berman F., Fox G. & Hey T. (eds). John Wiley & Sons Ltd, 809–824.

Hunter, A. & Liu, W. 2010. A survey of formalisms for representing and reasoning with scientific knowledge. *The Knowledge Engineering Review* **25**(2), 199–222.

Hunter, J. 2008. Scientific publication packages—a selective approach to the communication and archival of scientific output. *International Journal of Digital Curation* **1**(1), 33–52.

Ivarsson, M. & Gorschek, T. 2012. Tool support for disseminating and improving development practices. *Software Quality Journal* **20**(1), 173–199.

Khatri, P., Sirota, M. & Butte, A. J. 2012. Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Computational Biology* **8**(2), e1002375.

Kiritchenko, S., Bruijn, B. D., Carini, S., Martin, J. & Sim, I. 2010. ExaCT: automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making* **10**(1), 56.

Kitchenham, B. & Charters, S. 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical Report No. EBSE 2007-001, Keele University and Durham University Joint Report.

Langley, P. 1987. *Scientific Discovery: Computational Explorations of the Creative Processes*. MIT Press.

Langley, P., Zytkow, J. M., Bradshaw, G. L. & Simon, H. A. 1983. Three facets of scientific discovery. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence – Volume 1, IJCAI'83*, 465–468. Morgan Kaufmann Publishers, Inc.

Lenat, D. B. & Feigenbaum, E. A. 1991. On the thresholds of knowledge. *Artificial Intelligence* **47**(1–3), 185–250.

Lewis-Beck, M., Bryman, A. & Liao, T. F. 2004. *Encyclopedia of Social Science Research Methods*. SAGE Publications, Inc.

Lin, C., Lu, S., Fei, X., Chebotko, A., Pai, D., Lai, Z., Fotouhi, F. & Hua, J. 2009. A reference architecture for scientific workflow management systems and the VIEW SOA solution. *IEEE Transactions on Services Computing* **2**(1), 79–92.

Lord, P., Macdonald, A., Lyon, L. & Giaretta, D. 2004. From data deluge to data curation, In *Proceeding of the 3th UK e-Science All Hands Meeting*, 371–375.

Maccagnan, A., Riva, M., Feltrin, E., Simionati, B., Vardanega, T., Valle, G. & Cannata, N. 2010. Combining ontologies and workflows to design formal protocols for biological laboratories. *Automated Experimentation* **2**(1), 1–14.

Martinez-Fernandez, S., Santos, P., Ayala, C., Franch, X. & Travassos, G. 2015. Aggregating Empirical Evidence about the Benefits and Drawbacks of Software Reference Architectures, 2015 ACM/IEEE. *International Symposium, on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–10.

Mcdermott, J. 1988. Preliminary steps toward a taxonomy of problem-solving methods. In *Automating Knowledge Acquisition for Expert Systems, number 57 in The Kluwer International Series in Engineering and Computer Science*, Marcus S. (ed.). Springer, 225–256.

Mons, B. 2005. Which gene did you mean? *BMC Bioinformatics* **6**(1), 142.

Mons, B. & Velterop, J. 2009. Nano-publication in the e-science era. In *Workshop on Semantic Web Applications in Scientific Discourse*.

Moody, D. 2009. The 'physics' of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering* **35**(6), 756–779.

Motta, E., Rajan, T. & Eisenstadt, M. 1990. Knowledge acquisition as a process of model refinement. *Knowledge Acquisition* **2**(1), 21–49.

Newman, H. B., Ellisman, M. H. & Orcutt, J. A. 2003. Data-intensive e-science frontier research. *Communication of the ACM* **46**(11), 68–77.

Noblit, G. W. & Hare, R. D. 1988. *Meta-Ethnography: Synthesizing Qualitative Studies*. SAGE.

Novère, N. L., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M. I., Wimalaratne, S. M., Bergman, F. T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villéger, A., Boyd, S. E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T. C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D. B., Sander, C., Sauro, H., Snoep, J. L., Kohn, K. & Kitano, H. 2009. The systems biology graphical notation. *Nature Biotechnology* **27**(8), 735–741.

Petersen, K., Feldt, R., Mujtaba, S. & Mattsson, M. 2008. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE'08*, 68–77. British Computer Society.

Plant, R. T. 1991. Rigorous approach to the development of knowledge-based systems. *Knowledge-Based Systems* **4**(4), 186–196.

Rainer, A., Jagielska, D. & Hall, T. 2005. Software engineering practice versus evidence-based software engineering research. In *Proceedings of the 2005 Workshop on Realising Evidence-Based Software Engineering*, *REBSE'05*, 1–5. ACM.

Rook, F. & Croghan, J. 1989. The knowledge acquisition activity matrix: a systems engineering conceptual framework. *IEEE Transactions on Systems, Man and Cybernetics* **19**(3), 586–597.

Rzhetsky, A., Iossifov, I., Koike, T., Krauthammer, M., Kra, P., Morris, M., Yu, H., Duboué, P. A., Weng, W., Wilbur, W. J., Hatzivassiloglou, V. & Friedman, C. 2004. GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *Journal of Biomedical Informatics* **37**(1), 43–53.

Sackett, D. L., Rosenberg, W. M., Gray, J. A., Haynes, R. B. & Richardson, W. S. 1996. Evidence based medicine: what it is and what it isn't. *BMJ* **312**(7023), 71–72.

Sanders, T. J. M., Spooren, W. P. M. & Noordman, L. G. M. 1993. Coherence relations in a cognitive theory of discourse representation. *Cognitive Linguistics* **4**(2), 93–134.

Santos, P. & Travassos, G. 2013. On the representation and aggregation of evidence in software engineering: a theory and belief-based perspective. *Electronic Notes in Theoretical Computer Science* **292**, 95–118.

Santos, P. & Travassos, G. 2015. Aggregating empirical evidence about the benefits and drawbacks of software reference architectures. In International Symposium on Empirical Software Engineering and Measurement (in press).

Santos, P. S., Nascimento, I. & Travassos, G. H. 2015. A computational infrastructure for research synthesis in software engineering. In *XVIII Ibero-American Conference on Software Engineering*, 309–322. URP, SPC, UCSP, UCSP.

Schreiber, G. 2000. *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press.

Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press.

Shotton, D. 2009. Semantic publishing: the coming revolution in scientific journal publishing. *Learned Publishing* **22**(2), 85–94.

Shrager, J. 1990. *Computational Models of Scientific Discovery and Theory Formation*. Morgan Kaufmann Publisher.

Shull, F., Feldmann, R. & Shaw, M. 2006. Building decision support in an imperfect world. In *International Symposium on Empirical Software Engineering ISESE*, 33–35.

Shull, F., Singer, J. & Sjøberg, D. I. K. 2007. *Guide to Advanced Empirical Software Engineering*, 2008 edition. Springer.

Simon, H. A. 1977. Scientific discovery and the psychology of problem solving. In *Models of Discovery, Number 54 in Boston Studies in the Philosophy of Science*, Simon H. A. (ed.). Springer, 286–303.

Sjøberg, D. I. K., Dybå, T., Anda, B. C. D. & Hannay, J. E. 2008. Building theories in software engineering. In *Guide to Advanced Empirical Software Engineering*, Shull F., Singer J. & Sjøberg D. I. K. (eds). Springer, 312–336.

Slater, T., Bouton, C. & Huang, E. S. 2008. Beyond data integration. *Drug Discovery Today* **13**(13–14), 584–589.

Stock, K., Robertson, A., Reitsma, F., Stojanovic, T., Bishr, M., Medyckyj-Scott, D. & Ortmann, J. 2009. eScience for sea science: a semantic scientific knowledge infrastructure for marine scientists. In *Fifth IEEE International Conference on e-Science. e-Science' 09*, 110–117.

Studer, R., Benjamins, V. & Fensel, D. 1998. Knowledge engineering: principles and methods. *Data & Knowledge Engineering* **25**(1–2), 161–197.

Travassos, G., Santos, P., Neto, P. & Biolchini, J. 2008. An environment to support large scale experimentation in software engineering. In *13th IEEE International Conference on Engineering of Complex Computer Systems, 2008. ICECCS 2008*, 193–202.

Valdés-Pérez, R. E. 1996. Computer science research on scientific discovery. *The Knowledge Engineering Review* **11**(1), 57–66.

Vorms, M. 2011. Representing with imaginary models: formats matter. *Studies in History and Philosophy of Science Part A* **42**(2), 287–295.

Wallace, D. & Fujii, R. 1989. Software verification and validation: an overview. *IEEE Software* **6**(3), 10–17.

Wielinga, B., Schreiber, A. & Breuker, J. 1992. KADS: a modelling approach to knowledge engineering. *Knowledge Acquisition* **4**(1), 5–53.

Literature Survey References

Bölling, C., Weidlich, M. & Holzhüutter, H.-G. 2014. SEE: structured representation of scientific evidence in the biomedical domain using semantic web techniques. *Journal of Biomedical Semantics* **5**(Suppl 1), S1.

Boyce, R., Collins, C., Horn, J. & Kalet, I. 2007. Modeling drug mechanism knowledge using evidence and truth maintenance. *IEEE Transactions on Information Technology in Biomedicine* **11**(4), 386–397.

Brodaric, B., Reitsma, F. & Qiang, Y. 2008. SKIing with DOLCE: toward an e-science knowledge infrastructure. In *Proceedings of the Fifth International Conference on Formal Ontology in Information Systems (FOIS 2008)*, 208–219. IOS Press.

Ciccarese, P., Wu, E., Wong, G., Ocana, M., Kinoshita, J., Ruttenberg, A. & Clark, T. 2008. The SWAN biomedical discourse ontology. *Journal of Biomedical Informatics* **41**(5), 739–751.

Clare, A., Croset, S., Grabmueller, C., Kafkas, S., Liakata, M., Oellrich, A. & Rebholz-Schuhmann, D. 2011. Exploring the generation and integration of publishable scientific facts using the concept of nano-publications. In *Workshop on Semantic Publishing at ESWC2011*, 13–17.

Croft, D., O'Kelly, G., Wu, G., Haw, R., Gillespie, M., Matthews, L., Caudy, M., Garapati, P., Gopinath, G., Jassal, B., Jupe, S., Kalatskaya, I., Mahajan, S., May, B., Ndegwa, N., Schmidt, E., Shamovsky, V., Yung, C., Birney, E., Hermjakob, H., D'Eustachio, P. & Stein, L. 2011. Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Research* **39**(Suppl 1), D691–D697.

de Waard, A., Buckingham Shum, S., Carusi, A., Park, J., Samwald, M. & Sándor, Á. 2009. Hypotheses, evidence and relationships: the HypER approach for representing scientific knowledge claims.

de Waard, A. & Schneider, J. 2012. Formalising uncertainty: an ontology of reasoning, certainty and attribution (ORCA). In *Workshop on Semantic Technologies Applied to Biomedical Informatics and Individualized Medicine (SATBI + SWIM)*..

Dinakarpandian, D., Lee, Y., Vishwanath, K., Lingambhotla, R. 2006. MachineProse: an ontological framework for scientific assertions. *Journal of the American Medical Informatics Association* **13**(2), 220–232.

Ekaputra, F., Sabou, M., Serral, E. & Biffl, S. 2014. Supporting information sharing for reuse and analysis of scientific research publication data. In *Proceedings of the 4th Workshop on Semantic Publishing*, *SePublica '14*.

Groth, P., Gibson, A. & Velterop, J. 2010. The anatomy of a nanopublication. *Information Services and Use* **30**(1), 51–56.

Groza, T., Möller, K., Handschuh, S., Trif, D. & Decker, S. 2007. SALT: weaving the claim web. In *The Semantic Web*, Aberer K., Choi K.-S., Noy N., Allemang D., Lee K.-I., Nixon L., Golbeck J., Mika P., Maynard D., Mizoguchi R., Schreiber G. & Cudré-Mauroux P. (eds), *Lecture Notes in Computer Science* **4825**, 197–210. Springer.

Hunter, A. & Williams, M. 2012. Aggregating evidence about the positive and negative effects of treatments. *Artificial Intelligence in Medicine* **56**(3), 173–190.

Kraines, S. & Guo, W. 2011. A system for ontology-based sharing of expert knowledge in sustainability science. *Data Science Journal* **9**, 107–123.

Kuhn, T., Barbano, P. E., Nagy, M. L. & Krauthammer, M. 2013. Broadening the scope of nanopublications. In *The Semantic Web: Semantics and Big Data*, Cimiano P., Corcho O., Presutti V., Hollink L. & Rudolph S. (eds), *Lecture Notes in Computer Science* **7882**, 487–501. Springer.

Mancini, C. & Buckingham Shum, S. J. 2006. Modelling discourse in contested domains: a semiotic and cognitive framework. *International Journal of Human-Computer Studies* **64**(11), 1154–1171.

Marcondes, C. H. 2011. Knowledge network of scientific claims derived from a semantic publication system. *Information Services and Use* **31**(3), 167–176.

Pike, W. & Gahegan, M. 2007. Beyond ontologies: toward situated representations of scientific knowledge. *International Journal of Human-Computer Studies* **65**(7), 674–688.

Russ, T. A., Ramakrishnan, C., Hovy, E. H., Bota, M. & Burns, G. A. 2011. Knowledge engineering tools for reasoning with scientific observations and interpretations: a neural connectivity use case. *BMC Bioinformatics* **12**(1), 351.

Santos, P. & Travassos, G. 2013. On the representation and aggregation of evidence in software engineering: a theory and belief-based perspective. *Electronic Notes in Theoretical Computer Science* **292**, 95–118.

Sharma, R., Poole, D. & Smyth, C. 2010. A framework for ontologically-grounded probabilistic matching. *International Journal of Approximate Reasoning* **51**(2), 240–262.

van Valkenhoef, G., Tervonen, T., Zwinkels, T., de Brock, B. & Hillege, H. 2013. ADDIS: a decision support system for evidence-based medicine. *Decision Support Systems* **55**(2), 459–475.