Title	Ontology learning for systems engineering body of knowledge
Author(s)	Yang, Lan
Publication Date	2020-09-14
Publisher	NUI Galway
Item record	http://hdl.handle.net/10379/16169

Ontology Learning for Systems Engineering Body of Knowledge

Lan Yang

BSc, MEng, MSc,

Thesis submitted for
the Degree of Doctor of Philosophy to
National University of Ireland, Galway

Supervisor: Dr. Kathryn Cormican
School of Engineering,
College of Science and Engineering,
National University of Ireland, Galway

Co-supervisor: Prof. Ming Yu
Department of Industrial Engineering,
Tsinghua University

September 2020

TABLE OF CONTENTS

TABLE OF CONTENTS

DECLARATION

I hereby certify that the work contained herein is original and entirely my own work, except as otherwise explicitly cited and acknowledged within the text. I have not obtained a degree in this University, or elsewhere, on the basis of this work.

Signature:

Lan Yang

Date:

10th September 2020

EXECUTIVE SUMMARY

Systems engineering (SE) is a multidisciplinary and integrative approach that enables the successful realization of engineered systems. It encompasses fundamentals, principles, and models of foundational systems science, and associated scientific, technological, and management methods for the entire system life cycle.

However, the SE body of knowledge is fragmented, as seen from the various guidelines, handbooks, and standards existing in this domain. The lack of a cohesive body of knowledge and shared conceptual framework hinders the mutual understanding of the nature of SE, the practical application of SE approaches, and the sustainable development of the SE discipline. Therefore, the establishment of a common knowledge representation for the entire SE body of knowledge and a shared SE ontology is urgently called for and strongly advocated.

This thesis presents a study on the development of a formal ontology for the entire SE body of knowledge using a novel and emerging ontology learning approach. The study was completed using a well-defined scientific process. First, a systematic literature review on relevant cognate studies was carried out to understand the state of the art of ontology development and its application in SE. The literature relating to ontology-based systems engineering (OBSE) was synthesized and analyzed. Then, based on the literature, the gaps and limitations were identified and used to define the research questions and goals. This analysis revealed that manual codification is used to develop SE ontologies, which is tedious, time-consuming, and error-prone. There is a clear need for a formal ontology that depicts the entire body of knowledge. Therefore, to address this gap, this research proposes an ontology learning methodology that takes advantage of natural language processing and machine learning techniques and makes use of existing SE standards to learn an SE ontology derived from available SE knowledge assets. In terms of the development of the SE ontology, three ontology models were developed to portray the conceptual, logical, and data facets. Regarding the validation of the research method, a comprehensive case study was conducted to apply the proposed ontology learning methodology along with the ontology models. From the case study, a formal ontology for the SE body of knowledge was obtained, with a controlled vocabulary of SE terminologies, a concept hierarchy with nine top-level classes, and relations between concepts. To further demonstrate the application scenarios of the ontology, the concepts and relations in the system life cycle processes were separately studied and used for restructuring the life cycle processes more robustly and dynamically. Finally, the thesis incorporates vital learnings and insights to help both academic researchers and practitioners implement a comprehensive and generalizable strategy to create SE ontologies for other application domains or use cases.

ACKNOWLEDGMENT

In 2020, when all the topics are inseparable from Covid-19, I am delighted that I can have a pure land, which is an area that Covid-19 cannot reach, my doctoral research thesis.

The last few months to complete the thesis have been difficult, and I have faced considerable disruption in my personal and academic life, and this has resulted in having to adapt to a lot of change. Such disruption has also demonstrated my resilience and flexibility from adapting to online learning to connecting with my supervisors, families, and friends online and possibly learning skills as well as taking up new activities.

In this context, I feel that I am grateful to my family, supervisors, and friends more than ever, who gave me a lot of help and care during my PhD.

First, I would like to thank my parents. Even though they are far away in China, they remember me all the time and give me support and the most selfless love. Second, I would like to thank my husband. He always accompanied me when I was under the most stress. Next, I would like to thank my parents-in-law, grandparents, aunts and uncles, and all the family members who care about me. They always give me encouragement and support, so that I have full confidence in the future.

Most sincerely, I would like to thank my supervisors. They gave me guidance on my academic path. They let me gradually change from an ignorant student to a scholar with her own independent thinking abilities.

Many thanks to my friends. This includes my colleagues, university teachers, best friends in life, and strangers who helped me. I am very grateful to have their company. They make my life colorful.

Especially, I would like to thank my reviewers for the attention given and the time taken to provide suggestions and corrections.

Finally, I would like to pay a special tribute to the healthcare and other workers, who are at the front line of this unprecedented worldwide emergency. In solidarity, I thank you for your profound service. It is greatly appreciated.

PUBLICATIONS IN DOCTORATE PERIOD

1. Yang, L., Cormican, K., & Yu, M. (2020). Ontology Learning for Systems Engineering Body of Knowledge. *IEEE Transactions on Industrial Informatics*, 1–9. https://doi.org/10.1109/TII.2020.2990953
2. Yang, L., Cormican, K., & Yu, M. (2019). Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry*, *111*, 148–171. https://doi.org/10.1016/j.compind.2019.05.003
3. Yang, L., Cormican, K., & Yu, M. (2019). Learning Systems Engineering Domain Ontologies from Text Documents. *2019 International Symposium on Systems Engineering (ISSE)*, 1–7. https://doi.org/10.1109/ISSE46696.2019.8984550
4. Manenti, G., Ebrahimiarjestan, M., Yang, L., & Yu, M. (2019). Functional Modelling and IDEF0 to Enhance and Support Process Tailoring in Systems Engineering. *2019 International Symposium on Systems Engineering (ISSE)*, 1–8. https://doi.org/10.1109/ISSE46696.2019.8984539
5. Yang, L., Cormican, K., & Yu, M. (2017). Towards a methodology for systems engineering ontology development - An ontology for system life cycle processes. *2017 IEEE International Systems Engineering Symposium (ISSE)*, 1–7. https://doi.org/10.1109/SysEng.2017.8088299
6. Yang, L., Cormican, K., & Yu, M. (2016). An Ontology Model for Systems Engineering Derived from ISO/IEC/IEEE 15288: 2015: Systems and Software Engineering-System Life Cycle Processes. *International Journal of Information, Control and Computer Sciences*, *11*(1), 1–7. https://doi.org/10.5281/ZENODO.1339576

TABLE OF TABLES

TABLE OF FIGURES

TABLE OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Introduction

This chapter presents an introduction and rationale for the research undertaken. It begins by identifying the main problems existing in the SE domain and highlighting the key benefits of ontologies. Then, specific research questions are presented. Next, the goals of the research are outlined. Fourth, the research process used to study the problem is presented. Fifth, the study's findings and contributions are summarized. Finally, an outline of the thesis structure is provided.

1.2 Background

Large socio-technical systems are often characterized by a high degree of complexity, a strong interaction with their environment, and an integration between human, cyber, and physical components. Systems engineering (SE) is an interdisciplinary and scientific approach that leverages systems thinking, system life cycle models and system life cycle processes to enable the successful realization of complex systems, often characterized with large scale, high intricacy and significant heterogeneity (Buede and Miller, 2016; Haskins and Ruud, 2018; Kossiakoff *et al.*, 2011; Wasson, 2015). It is responsible for delivering complex projects to meet the needs of the stakeholders in the system life cycle.

SE is lauded to address various levels of complexity in projects, as evidenced by the broad application of SE approaches and processes. For example, the Øresund Bridge is the world's largest composite structure that has the longest cable-stayed bridge span in the world and costs 30.1 billion DKK (4.03 billion EUR). The success of this award-winning bridge could not be achieved without an outstanding SE team that helped define comprehensive requirements, carry out systematic risk analyses, and collaborate with all the stakeholders (Dahlberg, 2016). The Shanghai Transrapid project cost 10 billion Yuan (1.1 billion EUR) and took 2.5 years to complete. The super-high-speed train using state-of-the-art maglev technology can reach almost 320 km/h in 2 min. SE played an essential role in extending the effort performed in concept exploration (Prasad *et al.*, 2019). It reduced the risk of hasty commitments without adequate study and is praised for integrating numerous disciplines into coordinated team efforts, from the concept design to the disposal (Vargas *et al.*, 2016).

SE is a compelling methodology yet it is not so easy to fully master, as the SE body of knowledge covers a wide range of fundamentals, principles, models of foundational systems sciences, associated management, and engineering processes. It is a multi-faceted knowledge base system that spans multiple disciplines and requires academic and professional knowledge applications. As argued by Fraga and Llorens (2015), the primary challenge for the SE novice is SE itself because the SE body of knowledge involves a lot of terminologies, concepts, definitions, and models. All these fundamental elements consist of a variety of knowledge areas, and these knowledge areas make up the substantial and complicated SE body of knowledge (Haskins, 2014).

SE is very creative in nature and evolves in an unpredictable manner. However, this makes generating a shared conceptualization of the SE body of knowledge very challenging (Chourabi *et al.*, 2010; Honour and Valerdi, 2014). Although SE is proven to be the answer to many complex questions associated with cross-organizational integration, there is no universally accepted theory within the SE discipline. The main reasons that hinder the efficient implementation of SE knowledge are as follows.

First, the responsibility for implementing effective SE practices belongs to engineering and project management. In general, engineering and project management focus on international standards, industry-recommended practices, or organizational guidelines to assist in decision-making processes to adjust SE strategies properly. Unfortunately, there is limited information in common standards for SE, forcing project management to rely on individuals and organizations, rather than relying on knowledge-based system thinking decisions. Although the best practices of SE have been documented in various handbooks and standards, e.g., the ISO/IEC/IEEE 15288: system life cycle processes (ISO/IEC/IEEE International Standard, 2015), the International Council on Systems Engineering (INCOSE) SE Handbook (INCOSE, 2015a), the ISO/PAS 19450: object-process methodology (ISO/PAS International Standard, 2015), these resources while invaluable are still hardcoded and shared as PDFs which are not the format of choice for semantic representation of information and knowledge (Di Maio, 2011).

Second, SE is transiting from a document-based approach to a model-based discipline (Madni and Sievers, 2018a; Rosa *et al.*, 2019). Model-based SE (MBSE) depends on the creation of digital artifacts or models to simulate the system life cycle processes. It requires that all the digital artifacts should be traceable and machine-readable, as data exchange and information transfer between various artificial models are the key to success of realizing the system (Eito-Brun, 2016). Nevertheless, current SE standards still remain document-centric (Ernadote, 2017), neglect the importance of system interoperability (Givehchi *et al.*, 2017), and provides little guidance on knowledge sharing and integration (Engel *et al.*, 2018). Therefore, SE standards require an upgrade to radically change from the current static presentation and natural language description to more robust and explicit knowledge representation.

Third, SE is multidisciplinary. It is commonly seen that SE projects often involve many stakeholders who are equipped with a variety of competencies and skills (Haskins and Ruud, 2018; Sarder *et al.*, 2007). When considering the multitude of stakeholders involved in the entire system life cycle processes, it is hard to maintain a unique vocabulary due to the diverse set of terminologies used across the project team (Rousseau *et al.*, 2016). Even if there are some commonalities, each specialist will integrate specific terms according to the type of the system being designed (Ernadote, 2015). These issues are magnified when previously separate communities start working together. The same term is often applied to different concepts (semantic problem), and different terms may be used to denote the same entity (syntax problem) (Lin and Harding, 2007).

Due to these problems within the SE domain, researchers are increasingly calling for a new method to aid the model-based transition, enable efficient information exchange and communication, and portray the SE body of knowledge in a shared manner. This research aims at providing a possible solution for addressing these problems. The specific research gaps are identified from the literature and synthesized in the next section to reveal the true needs for this research.

1.3 Problems and Gaps

Research has shown that significant system failure costs are caused by a lack of adequate standardization of SE (van Ruijven, 2013). In other words, SE still does not have a comprehensive, detailed knowledge representation that describes the concepts that domain experts agree upon, as well as their terms, definitions, and semantics. Although some initial effort has been made to reduce the miscommunication and misunderstanding between stakeholders by formalizing SE terminologies, there still lacks a cohesive knowledge representation to conceptualize the entire SE body of knowledge (Adcock *et al.*, 2016; Di Maio, 2010; Martin *et al.*, 2013).

SE is a multidimensional, interdisciplinary knowledge process that contains complex problems that cannot be easily solved by traditional engineering paradigms. The multidisciplinary nature of SE creates a very high level of complexity and confusion about how to use, choose, and tailor SE standards and life cycle processes (Ward *et al.*, 2018a). This problem is accentuated by the existence of many different standards and processes in the SE knowledge domain. Moreover, two significant shortfalls have not been completely resolved: the difficulties in developing systems on budget and on time (Dwivedi *et al.*, 2013), and the considerable waste of resources dealing with the correction of mistakes (Hallberg *et al.*, 2012).

Four reasons can summarize the causes of these problems according to the literature: (1) the implicit nature of SE, (2) the limitations of best-practice standards and meta-models, (3) the absence of a widely accepted and consistent terminology, and (4) inefficient collaborations due to the misunderstanding and misinterpretation.

(1) The implicit nature of SE. SE originates from heuristics and personal experience. Research has shown that the SE discipline has been recognized for 50 years as essential to the

development of complex systems (Haskins and Dahl, 2013). However, SE is still treated primarily as heuristics learned by each practitioner during the personal experimentation of a career (Honour, 2004). The heuristics known by each differ, as shown by the fractured development of SE standards. As a result of this heuristic understanding of the discipline, it has been nearly impossible to quantify the value of SE (Hutchison *et al.*, 2017). However, both practitioners and managers intuitively understand that value. They typically incorporate some SE practices in every complex program. The differences in understanding, however, result in disagreement over the level and formality of the practices to include.

(2) Limitations of best-practice standards and meta-models. Although certain standards, such as ISO/IEC/IEEE 15288, provide information on how to implement the best practice of SE processes, they are limited to human-readable descriptions and are not computer interpretable (Yang *et al.*, 2017). Meanwhile, the Model-based Systems Engineering (MBSE) approach requires that the models which are created in the system life cycle should consistently produce the expected deliverables (Haskins, 2011). However, existing meta-models that support MBSE deploy a language that is sometimes unfamiliar to some intended users (Giachetti, 2015).

(3) The absence of a widely accepted and consistent terminology. Most of the views of SE presented by current standards and handbooks are primarily process-centric. Nevertheless, SE processes are not sequential, and the tasks are performed in a parallel and iterative manner (Chourabi *et al.*, 2010). Each step may produce engineering artifacts, such as technical documents. Experience shows that they are usually written in a language to suit local culture and circumstances (Sillitto, 2011). As a result, the same word may mean different things in different contexts, and different words are used in different domains to mean the same thing (Schindel, 1997). Such different interpretations of SE concepts by individuals and communities can lead to misunderstanding and misinterpretation in the development of systems (Dori and Sillitto, 2017).

(4) Inefficient collaboration caused by misunderstanding and misinterpretation. Research shows that the major reason for system failure costs is a lack of adequate information exchange and communication within projects (van Ruijven, 2013). A quarter of these failures arise during the design phase of a system, which can be traced back to a lack of efficient collaborations between parties involved in the system life cycle processes. According to Hallberg *et al.* (2014), there is no unambiguous and comprehensive use of concepts in the field of systems development. This causes misunderstandings, misinterpretations, and irritations in the development of systems, in the most severe cases inhibiting the functioning and usability of the emerging system.

1.4 The rationale for the Study

The effective application of SE practices requires an understanding and experience of SE domain knowledge and systems thinking. SE is a developing discipline, which is subject to ambiguity and interpretation due to its state of maturation. As an evolving knowledge-based discipline that originated from the need for a multidisciplinary and integrated approach, SE has grown to become a recognized academic field of study that is inherently rooted in the application of knowledge-based systems for the management of complex projects. To date, much research has been conducted focusing on defining the common language, including basic concepts, and describing the behavior of the integrated relationships of a shared conceptualization of SE, which are commonly referred as ontologies for SE (Ernadote, 2015; Giachetti, 2015; Martin *et al.*, 2013; Ring, 2002). Ontologies, an emerging means of knowledge representation and information technology, are lauded for providing explicit, formal, and shared specifications of a domain (Guarino, 1998, 1992; Guarino and Giaretta, 1995). Madni *et al.* (2001, 1998) assert that ontologies can ensure that multiple systems share a common terminology, which is the essence of knowledge sharing and reuse. Aslaksen *et al.* (2011) claim that ontologies allow a computer to understand the meaning of a text and enable a much more productive interaction between humans and machines. Mezhuyev (2014) indicates that formal definitions for the different properties and processes of SE would be a significant contribution toward improving accuracy and precision in the implementation of SE. By using a predefined ontology, it is possible to reduce the number of misinterpretations within projects.

However, ontology-based SE (OBSE) is still in its infancy. Despite the fact that several ontologies have been created for solving particular SE problems (Engel *et al.*, 2018; Givehchi *et al.*, 2017; Hallberg *et al.*, 2014), there is a lack of mature and formal ontologies for depicting the entire SE body of knowledge (Adcock *et al.*, 2016; Di Maio, 2010; Martin *et al.*, 2013). Researchers are calling for more research to develop a complete, formal, and heavy-weight SE ontology to specify the domain vocabularies, concepts, relationships, and axioms for the entire knowledge domain (Di Maio, 2011; Dogan *et al.*, 2014; Ring, 2002; Schneider *et al.*, 2012). The International Council on Systems Engineering (INCOSE) is advocating a classification ontology to specify the SE body of knowledge and accelerate the MBSE transition (INCOSE, 2015a).

However, manual creations of a comprehensive and complete SE ontology may be beyond human ability, as various SE concepts need to be fully taken into account (Hepp, 2006). The available ontologies for SE are either developed for partial SE subdomains or have low degrees of formality (Yang *et al.*, 2019a). Building new ontologies requires substantial investment in time, effort, and cost (Liu *et al.*, 2017). Not only is unifying the language and maintaining a consistent terminology difficult but also dealing with clashes of concepts and complex relations is even more challenging (Raskin, 2006). SE is seeking a new method to replace the traditional manual codification in ontology development to save the substantial investment in time, effort, and cost (Liu *et al.*, 2017).

Therefore, to address the knowledge acquisition bottleneck, ontology learning approaches are advocated to automate the process (Hourali and Montazer, 2011; Jirkovsky *et al.*, 2017; Zhou, 2007). Ontology learning deals with discovering concepts and determining how such concepts can be grouped, related, and subdivided according to their semantics (Asim *et al.*, 2018). It often uses natural language processing (NLP) techniques and machine learning (ML) methods to process textual documents and discover implicit semantic relations (Cimiano, 2006). It has been advocated to solve the knowledge acquisition bottleneck in manual constructions (Hourali and Montazer, 2011; Park *et al.*, 2010). It exploits recent progress in ML and NLP techniques, focusing on knowledge storage and retrieval to enable scientists to explore potential new materials more effectively (Remolona *et al.*, 2017).

However, little research has been found in developing SE ontologies through the ontology learning approach. The ontology learning approach is a new and emerging area. No one has used an ontology learning approach to establish a formal ontology for the entire SE body of knowledge. The state-of-the-art and the future roadmap of ontology-based SE (OBSE) are also not clear and require further and in-depth investigation. Therefore, this research aims at addressing these problems.

1.5 Goals of the Research

In order to leverage the advanced techniques in ontology learning and establish a formal SE ontology for the entire body of knowledge, this research is determined to tackle the following questions.

First,

What is the state of the art of ontology development in SE?

This question is to explore the status quo of ontology-based (OBSE) and further composed by the following three sub-questions.

1.1) What SE knowledge areas are supported by ontologies, and to what extent?

1.2) Why are ontologies created for these areas?

1.3) What SE ontologies are existing and how formal are they?

Second,

What can be further investigated in OBSE?

The following sub-questions are designed to aid the exploration.

> *2.1) How to improve the current manual creation of SE ontologies? How to automate the process (i.e., from ontology engineering to ontology learning)?*
> *2.2) How can the learning process make use of the extant SE standards?*
> *2.3) How to model the learned SE ontology?*

Third,

What improvement does the learned ontology make to the current SE body of knowledge?

The research is in three pivotal phases: (1) analysis of the literature, (2) development of a novel ontology learning approach, and (3) validation of the proposed approach. Therefore, the research goals are also delineated in three.

First, the research aims at understanding the state of the art of ontology development and application in the field of SE domain. This includes

- creating a typology for analyzing the extant literature,
- reviewing what SE knowledge areas are already supported by ontologies based on the typology,
- investigating what contributions that ontologies have made,
- reviewing the existing SE ontologies regarding their ontological primitives, and
- evaluating these ontologies from a technological ontology engineering perspective.

Through the literature review, this research identifies and defines the limitations and gaps in the extant studies and proposes a new roadmap for future research directions of OBSE.

Second, this research intends to propose a novel ontology learning methodology for learning a formal ontology for the SE body of knowledge. To be specific, this goal includes

- specifying the stages, tasks, activities, and methods of the ontology learning process,
- generating conceptual, logical and data models for depicting the SE ontology,
- formulating methods for extracting critical components of the SE ontology, and
- devising visualization strategies of the learned SE ontology.

Third, this research plans to evaluate the proposed ontology learning methodology through a case study. This consist of

- extracting the ontology components or ontological primitives from a real SE standard,
- populating the ontological primitives into the ontology models,
- developing the concept hierarchy and taxonomic relations,
- visualizing the ontology through sophisticated tools, and
- applying the ontology to a specific scenario to demonstrate its functions.

1.6 The Research Process

This research employs a thorough research plan to realize the research goals. It is demonstrated by the Integrated DEFinition (IDEF) approach in Figure 1.1.

Figure 1.1 employs the IDEF0 method, which is the functional modeling method. The two primary modeling components of IDEF0 are functions (represented on the diagram by boxes) and the data and objects that interrelate those functions (represented by arrows).

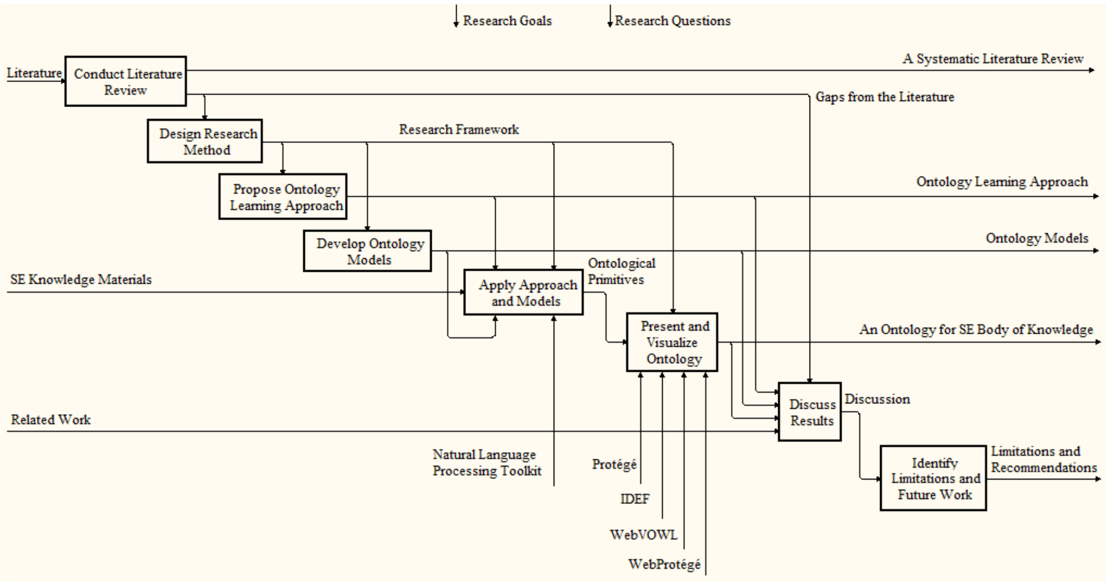


Figure 1.1 Research process modeled by IDEF0

The entire process can be divided into eight functional blocks, which are

- conducting a state-of-the-art literature review,
- designing a scientific research method,
- proposing an ontology learning approach,
- developing ontology models,
- applying the ontology learning approach and the ontology models to develop an ontology,
- presenting and visualizing the developed ontology,
- discussing the results, and
- identifying limitations and future work agendas.

The following explains each function in more detail.

(1) Conducting a state-of-the-art literature review. As seen in Figure 1.1, the top left box represents the first activity, which focuses on synthesizing the literature and identifying the research gaps. In order to understand the state of the art of the development and implementation

of ontologies in the SE domain, a systematic literature review (SLR) was developed. The review results can reveal the gaps and help analyze the status quo.

(2) Designing a scientific research method. Based on the gaps and problems identified from the literature, the research questions were defined. In terms of the research questions, a scientific research framework was generated to address the research questions and achieve the research goals. This research framework acts as a control of the whole research process. It helps steer the research directions and control the research progress.

(3) Proposing an ontology learning approach. This step is the core of the research process, which is to propose the ontology learning methodology for developing the SE ontology. As a methodology, it was specified in phases, tasks, activities, methods, as well as outputs. This ontology learning approach was designed based on the latest best practices. It is also a universal approach that can be tailored to suit other disciplines.

(4) Developing ontology models. The ontology models are developed under the control of the research framework. They can be seen as the overarching structure in which the SE knowledge can be gathered. Three ontology models, respectively representing the conceptual, the logical, and the data facets, are invented. The conceptual model is developed through an analysis of the characteristics of the SE body of knowledge standards, representing the terminologies, concepts, descriptions, and properties. The logical models consider the multiple parenting taxonomic relations in the SE ontology, providing a polyhedron hierarchy for the data representation. The data model deals with the precoordinated and postcoordinated expression rules and grammar in the ontology. All of the three models become a part of the well-shaped and predefined structure to capture the SE knowledge in a standard and systematic way.

(5) Applying the ontology learning approach and the ontology models to develop an ontology. This is the application and implementation stage of the research process. A case study was carried out to apply the proposed ontology learning approach and the developed ontology models in practice. Both the ontology learning approach and the ontology models need to be tested in real case studies. The INCOSE SE handbook (INCOSE, 2015a) was chosen to test the approach and the models to derive an SE ontology. The handbook, acting as the SE knowledge materials, was processed by Natural Language Processing (NLP) tools. The output from the processing is a series of ontological primitives. These are the components of an SE ontology. The complete case study was recorded and presented in detail.

(6) Presenting and visualizing the developed ontology. As an artifact, the SE ontology must be easily used and clearly presented. The visualization of ontology is also an important component of the whole research process. Sophisticated tools were selected to help present the complex relations in the SE ontology. The ontology was edited in web ontology language (OWL),

presented and visualized by a suite of software, such as Protégé and WebVOWL (Web-based Visualization of OWL).

(7) Discussing the results. The learning approach, ontology models, and the developed ontology were evaluated and compared with extant work.

(8) Identifying limitations and future work agendas. The final step is to summarize lessons learned and provide future research recommendations.

The key deliverables of this study include

- a comprehensive literature review detailing the state-of-the-art of OBSE,
- a novel ontology learning approach for learning an SE ontology,
- a set of ontology models to depict the SE ontology,
- a formal ontology deriving from authoritative SE standards to conceptualize the entire SE body of knowledge, and
- a visualization of the SE ontology by sophisticated ontology tools.

The next section elaborates on the findings from each research activity and the contributions of the research deliverables in more detail.

1.7 Findings and Contributions

The findings of this research can be summarized in four aspects.

(1) Literature review results from the state-of-the-art review on OBSE. At the beginning of the study, a comprehensive review of the literature was conducted. The entire SE knowledge domain was firstly divided into specific knowledge areas. Consequently, a typology representing the classification of the SE knowledge areas was created. Based on the typology, a systematic literature review (SLR) was conducted to see what knowledge areas have been supported by ontologies.

The results indicate that ontologies have been applied in various SE knowledge areas. However, only seven areas have more than ten research articles. Most areas have only one or two pieces of evidence. This shows that in the SE domain, there is a dearth of research on ontologies. The existing research is still at a very preliminary stage. Furthermore, it has been found that 53 out of 116 articles have reported their ontology models. This means ontology engineering has been applied to the SE domain. Unfortunately, only 14 has presented the details, which makes the ontology sharing, mapping, and reuse very limited. Nevertheless, only eight ontologies are supported by formal ontology engineering methods and tools, which indicates that most ontologies are developed based on heuristics or the authors' own experience. The quality, degree of formality, and completeness are thereby doubtful. Moreover, none of the ontologies considers the entire SE body of knowledge. Only manual codification methods are used in developing ontologies, leaving the application of the ontology learning approach a giant gap.

Based on the results of the literature review, conclusions can be drawn as follows.

- No ontology captures the entire SE body of knowledge.
- The methods of developing SE ontologies are all manual and in need of automation.
- The extant ontologies remain at a general level and lack a detailed representation.
- The development of the ontologies requires using sophisticated languages and tools to increase the formality.
- Little work has been done regarding the visualization of the ontologies and their potential application scenarios.

However, there is also some positive feedback from the literature review. For example, the advantages of ontologies are highly acknowledged by the SE community. Many ontologies are created and play an essential role in improving a particular aspect of SE. Although most of the time, the contributions of ontology are implicitly hidden in the research results, the benefits of ontologies are still praised by the SE community. In order to improve the current state, it is necessary for future research to clarify the intention or purpose of using or creating ontologies at the early stage of the study. Otherwise, there will be unfavorable consequences in the later development stage of ontologies.

(2) The ontology learning approach shows great advancement in SE knowledge acquisition. From a methodology perspective, this research proposes a novel methodology for developing SE ontologies through the ontology learning approach. As found from the literature, previous studies all define SE ontologies by manual codification. This traditional ontology engineering method is criticized as tedious, time-consuming, and error-prone, as it relies heavily on the inputs from domain experts and shows a knowledge acquisition bottleneck. On the contrary, the proposed ontology learning methodology results in a high-efficiency and high-intelligent ontology creation process. Also, it makes full use of the extant SE standards, enabling the reuse of rich legacy repository.

This novel methodology employs natural language processing (NLP) techniques to carry out the lexical and morphological analyses on the standard documents. From the learning process, important terminologies, synonyms, concepts, and relations constructing the SE body of knowledge are automatically recognized and classified. A formal and sophisticated SE ontology is achieved, which can be used to harmonize the extant standards, unify the languages, and improve the interoperability of the MBSE approach.

(3) A formal ontology for the SE body of knowledge. From the perspective of ontologies as an artifact, this research also contributes to a formal domain ontology for the SE body of

knowledge for both descriptive and classification purposes. First, this research created three ontology models for depicting the SE body of knowledge, i.e., the conceptual model, logical model, and data model for ontology learning and development. Through a comprehensive case study, these three models are proven to be applicable and valid, especially for their rationality. Second, the ontology created contains a concept hierarchy with nine top-level classes. These classes depict the entire universe of the SE domain and can be specified in subclasses. This research has identified areas in which there is limited information pertaining to the necessity for the SE ontology development. As the SE community has not yet identified leading ontologies that can be considered for adoption or modification, this ontology can be generalized to a wider audience. Third, the ontology provides a controlled vocabulary of the SE terminology. It reduces language chaos and eliminates terminological inconsistency and ambiguity. Considering the multidisciplinary nature of SE, this SE ontology can be a communication bridge that links various stakeholders involved in a complex SE project.

(4) Restructure of the system life cycle processes through ontology reasoning. Currently, the SE life cycle processes are presented in a linear sequential manner without providing an overall picture of how the processes are linked. The different processes rely solely on isolated diagrams, called the Input-Process-Output (IPO) diagrams (INCOSE, 2015b), showing each process as standalone without properly highlighting the connections with all the other processes. To apply SE successfully and tailor SE processes to different organizations, IPO diagrams need to have feedback and feedforward mechanisms. From the perspective of the reconceptualization of the system life cycle processes, the original static and isolated IPO diagrams are upgraded and connected into a more dynamic, robust, and interrelated knowledge representation. The appendices of this thesis provide all 34 life cycle processes in 16 diagrams with their interrelations derived from the SE ontology. Moreover, this knowledge representation can be very supportive of process tailoring and process reorganization.

The research progression spanned four years, beginning with an investigation into existing work; development of an ontology learning methodology and three ontology engineering models; and, finally, application of the methodology and models in a case study. The outcomes associated with this progression include six peer-reviewed papers as listed in Table 1.1.

Table 1.1 Publication artifacts

Paper	Title
Yang *et al.* (2016)	An ontology model for systems engineering derived from ISO/IEC/IEEE 15288: 2015: systems and software engineering-system life cycle processes
Yang *et al.* (2017)	Towards a methodology for systems engineering ontology development - An ontology for system life cycle processes
Yang *et al.* (2019a)	Ontology-based systems engineering: A state-of-the-art review
Yang *et al.* (2019b)	Learning systems engineering domain ontologies from text documents

Paper	Title
Manenti *et al.* (2019)	Functional modelling and IDEF0 to enhance and support process tailoring in systems engineering
Yang *et al.* (2020)	Ontology learning for systems engineering body of knowledge

The initial investigation is made to define the requirements for developing an ontology model for the system life cycle processes. Yang *et al.* (2016) contain a framework for creating ontology models for SE. The ontology models specify the concepts or classes, relationships, and logical axioms within the SE knowledge domain. They are used to infer the concept hierarchy based on the constraints made by the properties, which are known as the logic level of the ontology model.

Yang *et al.* (2017) lay the foundation of knowledge acquisition and knowledge representation and distinguish different knowledge representation models into four levels according to the semantic primitives. At the preliminary level, the typical representative is to use natural language to explain terminologies, also known as a glossary, which is a list of terms and their textual definitions. At the taxonomy level, terms are organized into a collection with a hierarchical structure, which describes parent-child relations between each term. At the thesaurus level, lexical relations are used in addition to parent-child relations to create a networked collection of concepts. Finally, at the ontology level, concepts are defined in a more formal logic-based language by combining the previous relations with other more complex relations between concepts to completely represent a certain knowledge domain.

Yang *et al.* (2019a) analyze the knowledge representation of SE and review the state of the art of ontology-based SE. The review is to draw a clear roadmap of how ontologies support SE and to determine what extent they have been applied in this domain. This review contributes to a holistic examination of the primary studies relevant to the topic of ontology-based SE, spanning nearly two decades. The findings provide an integrated and comprehensive understanding of and shed new light on (1) the SE knowledge areas supported by ontologies; (2) the contribution that ontologies make to SE problems; (3) the existing ontologies that are created to support SE; and (4) the techniques adopted from an ontology engineering perspective. It assesses the influence of ontologies in SE knowledge areas, expounding and highlighting the effects of ontologies.

Yang *et al.* (2019b) exploit the ontology learning approach and establish the architecture of the learning process for deriving the ontology, with a map between the adopted methods and the deliverables or outputs. Eight tasks are defined for learning domain ontologies according to the constitution of an ontology, i.e. term, synonym, concept, concept hierarchy, relation, relation hierarchy, axiom and general axiom. These elements are corresponding to the so-called ontology learning layer cake.

The case study contains two parts and was written in two articles. The first part involves applying the proposed ontology learning approach and the ontology models to a textual standard of SE, and obtained a formal ontology for the SE body of knowledge (Yang *et al.*, 2020). The second part is a collaborative effort. Manenti *et al.* (2019) reported the application scenarios of the SE ontology, which are to restructure the system life cycle processes using IDEF0 and conduct process tailoring.

1.8 Thesis Structure

This section provides an overview of the research progression cross-referenced with papers and outcomes, and an outline of the subsequent chapters in this thesis. Table 1.2 illustrates the related papers produced in the course of addressing the research questions.

Table 1.2 Research progression cross-referenced with papers

Thesis chapters	**Related papers**
Chapter 2: Theoretical Foundations	Yang *et al.* (2017)
Chapter 3: Literature Review	Yang *et al.* (2019a)
Chapter 4: Research Methodology	Yang *et al.* (2019b) Yang *et al.* (2020)
Chapter 5: Ontology Modeling	Yang *et al.* (2016)
Chapter 6: Case Study	Manenti *et al.* (2019) Yang *et al.* (2019b) Yang *et al.* (2020)

Chapter 2: Theoretical Foundations presents the theoretical lens for this research. It begins by introducing SE in terms of the definitions, important system fundamentals, and a classification of the SE knowledge areas. Then, the definitions and formalisms of ontologies are introduced, followed by the features and different types of ontologies. Next, the ontology learning method is introduced, including the definitions and the role in the SE ontology development. Then, the model of ontology learning layered cake is presented, followed by the ontology learning methods and tools. Finally, knowledge related to ontology modeling and process modeling is respectively expounded.

Chapter 3: Literature Review presents a state-of-the-art review of the literature on ontology-based SE (OBSE). It begins with an introduction to the systematic literature review (SLR) process adopted in this research. Then, it highlights the search strategy used in the literature retrieval and the 7-step SLR methodology designed for the literature review. Third, it presents the four review questions dedicated to investigating the state of the art of OBSE. Finally, this chapter reviews research on the development and application of ontologies in SE in terms of the knowledge areas, reasons for application, scope, and techniques used.

Chapter 4: Research Methodology outlines the overall framework of methodologies and methods used to develop SE ontologies through the ontology learning approach. First, it uses IDEF0 to present the research framework. Second, the chapter presents the proposed ontology learning methodology for learning an SE ontology to conceptualize the body of knowledge. Three stages are respectively described in the ontology learning process.

Chapter 5: Ontology Modeling presents three models for depicting different perspectives of ontology engineering of the SE ontology, which are respectively the conceptual model, logical model, and data model. These models are the overarching structure in which the SE knowledge can be gathered and ensure the SE knowledge is captured in a standard and systematic way.

Chapter 6: Case Study provides a comprehensive case application of the proposed ontology learning methodology and developed models. It also shows the SE ontology that is created for conceptualizing and formalizing the body of knowledge, including the terminology, top-level concepts, taxonomic relations, and the concept hierarchy. Moreover, an application scenario of the developed ontology is presented. It helps reorganize the system life cycle processes and exposes the implicit relations within the processes. A set of IDEF0 diagrams are generated to illustrate the life cycle processes after reorganization.

Chapter 7: Summary of the work, Contributions, Limitations, and Recommendations for Future Work, provides a summary of the research on learning ontologies for the SE body of knowledge. It highlights the contributions and limitations. It also provides a recommendation for future research directions. Finally, it concludes the thesis.

1.9 Conclusion

This chapter presents a summary of the research undertaken. First, a background to the study is provided, and particular emphasis is placed on the challenges identified between the document-centric standards and the model-based transition. Next, specific problems guiding this study are identified. Third, the goals and objectives of the study are outlined. Fourth, the research process used to study the problem is presented. Fifth, the study's findings and contributions are summarized. Finally, an outline of the thesis structure is provided.

CHAPTER 2 THEORETICAL FOUNDATIONS

2.1 Introduction

This chapter provides a theoretical foundation and lens for this research. In this chapter, the extant literature related to SE, ontologies, ontology learning, and ontology modeling is presented. First, it defines what SE is and provides a theoretical background about fundamental systems concepts, such as systems science, systems thinking, systems praxis, and system life cycle processes. Then, it presents an original classification of the SE knowledge areas. Next, the literature related to ontologies is synthesized, including the definitions, features, and classifications. Fourth, the definitions of ontology learning are presented, and the role of ontology learning in SE ontology engineering is depicted. Fifth, the ontology learning process is explained, focusing on the model of ontology learning layered cake and the technical learning methods and tools. Finally, the notion of process modeling is set forth.

2.2 Systems Engineering (SE)

Systems engineering (SE) is an overarching discipline, providing the tradeoffs and integration between system elements to achieve the best overall product and service. It is a multidisciplinary engineering field that needs to integrate many academic and technical fields. Although there are some important aspects of project management in the SE life cycle processes, it is still much more of an engineering discipline than a management discipline (Aslaksen *et al.*, 2011). It also overlaps with other engineering disciplines, such as software engineering, human factors engineering, and industrial engineering. The systems approach, as well as the systems theory, have evolved over the years. Therefore, it is essential to clarify the range of the SE body of knowledge first to define the scope of this study.

2.2.1 Defining SE

Among the several definitions of ‘system’ that exist, there are some that appear more commonly than others, although in slightly different shapes. Buede and Miller (2016, p. 3) define a system as “a collection of hardware, software, people, facilities, and procedures organized to accomplish some common objectives”. Kossiakoff *et al.* (2011, pp. 3–4) emphasize a system is “a set of interrelated components”. This emphasis implies a multiplicity of interacting parts that collectively perform a significant function. In other words, a system is “an integrated set of interoperable elements or entities,” as Wasson (2015, p. 3) argues, “each with specified and bounded capabilities, configured in various combinations.”

There are many ways to interpret SE. It can be a perspective, a process, and a profession, as illustrated by the following three different but representative definitions.

INCOSE (2015a, p. 11) defines that “SE is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: operations, performance, test, manufacturing, cost and schedule, training and support, and disposal. SE integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.”

According to Eisner (2008, p. 5), SE is “an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near-optimal manner, the full range of requirements for the system.” As can be seen from this definition, there could be more different definitions, applications, and the declarative statement that define SE, which can be accomplished within the context of which SE is applied. This is done through a formal and organized process in which governance guidelines and life cycle processes are used to manage, control, and objectively demonstrate stakeholder requirements at any and all applicable phases of the system life cycle.

In the field of science and technology, SE is a systematic interdisciplinary approach to project delivery of applications, structures, or components. The National Aeronautics and Space Administration (NASA) has played an influential role in promoting the development of SE. NASA (2016, p. 3) defines SE as “a methodical, multidisciplinary approach for the design, realization, technical management, operations, and retirement of a system. It is a way of looking at the big picture when making technical decisions. It is a way of achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the system within cost, schedule, and other constraints. It is a methodology that supports the containment of the life cycle cost of a system. In other words, SE is a logical way of thinking.”

For this research, SE is considered in terms of an interdisciplinary approach and means to enable the implementation of a successful system. It contains all the three aspects, a perspective, a process, and a profession. It emphasizes defining customer requirements and essential functions in the early development cycle and then conducts design synthesis and system validation. SE integrates all disciplines and specialized groups into one team, creating a structured development process from concept to production to operation. SE considers the business and technical needs of all customers intending to provide high-quality products that meet their needs. SE bridges the natural gap in subject-specific engineering practice by facilitating a systematic and structured process of implementation throughout the product or project life cycle.

Traditional definitions of SE have emphasized the sequential performance of SE activities (INCOSE, 2014). However, modern SE has a more global context shaped by the global environment, human and social needs, policy and business challenges, as well as the technologies that underlay systems. The entire intellectual and practical endeavor creating holistic solutions to complex system challenges consist of the SE knowledge domain. To sum up,

- SE is a management process of improving the delivery of high-quality products and services, with the correct people and performance features and metrics, at an affordable price, and on time.
- SE is a multidisciplinary approach to bringing systems into existence and is characterized in many ways.
- SE is described as a function of design structure, functional and performance requirements, or operational life cycle processes.
- SE is also characterized in terms of stakeholder requirements, capital investments, risks, availability, and reliability.

2.2.2 Systems Fundamentals

Systems science is both the "science of systems" and the "systems approach to science", covering theories and methods that contrast with those of other sciences, which are generally reductionist in nature (INCOSE, 2015d). It identifies, explores and understands patterns of complexity through contributions from three broad areas (Martin *et al.*, 2013): (1) foundations, which are meta-theories, such as methodology, epistemology, axiology, which help us to organize knowledge; (2) theories about systems, e.g., complexity, cybernetics, which identifies patterns abstracted from and applicable across domains and specialties; and (3) representations that allow insight into and communication about systems and their contexts, by describing, exploring, analyzing, and making predictions. The concepts, principles, and patterns of systems thinking arise from the work of practitioners applying the insights of systems science to real-world problems.

Systems thinking is another term that is used widely but without explicit agreement on meaning. The emergence of the systems movement in the 20th century was primarily due to advances in mathematical and computer technology. Systems thinking is the art of simplifying complexity. It is about seeing through chaos, managing interdependency, and understanding choices (Gharajedaghi, 2011).

Martin *et al.* (2013) define the relations amongst the three key SE fundamentals: systems science, systems approach, and systems thinking. Integrative systems science allows identifying, exploring, and understanding patterns of complexity relevant to a problem. The systems approach to practice draws on integrative systems science to address complex problems and opportunities. Systems thinking binds the two together through appreciative and reflective practice using systems-paradigm concepts, principles, and patterns.

Systems praxis (Singer *et al.*, 2012) refers to the entire intellectual and practical endeavor of creating holistic solutions to complex system challenges. Systems concepts, principles, and methods are designed to be integrative across traditional domain boundaries. However, multiple dimensions of complexity (social, technical, environmental, etc.) may require a blend of approaches and techniques from disparate systems traditions. There are many approaches to recognizing and creating systems. Systems praxis, as a human activity system, prescribes competencies, and processes for organizing various technologies into responsive systems. This activity is exceedingly complicated by varieties of systems types and the lack of common language among systems theories and practices. With a common language for systems praxis, practitioners, systems integrators, consultants, and their employers will find it easier and faster to work successfully across multiple communities of practice to achieve a common purpose (Martin, 2012).

2.2.3 System Life Cycle Processes

SE processes usually organize themselves around the concept of a life cycle. A life cycle can be defined as a series of stages through which a system or manufactured product passes (INCOSE, 2015e). As the definition of SE, the detailed conceptualization of the life cycle is by no means unique across the communities that employ the discipline. ISO/IEC/IEEE 15288 (ISO/IEC/IEEE International Standard, 2015) is an international standard that covers the knowledge of system life cycle stages and processes. It states that a system progresses through its life cycle as the result of actions performed and managed by people in organizations, using processes for execution of these actions. Life cycle stages include concept, development, production, utilization, support, and retirement. SE can also be considered from a generic life cycle process perspective as it can be defined as an interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a solution and to support that solution throughout its life (INCOSE, 2015e). Therefore, system generic life cycle processes are one of the most critical knowledge areas in SE.

2.2.4 SE Knowledge Areas

SE is an all-encompassing term since it represents more than a logical way of thinking (Matsuoka, 2014). It is worth noting that this domain is so broad that it must be specified in concrete knowledge zones. It will benefit from a classification of detailed sub-components to better understand its knowledge areas. Substantial publications of books can be acquired to explore a wide variety of topics within the SE body of knowledge. One of the documentations that has holistically structured and classified a broad array of knowledge areas is the "Guide to the SE Body of Knowledge (SEBoK)" (BKCASE Editorial Board, 2017). The top-level classifications of the SEBoK are referred to and tailored to conduct this research. The final classification of SE knowledge areas is enriched and extended with authoritative SE standards, such as the ISO/IEC/IEEE 15288 to further specify the knowledge areas of SE. Rather than a vague and abstract definition of SE, Figure 2.1 shows a typology of the SE body of knowledge, classifying the knowledge areas into specific discourses.

The creation of this classification is based on the understanding and synthesis of widely accepted standards and tailored to fit the scope of the literature review. It acts as a catalog so that the process of the literature retrieval is systematic, and the final results can be presented clearly.

It is worth noting that the terminologies used to describe the SE knowledge areas vary slightly in different publications. For example, in the generic life cycle stages, the 'development stage' can be expressed as 'realization stage', and 'utilization stage' can also be called 'deployment stage'.

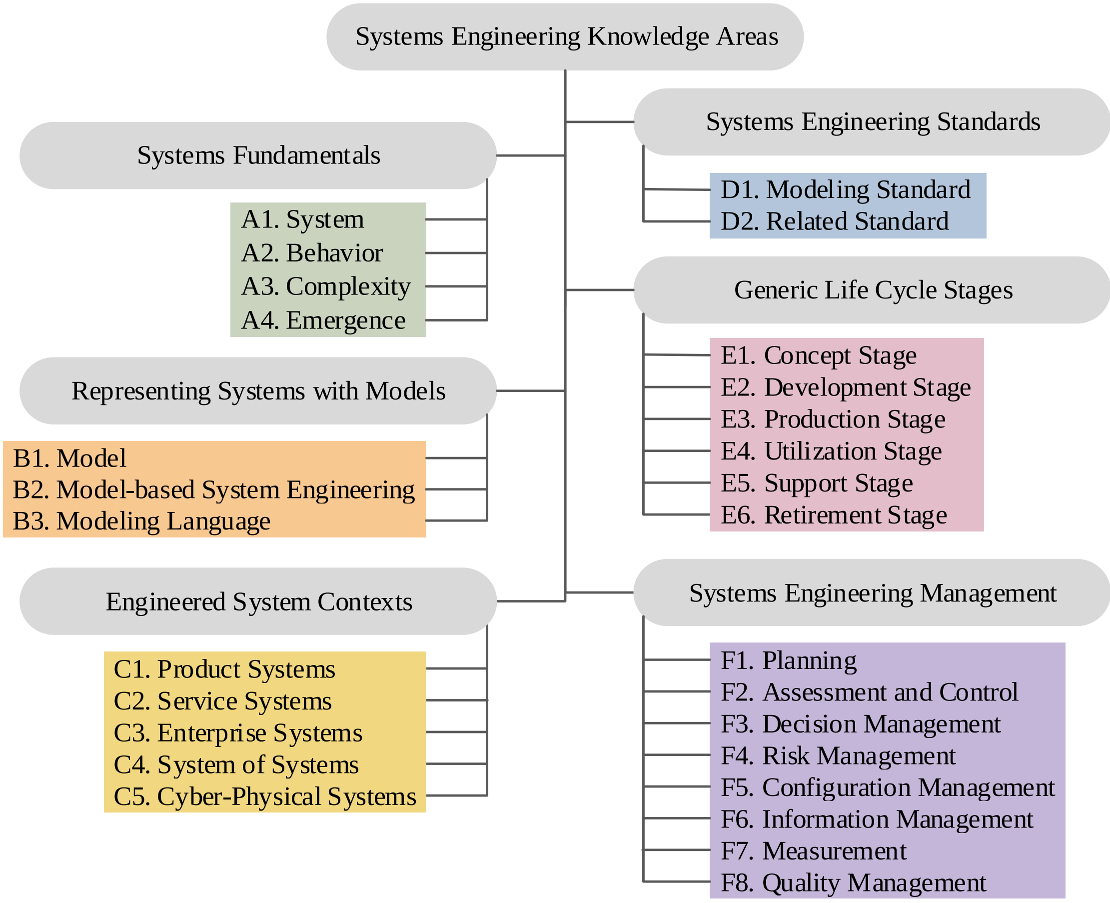


Figure 2.1 Classification of SE knowledge areas (Yang *et al.*, 2019a)

2.3 Ontologies

As stated, the SE domain is a dynamic domain with different definitions depending on the organization or standard used, which creates ambiguity. Ontologies are considered as a method to reduce ambiguity or disambiguate by providing semantics and common vocabularies upon which organizations can build descriptions and communicate activities. Ontologies facilitate the decomposition of knowledge and eliminate the need for duplicate definitions. Ontologies also lay the foundation for interoperability between different systems.

2.3.1 Defining Ontologies

In general, the word ‘ontology’ is used in two ways. On the one hand, when it is used as an uncountable noun (Ontology, with a capital O), it refers to a branch of philosophy dealing with the nature and structure of reality. On the other hand, when it is used as a countable noun (an ontology or ontologies, with lowercase letters), it refers to a special kind of computational artifact, known as “an explicit specification of a conceptualization” (Gruber, 1995). An example of the use of the term ‘ontology’ is that “ontological engineering is a branch of knowledge engineering which uses Ontology to build ontologies” (Guarino and Giaretta, 1995).

With increasing development and application of ontologies in computer science, the definition of computational ontologies is modified. Borst *et al.* (1997) emphasize that the conceptualization should express a shared view between several parties, thus redefines the meaning of an ontology as "a formal specification of a shared conceptualization." Studer *et al.* (1998) optimize this definition and consider an ontology as "a formal, explicit specification of a shared conceptualization."

The definitions of ontologies have evolved over the years, as can be seen from Table 2.1.

Table 2.1 Definitions of ontologies

Definitions from Literature	**References**	**Definitions from Literature**	**References**
explicit specification of a conceptualization	Alobaidi *et al*. (2018) Barforush and Rahnama (2012) Jiang and Tan (2010) Lau *et al*. (2009) Jung (2004)	formal and explicit specification of a shared conceptualization	Rani *et al*. (2017) Ruiz-Martínez *et al*. (2011) Gómez-Pérez and Manzano-Macho (2005)
formal and structural way of representing the concepts and relations of a shared conceptualization	Asim *et al*. (2018)	effectively formal and explicit specifications in the form of concepts and relations of shared conceptualizations	Wong *et al*. (2012)
shared conceptualization of a domain as they are assumed to reflect the agreement of a certain community or group of people	Cimiano *et al*. (2006)	formal conceptualization of a particular domain shared by a group of people	Gacitua *et al*. (2008)
shared conceptualizations for representing domain knowledge	Kong (2007)	shared formal conceptualization of particular domain between members of a community of interest, which help them exchange information	Benslimane *et al*. (2008)
form of formal representation of domain-specific knowledge	Dong and Hussain (2013)	explicit conceptualization of a problem domain	Wouters *et al*. (2005)
formal and rigorous approach for knowledge representation	Chen *et al*. (2013)	a standard for knowledge representation	Quan *et al*. (2006)
representation of entities and their relationships in a particular domain	Liu *et al*. (2011)	a formal description of a discourse domain	Hu *et al*. (2014)
the specification of the objects, properties, and relations that one would encounter in a particular domain of discourse	Cai *et al*. (2016)	a highly structured system of concepts covering the processes, objects, and attributes of a domain as well as all their pertinent complex relations	Li *et al*. (2009)
a shared understanding of some domains of interest, which is often conceived as a set of classes (concepts), relations, functions, axioms, and instances	Gaeta *et al*. (2011) Ding and Foo (2002)	a shared and common understanding of a domain that can be communicated between people and applications	Bhatt *et al*. (2004)

The table shows a collection of representative definitions of ontologies, which indicates a slight change over time to suit different research focuses. Many authors agree that an ontology is a specification of a conceptualization (Alobaidi *et al.*, 2018; Barforush and Rahnama, 2012; Gómez-Pérez and Manzano-Macho, 2005; Jiang and Tan, 2010; Jung, 2004; Lau *et al.*, 2009; Monika Rani *et al.*, 2017; Ruiz-Martínez *et al.*, 2011; Wong *et al.*, 2012).

Nevertheless, Rani *et al.* (2017), Ruiz-Martínez *et al.* (2011), Wong *et al.* (2012), and Gómez-Pérez and Manzano-Macho (2005) put extra emphasis on the characteristics that the specification and the conceptualization should possess. They argue that an ontology should have formality and explicitness in terms of the specification, and shared understanding regarding the conceptualization.

Asim *et al.* (2018), Kong (2007), Dong and Hussain (2013), Chen *et al.* (2013), Liu *et al.* (2011), and Quan *et al.* (2006) believe that an ontology is a knowledge representation or represents domain-specific knowledge. Asim *et al.* (2018) and Liu *et al.* (2011) further indicate that the knowledge representation is carried out in a way that characterizes the concepts (objects or classes) and relationships in a domain. Cai *et al.* (2016), Gaeta *et al.* (2011), Li *et al.* (2009), and Ding and Foo (2002) further point out that the components of an ontology should also contain properties (or attributes), functions, axioms, and instances.

Although these definitions reflect different viewpoints of what an ontology is, they share some common elements. Based on a synthesis of the extant definitions, this research defines ontologies as an explicit specification of a shared conceptualization of concepts and relations for formal knowledge representation in a particular domain of interest. It should be able to reflect the agreement of a specific community or group of people (Cimiano *et al.*, 2006), help them exchange information (Benslimane *et al.*, 2008), and provides a common understanding of a domain that can be communicated between people and applications (Bhatt *et al.*, 2004).

2.3.2 Terminological and Assertional Formalism

Ontologies are becoming extremely important in fields such as knowledge management, information systems, and semantic web, where they play a key role in defining agreed terminologies between agents, by providing essential concepts, taxonomies, relationships and domain axioms (Fensel *et al.*, 2011; Gaševic *et al.*, 2009). The effective use of ontologies requires not only a well-designed language but also rigorous logical reasoning. Therefore, built upon the well-defined description logic languages and theories, ontological knowledge bases are equipped with a terminological formalism, so-called ‘T-box’, and an assertional formalism, ‘A-box’ (Brachman *et al.*, 1983; Mann, 2003). For computational ontologies in information systems, ontologies come to mean two related things (Chandrasekaran *et al.*, 1999): a representation vocabulary (T-box), which provides a set of terms with which to describe the facts in any given domain; and a body of knowledge (A-box), meaning the fact associated with a conceptual model or ontologies within a knowledge base.

2.3.3 Classifications of Ontologies

Ontologies can also be classified into different types according to their level of generality, level of formality, and purpose of creation. Table 2.2 shows the different types of ontologies.

Table 2.2 Types of ontologies

Dimensions	**Classifications**		**References**
level of generality	top-level ontology		Navigli *et al.* (2003) Rani *et al.* (2017)
	domain ontology	foundational ontology core ontology specific domain ontology	
	task ontology		
	application ontology		
level of formality	informal ontology/lightweight ontology formal ontology/heavyweight ontology semi-formal ontology		Wong *et al.* (2012)
purpose	classification ontology descriptive ontology		Rani *et al.* (2017)

Ontologies can be represented in various forms depending on the level of abstraction or the degree of formality. They can be expressed as sets of declarative statements in natural languages. However, it is impossible for computers to process natural language statements. For more formal representations, web ontology language (OWL) is widely used in practice, supported by ontology modeling tools, such as Protégé (Musen, 2015). Well-structured and well-developed ontologies enable various kinds of examinations for logical consistency, and they also enhance the interoperability between different applications.

Based on the classification of different types of ontologies presented in Table 2.2, this research aims at creating an SE ontology that is in accordance with the following characteristics. Regarding the level of generality, this research will create a domain ontology for SE. In terms of the level of formality, this ontology will be formal and heavyweight, which means it will be developed in formal ontology languages by sophisticated tools and equipped with both T-box and A-box. For purposes, this ontology is both descriptive and a classification ontology to describe and classify the SE body of knowledge. The position of the created ontology is illustrated in Figure 2.2.

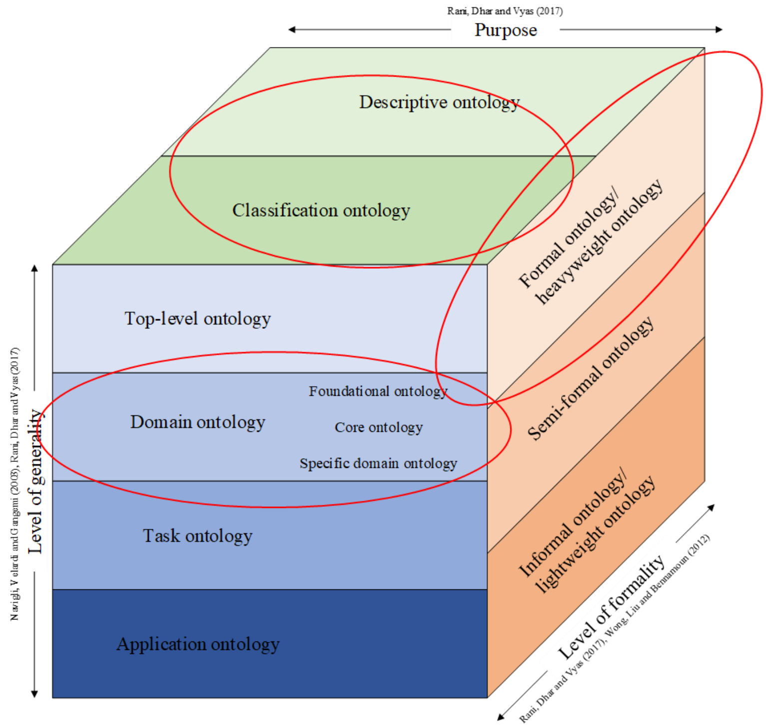


Figure 2.2 Position of the ontology developed by this research

This SE ontology can assist systems engineers and project management within the collaborative design team by providing accurate design process information and guideline. It is a hierarchical decision management tool that forms a set of terms that describe the design domain and can be used as a basis for knowledge base and shared conceptualization. As SE projects become information-driven, ontologies are becoming more and more relevant to include heterogeneous resources. The main idea behind ontology is to create common vocabularies that are logically well-defined and can be used to manage the information coming from different sources so that the information becomes integrated and the sources become interoperable.

2.4 Ontology Learning

This section presents the theoretical foundations concerning the emerging ontology learning approach. It starts by defining the concept of ontology learning. Then, an original framework of a four phases ontology engineering methodology is generated, especially for enabling the establishment of this research, indicating the role of the ontology learning approach in general ontology engineering methods. Third, a review is presented concerning the evolution of ontology learning processes over time. Fourth, the specific ontology learning methods used in this study and the supportive tools are briefly discussed to provide a foundation for later use.

2.4.1 Defining Ontology Learning

Ontology engineering is a field that studies the methodologies for building ontologies. Traditional ontology engineering methods relying on domain experts to manually create the ontologies are time-consuming, error-prone, and tedious (Hu *et al.*, 2014; Maedche and Staab, 2001; Villaverde *et al.*, 2009).

Ontology learning is an emerging approach that uses natural language processing (NLP) and machine learning (ML) techniques to solve the knowledge acquisition bottleneck in manual constructions (Hourali and Montazer, 2011; Remolona *et al.*, 2017).

From a synthesis of the extant literature, Table 2.3 provides a variety of viewpoints on what ontology learning means.

Table 2.3 Definitions of ontology learning

Definitions	**References**
The process of building an ontology for domains of interest by identifying the related concepts and the relations between those concepts in those domains using (semi-) automated approaches	Alobaidi *et al.* (2018)
A reverse process as the domain model is reconstructed from input text by exploiting the formal structure saved in the author's mind	Asim *et al.* (2018)
The process of converting text to ontology	Rani *et al.* (2017)
Automatically or semi-automatically generating ontology from some input information sources of types structured, semi-structured or unstructured	Idrissi *et al.* (2014)
The process of identifying terms, concepts, relations, and optionally, axioms from textual information and using them to construct and maintain an ontology	Colace *et al.* (2014) Wong *et al.* (2012)
The process of extracting ontological representations starting from an extensive amount of unstructured text	Gaeta *et al.* (2011)
The extraction of ontological elements from knowledge-rich resources	Liu *et al.* (2011)
extracting conceptual knowledge from several sources using a set of techniques for building ontology which can be done from scratch or enhancing the existing ontology in a semi-automatic fashion	Santoso *et al.* (2011)
Semi-automatically or automatically build ontologies from some given data with limited human intervention in order to speed up the ontology construction process and lessen its cost	Hazman *et al.* (2009)
The process of automatic or semi-automatic construction, enrichment, and adaptation of ontologies	Gil and Martin-Bautista (2014) Alves *et al.* (2009)

Definitions	References
Acquiring knowledge in the form of ontological categories such as concepts, taxonomies, properties or axioms from information sources describing a specific domain of interest	Simperl *et al.* (2008)
The set of methods and techniques used for building an ontology from scratch or enriching or adapting an existing ontology in a semi-automatic fashion using several sources	Kong (2007)
The application of a set of methods and techniques used for building an ontology from scratch by enriching or adapting an existing ontology in a semi-automatic fashion using distributed and heterogeneous knowledge and information sources, allowing a reduction in the time and effort needed in the ontology development process	Gómez-Pérez and Manzano-Macho (2005)

From Table 2.3, it can be seen that the definitions of ontology learning do not change dramatically. The definitions, in fact, share a lot of common points. Therefore, the definition of ontology learning in this research focuses on a process of extracting and identifying conceptual knowledge of a domain from the textual information, which uses a set of methods and techniques for constructing an ontology in an automatic or semi-automatic fashion from scratch or enriching an existing one.

2.4.2 Role of Ontology Learning in SE Ontology Development

As an emergent approach, ontology learning is used to extract ontological primitives during the early phases of ontology engineering. It is more cost-efficient and time-efficient to deal with the knowledge acquisition bottleneck than the traditional manual construction relying on domain experts' knowledge. Figure 2.3 presents the role of ontology learning in the development of an SE ontology in this research (Yang *et al.*, 2019b).

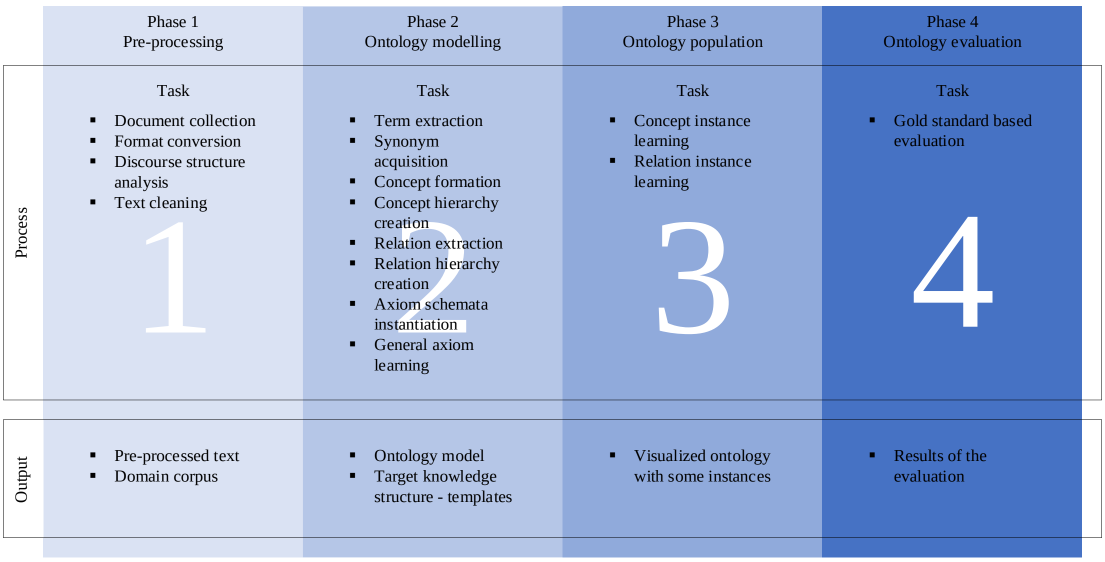


Figure 2.3 A four phases ontology engineering methodology driven by ontology learning

Figure 2.3 is a four phases ontology engineering research process. What is distinct is that it is enabled by ontology learning approaches.

Phase 1 is the pre-processing phase. In this phase, documents are collected, and available document resources in PDF format are obtained. Then, these files are converted from the PDF format to TXT format to allow text processing at a later stage. Also, structure analysis of the documents is essential in this early stage since it enables the content to be controlled for later phases. Also, the cleaning of the text should be done in this phase to obtain a plain text without messy characters. The output of this stage is a domain corpus that allows further parsing and manipulating.

Phase 2 is the core stage for constructing the domain ontology. Based on the well-known ontology learning layer cake (Cimiano, 2006), activities are designed to obtain the elements for an ontology. This phase contains eight activities according to the eight layers of the composition of 'the cake'. After the completion of each activity in this phase, an ontology can be gradually assembled by the elements, such as domain vocabularies, concepts, taxonomic and non-taxonomic relations.

Phase 3 is the process of building the ontology by populating the instances into the ontology model. As the ontology models are built in Phase 2, they can be enriched by, for example, increasing the depth of the concept hierarchy, adding new non-taxonomic relations, or extracting more complicated concepts and instances. The outcome of this phase is a visualized domain ontology consisting of SE domain concepts and relations.

The last phase (Phase 4) is to evaluate the ontology generated from the former three phases. The gold standard evaluation method proposed by Zavitsanos *et al.* (2011) can be referenced to help evaluate the learned ontology.

2.4.3 Evolution of Ontology Learning Process

The concept of ontology learning is firstly proposed by Maedche and Staab (2001). They propose a looping model of the ontology learning process. The principal activities are importing and reusing existing ontologies, extraction of ontology models, pruning ontologies, refinement of ontologies, and validating the resulting ontology. Figure 2.4 illustrates the looping process. Jung (2004) agrees with the looping process and indicates that ideally, ontology learning has four main phases that are importing, extracting, pruning, and refining.

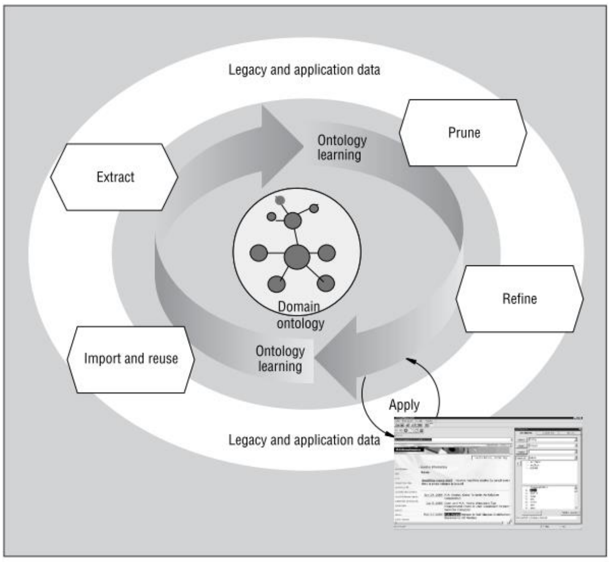


Figure 2.4 The ontology learning process model proposed by Maedche and Staab (2001)

Missikoff, Navigli, and Velardi (2002) propose a different model. They believe that the ontology learning architecture should contain three phases: terminology extraction and filtering, semantic interpretation, and domain concept forest generation. The three-phase process is illustrated in Figure 2.5.

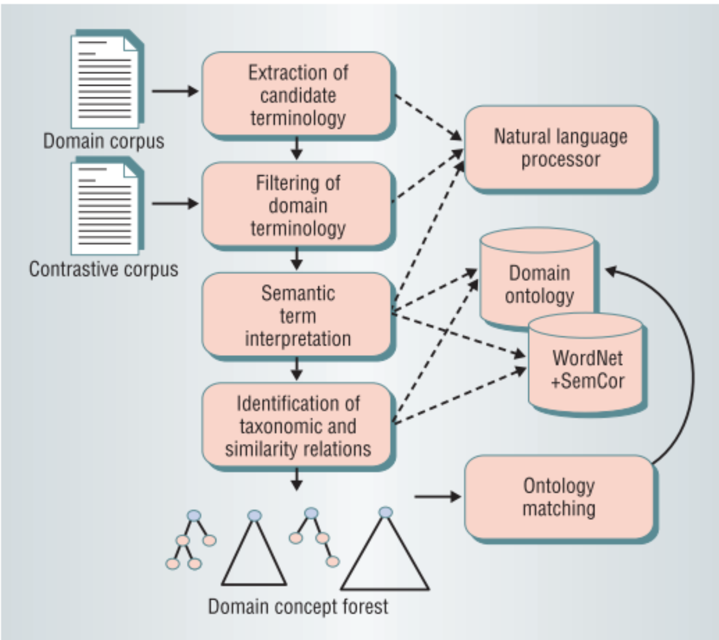


Figure 2.5 The ontology learning process model proposed by Missikoff, Navigli, and Velardi (2002)

These two models are contemporary. However, their focus is different. The former emphasizes the use of legacy data, and through a looping process, the data are pruned and refined. The later begins with a domain corpus and highlights the role of NLP. It is closer to the methodology and thinking of the current ontology learning approach, seeing the process through a perspective of extracting the components of an ontology, i.e., terminology, concepts, taxonomic relations.

Later on, Lee, Na, and Khoo (2003) propose a very similar model, as shown in Figure 2.6. However, it is mainly suitable for medical ontology learning. It can be seen as an application of the Missikoff, Navigli, and Velardi (2002)'s model.

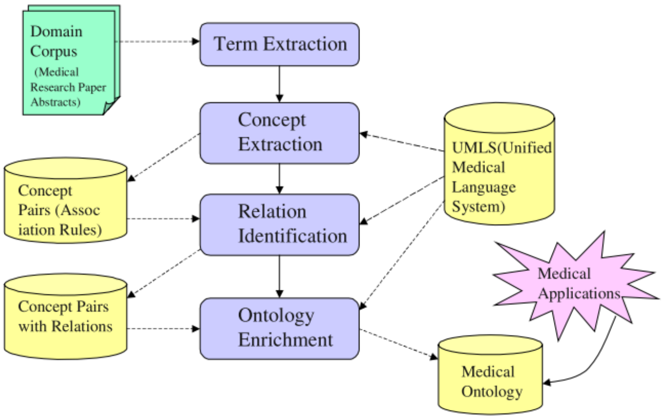


Figure 2.6 The ontology learning process model proposed by Lee, Na, and Khoo (2003)

The most widely known and accepted definition of the ontology learning process is the development of the 'ontology learning layered cake'. Cimiano (2006) argues that the ontological primitives that comprise the components of an ontology can be seen as a cake (as shown in Figure 2.7) From the bottom (Term) to the top (General Axiom), the abstract degree progressively arises, and the complexity of the learning process gradually increases (Asim *et al.*, 2018). Therefore, the course to obtain each of the layers of the cake naturally becomes the ontology learning process.

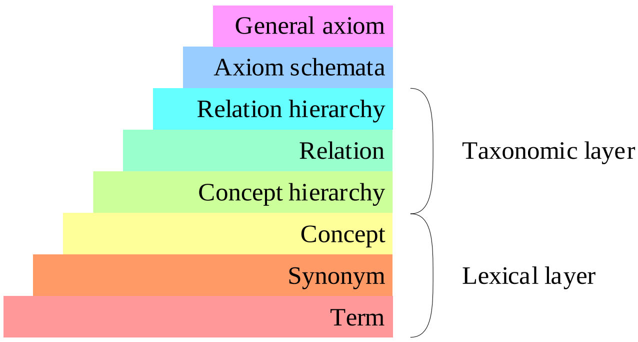


Figure 2.7 Ontology learning layer cake (adapted from Cimiano (2006))

A noticeable improvement can be seen in Kong (2007). The author combines the above models and generates an ontology learning framework dedicated to information organization and knowledge discovery. Figure 2.8 illustrates the framework. It can be seen that ML and NLP start to become the core of ontology learning. Also, this model clearly defines five phases of the ontology learning process, text processing, concepts extracting, relations extracting, formal representing, and application and evaluation.

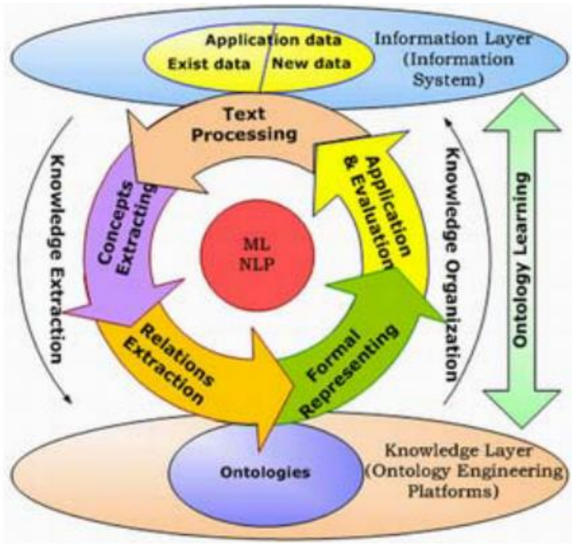


Figure 2.8 The ontology learning process model proposed by Kong (2007)

Sooner after, Gacitua, Sawyer, and Rayson (2008) emphasize the role of semantic annotation of a domain corpus, which is not mentioned in the other models. They define the ontology learning process into four phases: part-of-speech and semantic annotation of a corpus, extraction of concepts, domain ontology construction, and domain ontology edition. Figure 2.9 illustrates the proposed process.

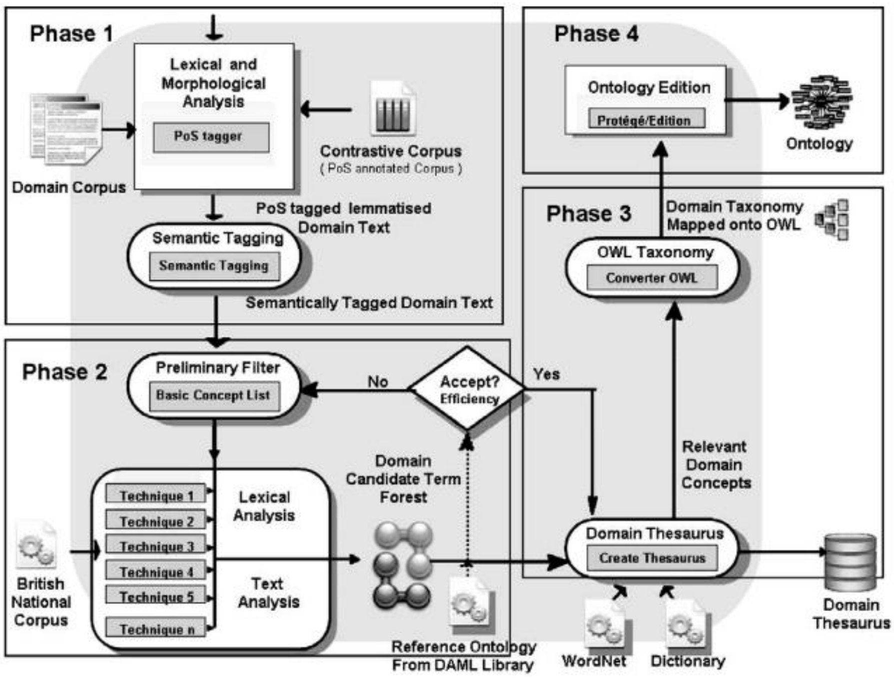


Figure 2.9 The ontology learning process model proposed by Gacitua, Sawyer, and Rayson (2008)

Simperl, Tempich, and Vrandečić (2008) generate a more comprehensive model of the whole ontology learning process. This model is based on the traditional ontology engineering process but added with learning-driven features. Figure 2.10 illustrates the model. It contains eight phases: feasibility study, requirements specification, selection of information sources, selection of ontology learning methods and tools, learning preparation, learning execution, ontology evaluation, and ontology integration.

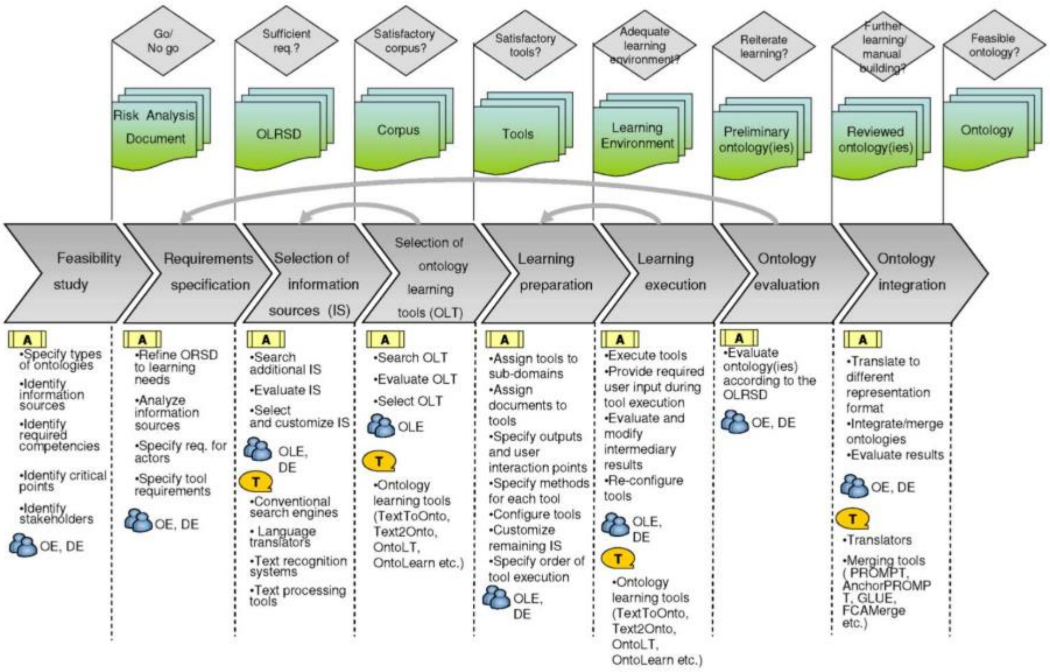


Figure 2.10 The ontology learning process model proposed by Simperl, Tempich, and Vrandečić (2008)

Villaverde *et al.* (2009) emphatically explore the process of discovering and labeling relationships of the entire ontology learning process. Figure 2.11 illustrates the model.

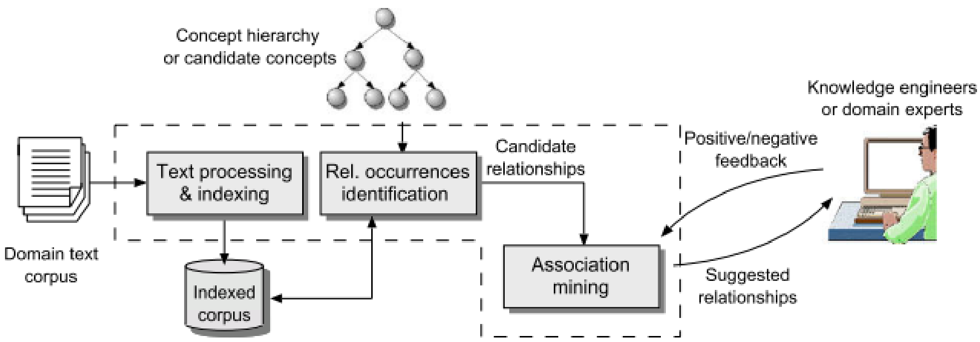


Figure 2.11 The ontology learning process model proposed by Villaverde *et al.* (2009)

Through the review of extant ontology learning process models, the following observations can be obtained.

From a methodological perspective, there is no detailed methodology for guiding the ontology learning process, regardless of the source used for learning or the method considered. Only methods that provide general guidelines exist, and they need to interact with users to achieve their goals. Also, there is no complete correspondence between the methods of ontology learning and the tools developed. The proposed method for learning ontologies from text is mainly based on natural language analysis technology, supplemented by statistical means. Such techniques are used to derive new concepts or relationships from selected sources. All of these methods require an ontologist to assess the accuracy of the final ontology and learning process.

From a technical point of view, it is possible to draw the following conclusions. There are no fully automated tools to perform the learning process. Some tools focus on helping to acquire semantic knowledge of vocabulary, while other tools help draw concepts or relationships from the preprocessing corpus. Also, no tool can evaluate the accuracy of the learning process, nor can it compare the different results obtained using different learning techniques. Therefore, the participation of ontology experts is required to evaluate the final ontology. Further ontology learning should benchmark ontology learning tools to measure their performance relative to a standard and compare similar processes in different situations.

Ontology learning is a suitable process to accelerate the knowledge acquisition activity of the ontology development process. It can be useful for building an ontology from scratch, reusing an existing one, or speeding up the construction of ontologies to be used for different purposes. However, the aim of automatic building an ontology is far from being achieved.

2.4.4 Ontology Learning Methods and Tools

When implementing the ontology learning approach, a lot of statistical, linguistic, and logic methods are used to enable the process. Figure 2.12 shows the architecture of the learning process for deriving the ontology, with a map between the adopted methods and the deliverables or outputs (Yang *et al.*, 2019b).

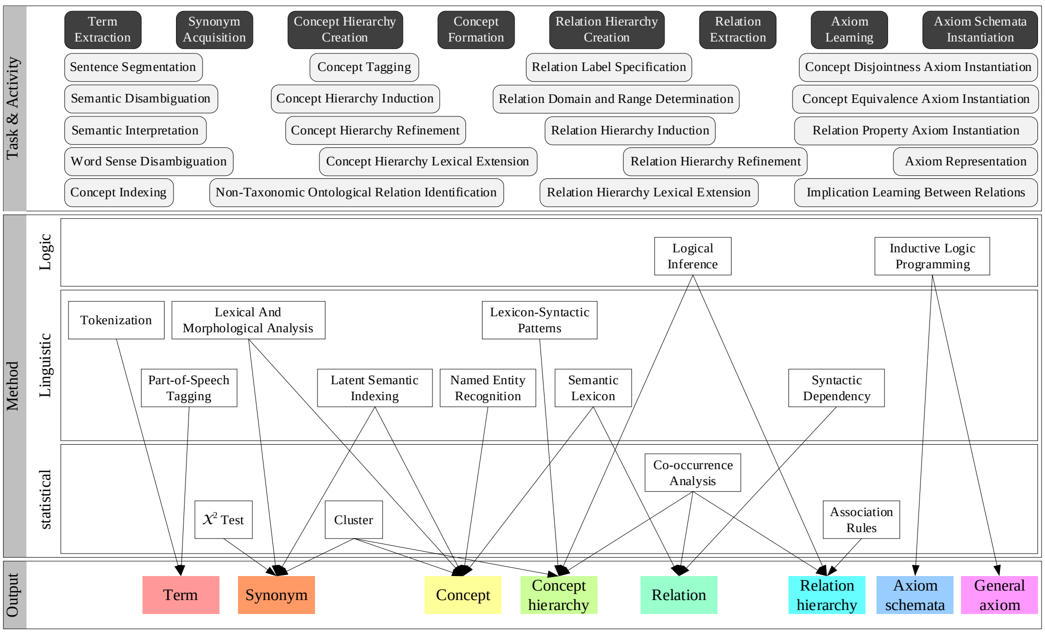


Figure 2.12 A mapping of activities, methods, and outputs in the ontology learning approach

The ontology learning approach is applied to the development of ontologies. So far, four phases are identified, i.e., pre-processing, ontology modeling, ontology population, and ontology evaluation. Eight tasks are defined based on the ontology layered cake, namely term extraction, synonym acquisition, concept hierarchy creation, concept formation, relation hierarchy creation, relation extraction, axiom learning, and axiom schemata instantiation.

The following sections briefly introduce each of the methods as they provide technological support for this research.

2.4.4.1 Statistical and Linguistic Methods

Ontology learning defines a set of methods and techniques for the fundamental development of new ontologies, for extension or adaption of already existing ontologies in an automatic way from various resources. Nine fundamental statistical and linguistic methods have been identified, and they are briefly described in the following paragraphs.

(1) Tokenization. A stream of text can be broken up into words, symbols, phrases, or other meaningful elements, i.e., tokens. This process is called tokenization. It is widely used to explore words in a sentence (Verma *et al.*, 2014). Without tokenization, large textual documents cannot be interpreted by computers. In NLP, it is required that information can be retrieved by words of the data set. Therefore, a parser is needed to process the tokenization of the text. It allows text to be stored in machine-readable formats. However, there are still some problems that have been left, e.g., punctuation marks must be removed, as well as other characters like brackets, hyphens, etc. The main use of tokenization is the identification of meaningful keywords. Another problem is abbreviations and acronyms, which need to be transformed into a standard form.

(2) Part-of-Speech Tagging. Following tokenization, Part-of-Speech Tagging aims at labeling each token with a unique tag that indicates its syntactic role, for example, plural noun, adverb, etc. The trending topics in Part-of-Speech Tagging are to train a classifier to tag a tokenized text automatically and achieve high accuracy. The best Part-of-Speech classifiers or taggers are based on classifiers trained on windows of text, which are then fed to a bidirectional decoding algorithm during inference (Verma *et al.*, 2014). Features include preceding and following tag context, multiple words context, and handcrafted features to deal with unknown words.

(3) χ^2 Test. The Pearson's χ^2 test, or just χ^2 test, is named after the mathematician Karl Pearson. It is also called a 'goodness of fit' statistic. In the NLP area, it is often used for feature selection. Feature selection is a very important step, as the selected feature words directly affect the accuracy of the classifier (Yao *et al.*, 2017). To check if there some common features between two categorical variables or not, this test can be useful in merging the synonyms among the feature words so that the dimension of feature space can be reduced.

(4) Lexical and Morphological Analysis. Sentence comprehension essentially relies on processes that recapture semantic and syntactic information from words and morphemes. However, lexical information is sometimes encoded incoherently across the word forms (Allen *et al.*, 2003). Usually, a word is composed of two recognizable morphemes: the stem, which encodes the content of the word (i.e., its meaning, grammatical category, complement options, etc.) and the suffix, which denotes the inflectional feature. In contrast, there are other words that cannot be decomposed into transparent morphemic constituents. Instead, a single form jointly encodes the content of the stem and the inflectional feature. Because regularly inflected forms are transparently compositional at the form level, the NLP task might recruit two autonomous access mechanisms to handle such forms—one that is dedicated to processing the lexical content of stems, and the other dedicated to extracting syntactic feature values from affixes. The potential benefit of such an arrangement is that the processor could identify the core lexical properties associated with a stem independently of the outcome of processes that compute is particular inflectional value. Based on these considerations, it would seem advantageous for the language processor to store and access all regularly inflected forms in a decompositional manner, even those forms that occur frequently enough to motivate encoding as whole-word units potentially. Performing a lexical and morphological analysis on text is very useful for synonym acquisition and concept formation.

(5) Latent Semantic Indexing. Latent Semantic Indexing is based on the vector-retrieval method in which predefined relationships between terms are modeled. The advantages of using the Latent Semantic Indexing lie in the fact that it allows semantic querying and respects the interrelatedness of the terms within a document (M. Rani *et al.*, 2017). It also measures the similarity of context and creates a reduced dimension feature-space representation.

(6) Cluster. Clustering can ensure partitions contain only semantically correlated data and are able to detect outliers when developing the concept hierarchy of ontologies. The cluster depends on the semantic similarity, or, more generally, relatedness between two terms or concepts. There is at least one maximal concept in a concept cluster (Zhou *et al.*, 2007). The concept clustering process is carried out between the parent and child concepts.

(7) Named Entity Recognition. Named Entity Recognition deals with identifying and classifying texts into pre-defined ontological classes. It enables relevant instances of concepts to be found. It includes two phases: detection of names, and classification of the names by the type of entity they refer to (Cimiano and Völker, 2005). The first phase is to ensure that names are defined to be contiguous spans of tokens, with no nesting. This segmentation is similar to chunking. The second phase requires choosing an ontology by which to organize categories of things. For this research, the emphasis is placed on the first phase.

(8) Lexicon-Syntactic Patterns. There has been considerable work conducted in regard to pattern-based extraction of ontological information. Lexico-syntactic patterns can model various semantic relations, although hyponymy seems to yield the most accurate results (K. Liu *et al.*, 2011). Moreover, they have the advantage of a frequent occurrence across many different text genres, and a reasonable overall accuracy even with little or no pre-encoded knowledge.

(9) Semantic Lexicon. A semantic lexicon is a digital dictionary of words labeled with semantic classes, so associations can be drawn between words that have not previously been encountered. Semantic lexicons are built upon semantic networks, which represent the semantic relations between words.

2.4.4.2 Natural Language Processing (NLP) Tools

Manually processing large amounts of information is time-consuming, repetitive, and hard to scale. Fortunately, Natural Language Processing (NLP) tools can help discover valuable insights in unstructured text and solve a variety of text analysis problems, like sentiment analysis, topic classification, and more. NLP is a discipline within artificial intelligence that leverages linguistics and computer science to make human language intelligible to machines. By allowing computers to analyze massive sets of data automatically, NLP tools can help find meaningful information very efferently. The most commonly used tools are the Natural Language Processing Toolkit (NLTK) and the Stanford CoreNLP.

(1) Natural Language Processing Toolkit. The Natural Language Toolkit, or commonly referred to as NLTK, is a suite of libraries and programs for symbolic and statistical NLP for English, written in the Python programming language (Bird and Loper, 2020). NLTK is very supportive to research in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. It has been used in many universities for teaching and research all over the world. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning, and many other functionalities.

(2) Stanford CoreNLP. Stanford CoreNLP provides a set of human language technology tools, including the part-of-speech tagger, the named entity recognizer, the parser, the coreference resolution system, sentiment analysis, bootstrapped pattern learning, and the open information extraction tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations

between entity mentions, and get the quotes people said (Manning *et al.*, 2015). Moreover, an annotator pipeline can include additional custom or third-party annotators.

2.5 Ontology Modeling

This section introduces the modeling methods and tools that are used in this research to develop the SE ontologies.

2.5.1 Logic Modeling Methods

Knowledge extraction methods range from statistical techniques to logical techniques. Logical methods are also used to extract ontological knowledge from the input. Logic-based learning methods may discover new knowledge by deduction or induction and represent knowledge by propositions, first-order, or higher-order logic. Five logic modeling methods are introduced as they are used in this research.

(1) Logical Inference. A semantic reasoner, reasoning engine, rules engine, or simply a reasoner is a piece of software that is able to infer logical consequences from a set of asserted facts or axioms in an ontology. The inference rules are commonly specified using an ontology language, and often a description logic language. Many reasoners use first-order predicate logic to perform reasoning. It is also useful for evaluating the ontology in terms of whether its logic is meticulous. In most of the ontology editing tools, a reasoner is embedded and should be performed during the construction of the ontology to evaluate if the classification is correct or the relationships are valid.

(2) Co-occurrence Analysis. Co-occurrence analysis assumes that two semantically related terms regularly co-occur in the same text segments (Liu *et al.*, 2005). Therefore, this method is used for enriching both the concept and relation hierarchy. A small set of terms from domain experts or from known ontology repositories is first selected as a seed ontology. Then, the seed ontology terms are through a lexical and morphological analysis. Terms are selected according to a threshold value on the co-occurrence significance.

(3) Syntactic Dependency. A syntactic dependency is a relation between two words in a sentence with one word being the governor and the other being the dependent of the relation. Syntactic dependencies often form a tree.

(4) Association Rules. Association rules are expressions of the type “if antecedent then consequent” (David *et al.*, 2006), representing implicative tendencies between conjunctions of attributes in ontology queries. Association rules are based on the notion of a transaction, which is an observation of the co-occurrence of a set of items (Nebot and Berlanga, 2012). This is

basically a set-based representation of the world, which contrasts with the numerical vector-based representations used in clustering and classification.

(5) Inductive Logic Programming. Prior conceptual knowledge is a core ingredient in inductive logic programming that was born at the intersection of concept learning and logic programming (Lisi and Esposito, 2009). Thus, it has been historically concerned with rule induction with the aim of prediction.

2.5.2 Ontology Modeling Tools

Tools are essential to aid an ontologist in constructing an ontology and merging multiple ontologies. Such conceptual models are often complex and multi-dimensional that are difficult to manage. The ontology modeling tools also usually contain mechanisms for visualizing and checking the resulting model, over and above the logical means for checking the satisfiability of the specified model. Therefore, Protégé, WebProtégé, and WebVOWL are selected to develop SE ontologies. They are essential for maintaining the complex SE ontologies that are necessary for capturing the SE body of knowledge.

(1) Protégé. Protégé Desktop is a feature-rich ontology editing environment with full support for OWL (Tudorache, 2019). It has direct in-memory connections to description logic reasoners. It supports the creation and editing of ontologies in a single workspace via a completely customizable user interface. The visualization tools allow for the interactive navigation of ontology relationships. The advanced explanation support contributes to tracking down inconsistencies. It is available from https://protege.stanford.edu/.

(2) WebProtégé. WebProtégé is another powerful tool developed by Stanford University for collaborative ontology development (Tudorache *et al.*, 2013). It provides the following additional features. It provides a full change tracking and revision history. Collaboration tools such as sharing and permissions, threaded notes and discussions, watches, and email notifications are embedded. Multiple formats are supported for uploading and downloading of ontologies (supported formats: RDF/XML, Turtle, OWL/XML, OBO, and others). It is accessible from http://webprotege.stanford.edu/.

(3) WebVOWL. WebVOWL is primarily used for ontology visualization in this study. It a web-based visual ontology modeling application that is independent of a particular device and interaction context (Wiens *et al.*, 2018). Ontologies are visualized using the VOWL notation, which is a well-specified visual language for the user-oriented representation of OWL ontologies.

2.6 Process Modeling

There are a lot of methods and tools available for process modeling, such as the business process model and notation, unified modeling language, flowchart techniques, and data flow diagrams. For this research, the IDEF methods are chosen since the IDEF0 modeling method is very suitable for presenting the life cycle processes in the case study. IDEF methods are a suite or family of methods that support a paradigm capable of addressing the modeling needs of systems and software engineering fields. IDEF0 is the function modeling method that is designed to model the decisions, actions, and activities of an organization or system. IDEF5 is the ontology description capture method, which assists in creating, modifying, and maintaining ontologies.

2.7 Conclusion

This chapter provides a theoretical basis for the research in this thesis. It mainly includes five aspects, theoretical foundation relevant to SE, ontologies, ontology learning, and ontology modeling, and process modeling.

In theory relevant to SE, the definition of SE was discussed, followed by a brief introduction to the fundamentals of the system, such as systems science, systems thinking, systems approach, and systems praxis. The definition of system life cycle processes was presented to provide an introduction to the SE standards. In the end, an original classification of the SE knowledge areas was proposed, which identifies six groups of SE knowledge domains with 28 knowledge areas.

Regarding the theory related to ontologies, a synthesis of the definitions was provided first, followed by the ontology formalisms, terminological formalism (T-box), and assertional formalism (A-box). Then, different types of ontologies were discussed, clarifying what kind of ontologies this research aims to develop.

This chapter also provides all the relevant theoretical foundations for the ontology learning approach. The definitions of ontology learning were introduced. It also clarified the role of ontology learning in this research. The basic theory of the ontology learning layer cake model and the appropriate modeling methods and tools were discussed.

As this research aims at developing an SE ontology for the entire SE body of knowledge, the relevant theories about ontology modeling were presented in this chapter, focusing on the logic modeling methods and the modeling tools. The methods and tools were used in the following chapters.

Finally, a brief introduction to the process modeling was presented. The IDEF methods for modeling business and functional processes were introduced, as they were used to restructure the system life cycle processes.

To sum up, this chapter provides a theoretical foundation for the following research activities. It comprises five key aspects:

- systems engineering - the application field of this research,
- ontologies - the deliverables of this research,
- ontology learning - the methodology of this research,
- ontology modeling - the logical modeling methods and tools, and
- process modeling - the process modeling methods and tools.

The next chapter moves into a detailed and comprehensive literature review on the state of the art of OBSE.

CHAPTER 3 LITERATURE REVIEW

3.1 Introduction

In order to understand the state of the art of the development and implementation of ontologies in the SE domain, an SLR is conducted for analyzing the status quo and revealing the gaps. Given that an in-depth literature review can provide theoretical support for research objectives and directions, the following sections present the review process and review results in detail. The structure of the section is outlined as follows.

- Section 3.2 presents a systematic methodology for conducting the literature review.
- Section 3.3 states the review questions.
- Section 3.4 to Section 3.6 elaborates on the results of each review question using the collected literature.

3.2 Literature Review Methodology

This section presents the methodology that is used to conduct a state-of-the-art review on OBSE. It first introduces the definition of a systematic literature review (SLR). Then, the search strategy of retrieving cognate studies is discussed. Third, a 7-step SLR methodology designed for and adopted by this study is described in detail.

3.2.1 Systematic Literature Review (SLR)

A systematic literature review (SLR) is a means of identifying, assessing, and analyzing primary studies' results relevant to a research problem, topic area, or phenomenon (Kitchenham *et al.*, 2015). There are many SLR methodologies proposed in the literature, but the majority are designed for medical science (Tranfield *et al.*, 2003). Kitchenham and Charters (2007) adapt the work conducted in the medical domain and implement it in software engineering. As software engineering and SE are intimately intertwined disciplines, this review follows their guidelines and also refers to other SLR methods. The critical elements of a systematic review process are learned, such as framing questions, identification of relevant work, assessing quality, summarizing the evidence, and interpreting findings (Cronin *et al.*, 2008; Khan *et al.*, 2003). For electronic search strategies, Page (2008) is referenced.

3.2.2 Search Strategy

A search strategy is deployed to find publications systematically to retrieve primary studies as comprehensively as possible. As ontologies are the primary concern in this review, the search focuses on papers that develop ontologies for SE or use ontologies in SE. Additionally, the SE body of knowledge is broken down into specific areas. In Chapter 2, Figure 2.1 presents the breakdown structure. It contains six groups of SE knowledge areas, including systems fundamentals, representing systems with models, engineered system contexts, SE standards, generic life cycle stages, and SE management. Each group is subdivided into more specific knowledge areas, a total of 28. The literature review follows the structure of these SE knowledge areas to ensure a systematic and thorough retrieval and analysis of the existing literature. During the design phase of the literature review, the term ‘systems engineering’ acts as the broadest keyword in the search string and is combined with each knowledge area defined in the classification to narrow down the topics into concrete areas. The classification ensures a relatively comprehensive paper collection process in the early phase of the research. Various terminologies related to the same knowledge area are thoroughly considered during literature retrieval.

3.2.3 7-step SLR Methodology

A well-designed 7-step methodology is used to conduct the literature review, which is illustrated in Figure 3.1.

To begin with, papers were thoroughly retrieved according to all possible terminologies in mainstream electronic databases, including IEEE Xplore, Science Direct, Scopus, Web of Science, and Wiley Online Library. After eliminating the duplicated papers, more than 600 papers entered the second-round selection. In this round, according to the inclusion and exclusion criteria shown in Table 3.2, the abstracts of these papers were reviewed. The number was then reduced to 183 papers, including journal articles, conference papers, and book chapters. Next, the 183 papers were retrieved for full texts and carefully reviewed, leaving 116 for the final inclusion review. These papers were then evaluated to obtain relevant information for each research question.

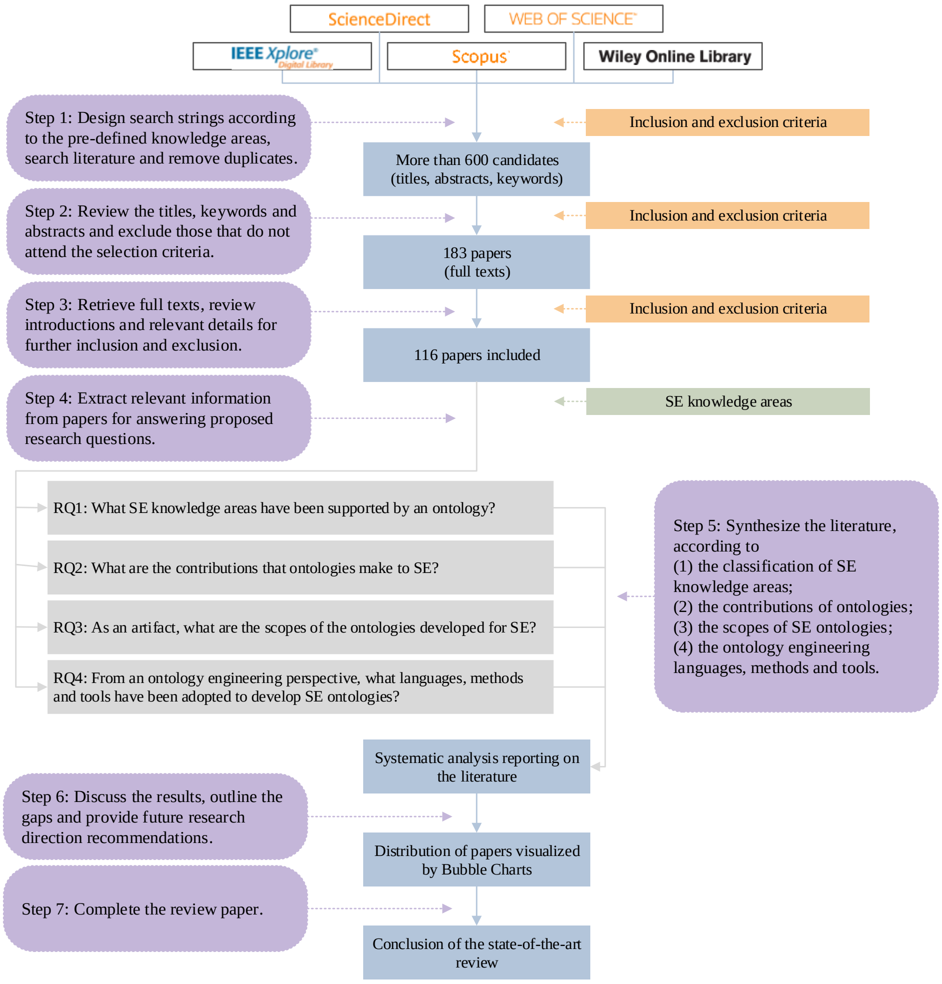


Figure 3.1 Literature Review Methodology

The next section elaborates on each question in more detail and explains the literature inclusion and exclusion criteria.

3.3 Review Questions

This state-of-the-art review aims to investigate *how* ontologies support SE and to ascertain *to what extent* they have been applied. So far, there is insufficient work that evaluates the values of ontologies to SE. In this study, four research questions were proposed to explore the state of the art of OBSE, as presented in Table 3.1.

Table 3.1 Review Questions

#	Review Questions	Descriptions and Motivations
RQ1	What are the SE knowledge areas supported by an ontology?	This question is designed to decompose SE into its fundamental aspects. The classification of the SE knowledge areas is defined in Figure 2.1.
RQ2	What are the contributions that ontologies make to SE?	This question aims to explore the effects of ontologies on SE.
RQ3	As an artifact, what are the scopes of the ontologies developed for SE?	Since ontologies can be final deliverables, this question is raised to summarize the existing ontologies for SE.

#	Review Questions	Descriptions and Motivations
RQ4	From an ontology engineering perspective, what languages, methods, and tools have been adopted to develop SE ontologies?	The answer to this question is to summarize the ontology engineering languages, methods, tools that are used to develop SE ontologies.

As shown by Table 3.1, this review aims to clarify *what*, *where*, and *how* ontologies are used in the field of SE, followed by *to what extent* they are used. As Figure 2.1 predefines the SE knowledge areas, it is reasonable to start the review by examining whether these areas are supported by ontologies, which yields RQ1 *where* ontologies are applied in the field of SE. Then, RQ2 follows to discern what roles ontologies play in SE and what benefits that ontologies bring - in other words, RQ2 illuminates *how* ontologies are used. RQ3 focuses on specific SE ontologies, which are computational artifacts, as RQ3 deals with *what* ontologies exist to date. Finally, RQ4 evaluates the existing SE ontologies from an ontology engineering perspective to establish *to what extent* ontology engineering techniques are applied in SE.

Table 3.2 lists the inclusion and exclusion criteria applied in the paper selection process to refine the review scope. The overarching research question is, how do ontologies support SE? However, during the review process, it is found that many papers discuss how SE approaches are applied to advance ontology design and development, contrary to what the review targets. Therefore, these papers were excluded. Papers were also excluded if they discuss Ontology in terms of philosophy. For example, Oliga (1988) clarifies the underlying metatheoretical assumptions for the foundations of systems methodologies. In this scenario, Ontology reflects the essence of things and phenomena, while this research only focuses on ontologies that are computational artifacts. Keating (2005) explores the foundations for the system of systems (SOS) engineering along the epistemological, ontological, methodological, application, and method domains for research. It is based on the philosophical underpinnings used to inform the epistemological and ontological perspective of SOS engineering, which is not what this study concerns. The subjective criteria (#3, #5, #6, #7 and #8) are detailed below for clarity.

Table 3.2 Inclusion and exclusion criteria

#	Inclusion criteria
1	Peer-reviewed papers
2	Primary studies
3	Studies that use an ontology to support SE
#	**Exclusion criteria**
4	Not written in English
5	Not describing ontologies for SE as the papers' main purpose
6	Falling out of the scope of the pre-defined knowledge areas
7	The search words match, but the actual semantics are inconsistent
8	Irrelevant to SE although it mentions some terms about SE
9	Full text not available

Inclusion criteria #3 states that studies that use an ontology to support SE should be included. This is a loose condition that enables, including as many papers as possible in the first-round selection (Step 2). The abstracts were reviewed to understand the main idea of a paper. However, it was found that many papers were not targeted at SE. For example, a well-cited paper by Wand and Weber (1990) was retrieved as its title is ‘an ontological model of an information system’, and the abstract contains terms such as subsystem and system decomposition. However, it was excluded in this review as it focuses on information systems rather than SE.

Based on the exclusion criteria, papers were also excluded for not building ontologies for SE as their main purpose (#5). For example, Derler *et al.* (2012) suggest that using domain-specific ontologies should enhance modularity and prevent misconnected model components in cyber-physical systems (CPS). However, describing the ontology is not the main topic of this paper; thus, it was excluded.

Criterion #6 refers to the pre-defined knowledge areas in Figure 2.1 and clarifies the scope of this review. The pre-defined classification was designed to answer specific research questions in this research. However, it is not equivalent to the entire universe of SE body of knowledge. Many papers were excluded when they fell outside the pre-defined knowledge areas. This exclusion criterion led to the omission of much work from other SE knowledge areas, such as the agile system, multi-agent system, enabling system, acquisition process, and so on.

Criterion #7 describes the situation where a paper matched the searching strings, but the matched terms have different semantics. For example, papers were come across that discuss ontologies for ‘process systems engineering’ (Dombayci *et al.*, 2015; Trokanas and Cecelja, 2016). However, ‘process systems engineering’ is not one of the SE processes.

Finally, criterion #8 excludes the papers that are irrelevant to SE, despite mentioning some SE related terms.

To sum up, the classification in Figure 2.1 provides a comprehensive index for classifying the SE knowledge areas. This classification, along with the inclusion and exclusion criteria in Table 3.2, ensure that the papers reviewed are within the scope of the research topic. The 7-step review methodology in Figure 3.1 plays an essential role in analyzing the research gaps and identifying future research directions.

In the following sections, findings from the 116 selected papers are presented. Each research question is addressed separately in an independent section.

3.4 Ontologies in Various SE Knowledge Areas (RQ1)

RQ1 focuses on identifying which SE knowledge areas are supported by an ontology. It is too abstract to discuss how ontologies are applied in SE in general. It is also not possible to answer RQ1 if there is no agreed classification about the SE body of knowledge. Therefore, RQ1 is answered based on the original classification of the SE knowledge areas demonstrated in Figure 2.1. The six groups of knowledge areas are shown on the left of Figure 3.2, which shows a visualization of the papers' distribution against the classification.

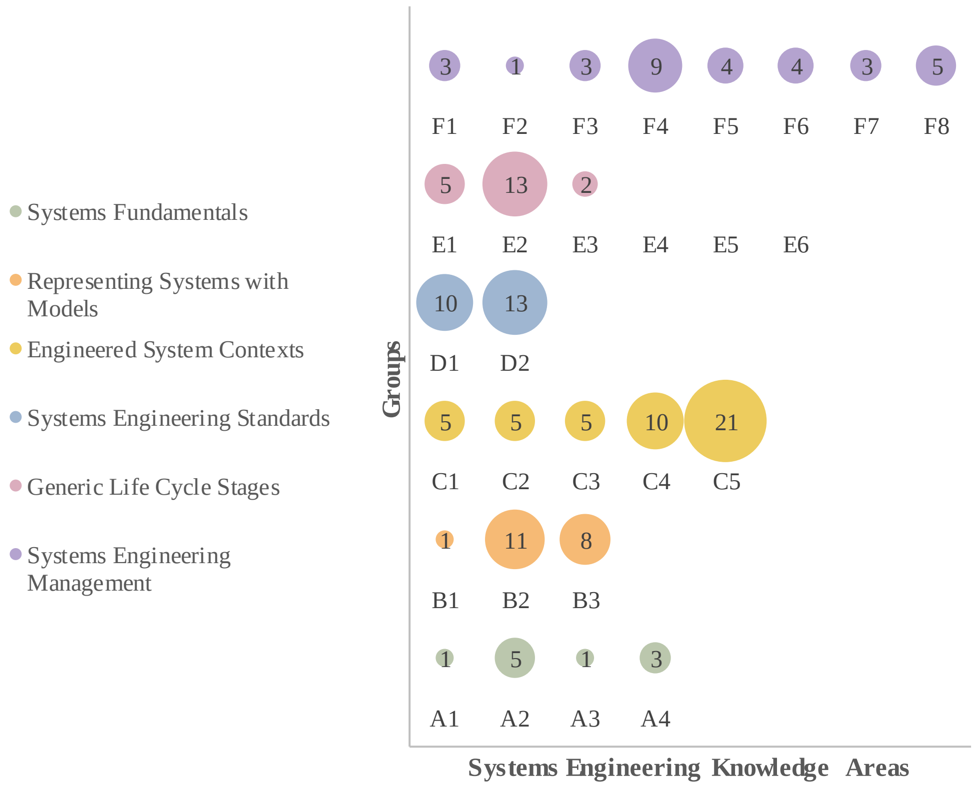


Figure 3.2 Distribution of papers for SE knowledge areas

The horizontal axes in Figure 3.2 are the codes of the knowledge areas: systems fundamentals (A1 to A4), representing systems with models (B1 to B3), engineered system context (C1 to C5), SE standards (D1 to D2), generic life cycle stages (E1 to E6), and SE management (F1 to F8). The correspondence between codes and the knowledge areas can also be found in Figure 2.1. The magnitude of the 'bubbles' correlates to the numbers of papers. Note that one paper can satisfy more than one knowledge area so that it will be counted more than once.

Table 3.3 summarizes all the papers that are included in this review with their classifications in the knowledge areas. Because of the volume of the literature, Table 3.3 provides an index of all the 116 papers considered in the review with their most relevant topics in the 28 knowledge areas. It is worth noting that one paper can satisfy more than one knowledge area. Based on

Figure 3.2, it is easy to see which knowledge areas are the hot spots of ontologies research and which knowledge fields are relatively deficient in ontologies research. For each knowledge area, the relevant literature is detailed reviewed and synthesized, in the following six subsections after Table 3.3.

Table 3.3 An index of the references mapped to the corresponding knowledge areas

Knowledge areas	**A1. System**	**A2. Behavior**	**A3. Complexity**	**A4. Emergence**
References	Dori and Sillitto (2017)	Easterbrook (2014) Mason (2005) Kaderka *et al*. (2018) Borgo *et al*. (2009) Gero and Kannengiesser (2007)	Jacobson *et al*. (2011)	Schmid (2009) Langford and Langford (2017) He *et al*. (2014)
Knowledge areas	**B1. Model**	**B2. MBSE**	**B3. Modeling language**	**C1. Product systems**
References	Sánchez *et al*. (2009)	Dori (2016) Herzig *et al*. (2011) Orellana and Madni (2014) Hoppe *et al*. (2017a) Shani (2017) Ernadote (2015, 2017) Hennig *et al*. (2016) Bermejo-Alonso *et al*. (2016) Madni and Sievers (2018a, 2018b)	Dori (2002, 2016) Mezhuyev (2014) Guizzardi *et al*. (2015) Figueiredo *et al*. (2018) Al-Fedaghi (2015) Mandutianu *et al*. (2009) Wagner *et al*. (2012)	Ball and Runge (2014) Borgo and Leitão (2007) Annamalai *et al*. (2011) Rese *et al*. (2013) El Kadiri and Kiritsis (2015)
Knowledge areas	**C2. Service systems**	**C3. Enterprise systems**	**C4. SOS**	**C5. CPS**
References	Nardi *et al*. (2015) Dong *et al*. (2011) Gonsalves and Itoh (2008) Lemey and Poels (2011) Miao and Sun (2006)	Ahmad *et al*. (2011) Green *et al*. (2005) Weichhart *et al*. (2016) Roche (2000) Lee and Goodwin (2006)	Langford and Langford (2017) Ormrod *et al*. (2015) Benali *et al*. (2014) He *et al*. (2014) Dogan *et al*. (2014, 2012) Ferreira and Tejeda (2011) Samhan *et al*. (2016) Zhu *et al*. (2017) Madni and Sievers (2014)	Trappey *et al*. (2018) Jeong *et al*. (2018) Ma *et al*. (2017) Vanherpen *et al*. (2016) Hildebrandt *et al*. (2018) Wan *et al*. (2018) Torsleff *et al*. (2018) Daun *et al*. (2016) Brings *et al*. (2018) Balduccini *et al*. (2018)

Knowledge areas	C2. Service systems	C3. Enterprise systems	C4. SOS	C5. CPS
				Lynch *et al*. (2016, 2017) Schmit *et al*. (2016) Petnga and Austin (2016, 2013, 2015) Sartakov (2015) Teslya and Ryabchikov (2018) Eskins and Sanders (2011) Herrera *et al*. (2013) Ali and Hong (2018)
Knowledge areas	**D1. Modeling standard**	**D2. Related standard**	**E1. Concept stage**	**E2. Development stage**
References	Dori and Reinhartz-Berger (2003) ISO/PAS International Standard (2015) Dori *et al*. (2018) Lopez-Lorca *et al*. (2011) Madni *et al*. (2001, 1998) Sarder and Ferreira (2007) Hoppe *et al*. (2017b) Chourabi *et al*. (2010) Triantis and Collopy (2014) Aslaksen *et al*. (2011)	van Ruijven (2013, 2015) Yang *et al*. (2017) Agrawal (2016) Blokland and Reniers (2018) Eito-Brun (2016) Henderson-Sellers *et al*. (2014) Gonzalez-Perez *et al*. (2016) Guessi *et al*. (2015) Martin *et al*. (2017) Roldán *et al*. (2018) Ferchichi *et al*. (2008) Pardo-Calvache *et al*. (2014)	Bergholtz and Eriksson (2015) Pfaff and Krcmar (2018) Silega *et al*. (2016) Ryan and Wheatcraft (2017) Fraga and Llorens (2015)	Hallberg *et al*. (2014) Miller (2017) Hatchuel *et al*. (2013) Sim and Duffy (2003) Sarder *et al*. (2007) Christophe *et al*. (2009) Ryan *et al*. (2013) Witherell *et al*. (2007) Vanherpen *et al*. (2016) Cruz *et al*. (2018) Fraga and Llorens (2015) Guessi *et al*. (2015) Martin *et al*. (2017) Roldán *et al*. (2018)
Knowledge areas	**E3. Production stage**	**E4. Utilization stage**	**E5. Support Stage**	**E6. Retirement Stage**
References	Madni and Sievers (2014a, 2014b)	N/A	N/A	N/A

Knowledge areas	F1. Planning	F2. Assessment and control	F3. Decision management	F4. Risk management
References	Bouras *et al.* (2016) Lee *et al.* (2008) Líska and Návrat (2010)	Hahn *et al.* (2008)	Wulandari *et al.* (2018) Cruz *et al.* (2018) Gorshkov *et al.* (2016)	Tserng *et al.* (2009) Nota *et al.* (2010) Lykourentzou *et al.* (2011) Ansaldi *et al.* (2012) Birkholz *et al.* (2012) Jiang and Zhang (2013) Guo and Nunes (2009) Agrawal (2016) Blokland and Reniers (2018)
Knowledge areas	**F5. Configuration management**	**F6. Information management**	**F7. Measurement**	**F8. Quality management**
References	Zhang (2014) Eito-Brun (2018) Dong *et al.* (2011) Samhan *et al.* (2016)	Blanco *et al.* (2008) Wimalasuriya and Dou (2010) Grubic and Fan (2010) Mikroyannidis and Theodoulidis (2010)	Honour and Valerdi (2014) Bertoa *et al.* (2006) Kim and Fox (2002)	Fraga and Llorens (2015) Geisler *et al.* (2016) Kim and Fox (2002) Ferchichi *et al.* (2008) Pardo-Calvache *et al.* (2014)

3.4.1 Systems Fundamentals

The first group of the SE knowledge areas deals with systems fundamentals. An analysis of the relevant papers reveals that ontologies are used to support key SE concepts, such as system, behavior, complexity, and emergence. The following subsections discuss the relevant literature regarding the application of ontologies that supports defining the four SE concepts.

A1. System

The study of what a system means to the SE domain has been continuously debated. Among them, Dori and Sillitto (2017) establish an integrative ontological framework to classify and map over 100 definitions of ‘system’. They conclude that one single definition of a system cannot be both precise enough, to be useful, and general enough to satisfy the widest possible range of systems community. This ontological framework, as a system typology, lays a foundation for achieving a widely accepted family of definitions of ‘system’.

A2. Behavior

“A system's behavior is a system event(s) which is either necessary or sufficient for another event in that system or its environment” (Ackoff, 1971). Systems thinking is a critical strength in the understanding of transformational changes that are needed to achieve sustainability. Therefore, key concepts for sustainability are discussed by Easterbrook (2014) for understanding and reasoning about system behavior. Mason (2005) defines a set of properties, especially for organizational behaviors. Furthermore, Kaderka *et al.* (2018) develop a tool to allow engineers to specify system and component behaviors. This tool relies on an underlying ontology that includes core concepts about behavior and scenario.

Borgo *et al.* (2009) define the meanings of behavior and function of technical artifacts in a uniform and rigorous foundational ontology. They assess five meanings of artifact behavior and the two meanings of function by incorporating these engineering notions in the descriptive ontology for linguistic and cognitive engineering (DOLCE), concluding that “two meanings of artifact functions, namely, device-centric and environment-centric functions, can be captured in DOLCE via the concepts of behavioral constraint and mode of deployment of an artifact.”

Gero and Kannengiesser (2007) propose a function-behavior-structure (FBS) ontology and apply the FBS ontology to classify processes according to the FBS view of objects and processes. Integrating function and behavior in a process ontology is useful for knowledge representations of processes, as they add semantics in a purposive context to generate, compare, and execute specific processes.

A3. Complexity

Jacobson *et al.* (2011) report an experiment of students learning core conceptual perspectives on system complexity. They find that students who are provided with an ontology about the characteristics of complex systems perform at a significantly higher level on problem-solving tasks.

A4. Emergence

As defined by Wheeler and Checkland (2000), emergence is "the principle that whole entities exhibit properties which are meaningful only when attributed to the whole, not to its parts." Schmid (2009) proposes a review of ontological and epistemological meanings of emergence, with a context of the particular role of computer simulation as an analytical tool for studying emergent properties and processes. Langford and Langford (2017) argue that a valid formal ontology can help expose the true nature of emergence. It provides flexible dimensionalities and a proper atomic form to allow integrations according to the rules of part-whole mereology. Therefore, relationships between structures can be captured, processes can be delineated, and every interaction between objects in both event-based and time-based contexts can be stipulated. Substantial transformation and accumulation of field knowledge comprised of SOS lead to emergence and uncertainty. He *et al.* (2014) analyze the emergence behavior mechanism of SOS and divide emergence into three levels: synthetic, application, and component. They also develop an SOS ontology to interpret and analyze the emergent properties.

3.4.2 Representing Systems with Models

This subsection reports the use of ontologies in representing systems with models. It begins with an ontology for the definition of a model and then discusses the application of ontologies in MBSE and their support in developing modeling languages.

B1. Model

The concept of 'model' is to be found in many different areas in the field of computing. An ontological definition of 'model' specific for SE cannot be found in the literature. However, the work conducted by Sánchez *et al.* (2009) is highly relevant to SE. They present an ontology of models in the field of information systems development, seeking to clarify and classify the heterogeneous terms concerning 'models'. The ontology tells the differences between schema, diagram, ontology, meta-model, pattern, architecture, architecture style, and process model.

B2. MBSE

MBSE refers to SE that is based on formal modeling of conceptual, mathematical, and physical elements (Rauzy and Haskins, 2019). Dori (2016) defines an object-process methodology (OPM) to represent the systems modeling paradigm. OPM is founded on a universal minimal ontology, which is very simple but rigorous. According to this ontology, objects exist, while processes transform them. The elements of the OPM ontology are entities (things and states) and links.

In the context of MBSE, models, created by differing formalisms and stakeholders with diverse views, need to remain consistent during the SE life cycle. Herzig *et al.* (2011) examine the fundamentals of consistency management by introducing a mathematical definition of consistency formally and then creating an ontology of inconsistency to reveal the different types. Apart from identifying the types, a distinction between internal and external consistencies is also compared. In this case, ontologies help to understand how inconsistency happens during the modeling. However, it is not captured in formal ontology modeling language, thus limiting its sharing and reuse.

Orellana and Madni (2014) take human factors into account to enhance MBSE. The critical issue is that people not trained in human factors are unable to communicate with those that are due to differences in terminology. Therefore, they propose a human-system integration ontology to provide new semantics extending current system modeling semantics. This ontology plays a vital role in better communications between all engineers, particularly, system architects, system engineers, and human specialty engineers.

Hoppe *et al.* (2017a) present an approach to trace the data among different digital models during the whole MBSE life cycle. The key technique of this approach is the definition of a holistic conceptual data model, which is enhanced by an ontology profile. It specifies required and non-admissible features for each SE phase and defines the specific behavior of each phase's characteristics. This results in the ability to have different views representing the level of abstraction needed for a certain stage on the same data structure. Shani (2017) uses ontologies to enable interoperability among MBSE tools and to allow model reuse to keep obsolete or legacy models alive. To be specific, ontologies become as a universal language to represent modeling data, allowing the models to withstand time through reuse among different tools and different languages as they evolve and progress.

Ernadote (2017) presents an approach to support MBSE, which combines the advantages of standard meta-models such as unified modeling language (UML) and systems modeling language (SysML), with dedicated project ontologies. It provides a solution to ease the communication between SE stakeholders by synthesizing standard meta-models and domain ontologies. Therefore, one has the flexibility to add, change, or remove concepts of interest using a specific meta-model while leaving the underlying project data unaffected. This solution contributes to the rapid creations of model-based documents.

In MBSE, a model specifying the system's design is shared across a variety of disciplines and used to ensure the consistency and quality of the overall design. Existing implementations for describing these system models exhibit several shortcomings regarding their approach to data management. Hennig *et al.* (2016) propose an MBSE ontology to solve this issue. The ontology provides increased semantic soundness of the underlying standardized data specification, enables reasoners to identify problems in the system, and allows the application of operational knowledge collected over previous projects. It is used to design a satellite system to validate and prove its value.

Autonomous SE is hampered by the lack of systematic methods to incorporate cognitive capabilities into systems. Bermejo-Alonso *et al.* (2016) capture the core concepts and relationships in the domain of the autonomous system, to enable the systematization of knowledge, and its use in the model-based engineering of autonomous systems. Formalized ontologies develop into exercisable models. They are enablers for modeling and used as core assets fueling the whole MBSE process.

Models with explicit agreement on basic terminology and relationships are key to successful MBSE. Ontologies, providing a uniform and consistent basis for representation and analysis, facilitate the creation of MBSE meta-models. Madni and Sievers (2018a) distinguish ontologies and meta-models in terms of their roles in advancing MBSE while affirm they are also an integral aspect of MBSE. Madni and Sievers (2018b) provide a long list of the rationale for constructing ontologies and justify how MBSE can benefit from these merits. Ontologies provide MBSE the capability of tracing model elements from an abstract model to progressively more specific models, and vice versa.

B3. Modeling language

As previously mentioned, OPM is both a language and a methodology (Dori, 2002). The language part is defined by the specification of its syntax, semantics, and ontology (Dori, 2016).

Mezhuyev (2014) proposes an ontology-based approach to develop domain-specific languages for SE. The meta-model level of the domain-specific language can be regarded as a SE ontology expanded by grammar rules and mathematical methods, which depict key concepts and their relationships in SE life cycle processes.

Guizzardi *et al.* (2015) discuss the development of the conceptual modeling language, OntoUML, the most successful application of the unified foundational ontology (UFO). They synthesize several methodological and computational tools, which have been developed over the years to support the OntoUML community. Figueiredo *et al.* (2018) present an approach to equip ontologically neutral modeling languages with real-world semantics. This approach can extract ontological views from conceptual models represented in the OntoUML. It makes systematic use of the real-world ontological semantics of OntoUML to propose a structure of views. On the one hand, the structure preserves all the information content of the original model;

on the other, the structure breaks down the information in different modules centered around different ontological concerns.

However, Al-Fedaghi (2015) argues that SysML achieves success as a modeling tool because of its multiplicity and fragmentation of representations, but fails to furnish a nucleus around which different phases of the engineering process evolve. While different views of the system are essential, there is still a need for an underlying specification that ties the different models together into a uniform conceptual picture. Therefore, he proposes a core ontology acting as a central reference in system specifications to improve SysML-based conceptual descriptions.

Nevertheless, Mandutianu *et al.* (2009) believe that although using a generic SE modeling language certainly helps increase understandability between systems stakeholders; it is not always enough. They prove that the understandability of modeling can be improved by using ontologies because they act as a formal language or a grammar which modelers can use to express their domain models. They compare the differences between SysML and OWL and conclude that SysML lacks formal semantics while OWL provides the logical mechanisms for semantic integration such that the meaning of the concepts is captured independently of the domain of interest and supports automated reasoning. A desirable solution is to combine the two languages in a coherent and meaningful way within the conceptual SE framework.

State analysis is a formal methodology that extends basic concepts from control theory and software architecture to aid in the design of complex control applications. To effectively apply this methodology with SysML tools, Wagner *et al.* (2012) develop ontological definitions of the concepts and relations in state analysis methodology to map to SysML and enforce structural constraints in a SysML model. The ontology for state analysis is developed in OWL2 via Protégé.

3.4.3 Engineered System Contexts

This subsection explores the application of ontologies in different engineered contexts. The SEBoK (BKCASE Editorial Board, 2017) suggests four: product, service, enterprise, and system of systems, while a fifth context, cyber-physical systems (CPS), is added because it is emerging as a critical area that benefits from ontologies.

C1. Product systems

Ball and Runge (2014) present the initial implementation of an ontology-based modeling and simulation system to support concurrent engineering, named “producing reusable engineered systems through ontology” (PRESTO). The PRESTO system contains a product ontology to enable the ability to leverage within the context of the product model.

ADAptive holonic COntrol aRchitecture for distributed manufacturing systems (ADACOR) is an agile and adaptive manufacturing control architecture. Borgo and Leitão (2007) develop a formal ontology for manufacturing scheduling and control environments by aligning the ADACOR architecture with a widely used foundational ontology, DOLCE. The ADACOR concepts, predicates, and attributes are analyzed for their ontological commitment and formalized in DOLCE, resulting in a core ontology of manufacturing. The ontology is suitable for developing adaptive and knowledge-based manufacturing processes, which is the key to the success of a manufacturing enterprise.

Annamalai *et al.* (2011) create a fundamental structure of a product-service systems ontology for providing an explicit formal specification of the terms in the domain and the relations amongst them. Through the ontology, the commonalities and differences of product-service systems between research groups and industries are understood, which helps industries to develop viable product-service systems by providing excellent communication between the stakeholders. Moreover, Rese *et al.* (2013) specifically focus on analyzing the definition of the business model in the context of product-service systems. They propose an ontology for characterizing and comparing various business models and provide a reference model for generating new business models.

Moreover, Kadiri and Kiritsis (2015) conduct a state-of-the-art literature review on ontologies in product life cycle management. Ontologies have seven key roles according to their study, namely, trusted source of knowledge, database, knowledgebase, the bridge for multiple domains, mediator for interoperability, contextual search enabler, and linked data enabler.

C2. Service systems

Nardi *et al.* (2015) propose a core ontology for services that address the notion of service in general, contributing to the harmony of different service perspectives.

As the structures of service products become increasingly complex, automatically configuring a customizable service product satisfying the customer's requirement becomes challenging. Dong *et al.* (2011) propose the use of ontologies for service product configuration. They develop a service ontology whose core concepts are service element, port, property, constraint, and resource. This ontology is used to model a customizable mobile service that shows merit in satisfying the customer's requirement.

Gonsalves and Itoh (2008) develop a performance ontology for service systems. The core of the performance ontology describes customers, resources, service and protocols, and the semantic relations among them. The ontology provides a framework for integrating the quantitative performance evaluation of service systems specified as queuing networks or as stochastic Petri nets. The goal of the domain ontology is to eliminate the conceptual and terminological confusion among the members of a virtual community of performance analysts and designers.

Lemey and Poels (2011) create an ontology of service systems worldview, which contains ten foundational concepts and their relationships in service science. They also map the foundational concepts to the concepts in other service theories and frameworks.

Miao and Sun (2006) discuss the role of ontologies in service-oriented systems development. They argue that ontologies can improve users' initial query refinement in the client's end.

C3. Enterprise systems

Ahmad *et al.* (2011) present an SLR of existing research in ontology-based knowledge management for enterprise systems. They identify the major activities in the enterprise systems life cycle and synthesize the definitions of the knowledge management process. Then, they link the knowledge management process and the enterprise system's life cycle. Among all the knowledge management research, they focus on reporting those based on ontologies.

Enterprise systems interoperability is currently an essential topic for business. Green *et al.* (2005) conduct an ontological evaluation of the constructs contained in the ebXML business process specification schema (BPSS). They map the constructs of the Bunge-Wand-Weber (BWW) ontological representation model with the ebXML constructs and analyze the shortcomings of ebXML. In ontological analyses, two types of analytical mapping are carried out: a representation mapping and an interpretation mapping. The study shows the usefulness of the ontological model for analyzing, evaluating, and engineering techniques in the areas of traditional and structured systems analysis, object-oriented modeling, and process modeling for enterprise systems. Moreover, Weichhart *et al.* (2016) define an ontology of enterprise interoperability whose core is based on systems science-related concepts and related properties. The ontology describes the underlying conceptualization of complex adaptive systems, composed of a structure of system elements, the relation between them, and interfaces through which interoperability occurs.

Concurrent engineering is based on the co-operation and collaboration of multi-disciplinary people who need to communicate and exchange information in enterprise systems. Roche (2000) proposes to use ontologies as an agreed vocabulary of standard terms and meanings shared within enterprise systems and develop a software environment, named ontological knowledge station (OK Station). This software is used to define the terminology used by the development staff and to define enterprise ontologies.

Ontologies could significantly reduce the costs of deploying, integrating, and maintaining enterprise systems. The barrier to the more widespread use of ontologies for such applications is the lack of support in the currently available middleware stacks used in enterprise computing. Lee and Goodwin (2006) describe several enterprise systems where ontology management can be useful, especially the SnoBase ontology management system.

C4. SOS

ISO/IEC/IEEE 15288 (ISO/IEC/IEEE International Standard, 2015) provides a definition of SOS: "An SOS brings together a set of systems for a task that none of the systems can accomplish on its own. Each constituent system keeps its management, goals, and resources while coordinating within the SOS and adapting to meet SOS goals." Langford and Langford (2017) argue that ontologies can be applied to determine taxonomy that illuminates the normative constructs of SOS. Ormrod *et al.* (2015) believe that efforts to model and trace the propagation of cyber effects across multiple distinct domains in an SOS require a specific ontology.

Benali *et al.* (2014) propose an approach to building an SOS interoperability conceptual model and a foundational ontology adapted from DOLCE to depict the SOS interoperability context. Based on the ontology, they propose a context-based SOS interoperability content ontology design pattern.

Capability is the ability to do something which has an overarching approach that links value, purpose, and solution of a systems problem. He *et al.* (2014) develop an SOS ontology to evaluate capability, while Dogan *et al.* (2012) present an approach to develop an ontology for capability engineering. A capability ontology is an enabler of semantic interoperability, and it supports a formal and explicit specification of a shared conceptualization for the concept of capability engineering. Dogan *et al.* (2014) also present another remarkable research. They create an SOS engineering thesaurus to ensure that concepts and terms are consistently interpreted, which can be a basis for an ontology.

Ferreira and Tejeda (2011) develop an ontology for the unmanned and autonomous system (UAS) of systems test and evaluation domain. The ontology assists those interested in the field of UAS SOS test and evaluation to understand the entities, relationships, and terminology within the domain.

Samhan *et al.* (2016) design a framework to address configuration management challenges in SOS engineering, in particular, change management issues. They identify four main stages to realize the framework, and ontologies are used in the first two stages to define a holistic specification of the change management business process of SOS engineering. However, the framework is still at a theoretical level, lacking the results of practice and, therefore, unable to prove its feasibility and validity.

Zhu *et al.* (2017) analyze the relations between mission and other concepts in SOS requirements modeling and present an ontology of the mission concept model. The ontology ensures SOS architects build mission models rigorously and analyze the mission model flexibly to make it expandable.

C5. CPS

"CPS are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa (Lee, 2008)." Trappey *et al.* (2018) depict a domain ontology to highlight technology and functions derived from a previous CPS domain ontology that shows the properties and the relations between key technology components. Jeong *et al.* (2018) propose an upper ontology that describes typical missions, functions, and interfaces of the subsystems to overcome heterogeneity inherent in automatic service composition for CPS.

The design of CPS involves many stakeholders when designing the system. Various stakeholders tend to express individual concerns specific to their views on the system under design. Despite their different views, they all relate to the design of the same system. Therefore, effective communication and consistency of design properties are critical to reducing errors. Ma *et al.* (2017) propose an ontology-based language, named OntoEvent, for semantic sophisticated event modeling and detection in CPS.

Moreover, Vanherpen *et al.* (2016) introduce ontological properties and their relations as the link between the view-specific properties used by the stakeholders to solve this problem. The view-specific properties or linguistic properties stemming from different semantic views can be related to each other through a shared ontology or a set of ontologies. These ontological properties and their relations with the linguistic world allow reasoning and tracing the view-specific properties that are needed for a specific design contract. By making the relations explicit, engineers can negotiate contracts with a common understanding of how properties are linked. Hildebrandt *et al.* (2018) propose a method tailored towards the needs of CPS to build ontologies since they can represent the information being shared among different stakeholders with open context. They demonstrate this method to build an ontology for communicating information between CPS about processes and machine states in the manufacturing domain. Wan *et al.* (2018) develop an ontology to describe the intelligent manufacturing resource to meet production requirements for fast iteration and to realize agile and efficient manufacturing CPS resource allocation from the perspective of resource utilization.

Torsleff *et al.* (2018) propose an ontology development approach. The aim is for modeling a coherent specification of openness and dynamicity in the structural, functional, and operational contexts across multiple systems. The ontologies also enable such collaborative CPS to perform context-related reasoning and exchange context-related information. Furthermore, Daun *et al.* (2016) and Brings *et al.* (2018) proposes a context ontology to cope with highly dynamic contexts of CPS by explicitly differentiating between not only the system and its context but also between the CPS network the system participates in, as well as the system network's context. They use the ontology to keep the different contexts of multiple CPS consistent with one other and the super system's specifications. Balduccini *et al.* (2018) develop an ontology for the trustworthiness aspect in CPS according to the CPS framework released by the US National Institute of Standards and Technology (NIST). The ontology makes it possible to reason about aspects and concerns of CPS, such as their interdependencies and the implications about the other systems.

Lynch *et al.* (2016) use ontologies to validate model integration and meta-model description in the CPS engineering domain. Lynch *et al.* (2017) implement an ontology-based approach to compress engineering development times and quickly prune the conceptual design space for CPS. They developed a lightweight ontology to facilitate a shift of the development focus to high-value design alternatives early in the SE life cycle. The ontology identifies relationships between design knowledge to component knowledge, enabling engineers to quickly see the results of design decisions in terms of existing components and products. The ontology is built by OWL and manipulated in Protégé. Furthermore, Schmit *et al.* (2016) use the same ontological structure to develop an ontology as a component model library to store and extract the information required for design space refinement. They present a case study of engineering a notional missile system, a kind of CPS, which proves this ontology's value.

Petnga and Austin (2013) propose a time-based reasoning framework for CPS, which contains an ontology that describes system behavior in terms of time, intervals of time, and relationships among intervals of time. Elsewhere, Petnga and Austin (2015) examine the role of a spatial ontology, which can formally represent spatial domain entities occurring in CPS. Furthermore, Petnga and Austin (2016) develop an ontological framework for knowledge modeling and decision support in CPS, named CPS-KMoDS. It relies on the composition of domain-specific ontologies along with corresponding knowledge bases on the one hand and domain-specific semantics extensions, an integrator, and the cyber-physical application on the other hand.

Sartakov (2015) uses an ontology to support the representation of networks in intrusion-detection systems, which prevent intrusion into networks of cyber-physical objects. The representation allows implementation both at the software level – comparing the movement of network traffic with its model, and the physical level – controlling connections of network devices. Ontological representation provides a model of the network which is used for creation specifications for intrusion-detection systems.

Teslya and Ryabchikov (2018) develop an upper-level ontology to describe the main components of industrial socio-cyber-physical systems and the connections between them. Eskins and Sanders (2011) introduce a definition of the cyber-human system and its elements. They define an opportunity-willingness-capability ontology for classifying cyber-human system elements concerning system tasks. The elements are classified into four types: components, participants, processes, and tasks.

The Mixed-criticality system is an integrated suite of hardware, operating system, middleware services, and application software that supports the execution of safety-critical, mission-critical, and non-critical software within a single, secure computing platform. The design of such a system is identified as a core foundational concept in the design of CPS. Herrera *et al.* (2013) first identify main design disciplines involved in the mixed-criticality system at both system-on-chip scale and SOS scales and then propose a core ontology for modeling a mixed-criticality system at both scales. The proposed ontology provides a core terminology of the design of a mixed-criticality system. However, although it provides some statements, it is still a work in progress and needs further development in formal ontology modeling language to allow validation.

Failure modes, effects, and criticality analysis (FMECA) model is an approach for the detection and prevention of sensor failure in CPS. Ali and Hong (2018) transform the FMECA model into a UML diagram and implement the UML class model in Protégé to build an ontology for failure detection and prevention.

3.4.4 SE Standards

This subsection reports the development of ontologies in support of system modeling and SE relevant standards. There are ten papers found in the literature, which deal with using ontologies to improve SE standards in relation to modeling. Also, 13 papers are identified for applying ontologies in aid of better understanding and practicing SE relevant standards. It can be seen that ontologies have contributed to SE standards in many ways.

D1. Modeling standard

Dori and Reinhartz-Berger (2003) develop OPM, a holistic formal yet intuitive conceptual modeling approach, for the development of complex socio-technical systems and knowledge management. In December 2015, the International Organization for Standardization recognized OPM as ISO/PAS 19450 (ISO/PAS International Standard, 2015). An overview of the evolution of OPM modeling tools, the older OPA CASE Tool, the current OPCAT, and the future OPCloud are discussed in Dori *et al.* (2018).

Lopez-Lorca *et al.* (2011) propose a process to support developers in modeling tasks using ontologies to validate and improve the quality of requirement analysis models as they are being developed and, at the same time, bridging the traditional gap between developers and clients. Ontologies can enable the reasoning of assigning properties to the concepts defined in the domain automatically. Madni *et al.* (2001, 1998) argue that enterprise modeling and process management share a common set of concepts. Therefore, they create an ontology, named IDEON/IPPD, support the design and tailoring of SE processes from an integrated product-process development (IPPD) perspective. This conceptually unified ontology for SE process design and management is key to supporting process redesign and streamlining.

Sarder and Ferreira (2007) focus on the functional domain in SE. An ontology is designed to assist interested parties in understanding the broad and multi-faceted nature of the discipline of SE.

Hoppe *et al.* (2017b) use ontologies to increase the quality levels of developing increasingly complex systems by creating semantically rich data models. While classical model-based applications provide well-established engineering functions, ontologies contribute several advantages: reasoning, classifying, and sharing. Reasoning derives further knowledge automatically from data based on rules that have been applied manually by system engineers in the past. Classifying creates additional types and more detailed types than those that have already been applied to derive not explicitly covered knowledge. Data shared between engineering domains can be analyzed in an overall context to detect inconsistencies and provide means to generate overall project metrics.

Chourabi *et al.* (2010) propose a flexible ontology-based schema with formally defined semantics to enable the capture and reuse of SE experience. It contains the fundamental concept for a holistic SE knowledge model. This general ontology is developed in a domain, product, and process facet. The three levels provide a comprehensive semantic model for the SE project asset through an integrated representation of its semantic content, its structural content, and its design rationale.

Triantis and Collopy (2014) carry out initial work for building a SE ontology by discussing some key concepts in SE, such as artifact, system, subsystem, and component. They also express their views on the relations between systems and organizations. They believe that the concept organization is at the core of SE, which should be understood in two ways: (1) characterizing the structure within an artifact, and (2) referring to a cultural institution within which people work together in a structured manner. Aslaksen *et al.* (2011) develop a high-level ontology for SE under the premise that SE is a sub-process within the overall process of engineering. Therefore, the concepts and relations inherit many of the features of engineering. The ontology is used for providing a shared vocabulary for communications about SE.

D2. Related standard

van Ruijven (2015, 2013) develop an ontology of the interpretation of ISO/IEC/IEEE 15288 based on the author's years of experience with the standard. Moreover, Yang *et al.* (2017) develop a formal ontology for ISO/IEC/IEEE 15288 according to the Input-Process-Output (IPO) diagrams defined in the INCOSE SE handbook.

Agrawal (2016) develops an ontology to structure and organize core concepts of the risk assessment phase of ISO/IEC 27005: 2011 standard. Blokland and Reniers (2018) propose an ontological and semantic foundation for safety science, based on an etymological and etiological study of the concepts of risk and safety. This foundation is aligned with the semantics and concepts used in the ISO 31000 risk management standard.

Eito-Brun (2016) develops ontologies to manage the different artifacts and information items requested in the European Space Agency (ESA) ECSS standards, including the ECSS-M-ST-40.

Software engineering standards developed under the auspices of ISO/IEC JTC1's SC7 have been identified as employing terms whose definitions vary significantly between standards. Henderson-Sellers *et al.* (2014) and Gonzalez-Perez *et al.* (2016) create an ontological infrastructure that aims to be a single coherent underpinning for all SC7 standards, including ISO/IEC/IEEE 24765 Systems and Software Engineering - Systems and Software Engineering Vocabulary. To develop this infrastructure, they identify five distinct areas where conceptual modeling and ontologies might help reorganize SC7 standards. Among them, an abstract domain ontology, named definitional elements ontology, forms the basis by providing non-specific details of any standard.

Guessi *et al.* (2015a) develop a formal ontology, named OntolAD, for ISO/IEC/IEEE 42010 Systems and Software Engineering - Architecture Description to support automatic conformance validation and enhance architectural descriptions reuse. Martin *et al.* (2017) define an ontology, named ArchiMEO, for capturing enterprise-specific knowledge. This ontology also consists of concepts defined in ISO/IEC/IEEE 42010 to depict architecture viewpoints. Roldán *et al.* (2018) develop an ontology-based approach for sharing, integrating, and retrieving knowledge from different architectural knowledge sources, which is based on ISO/IEC/IEEE 42010.

Ferchichi *et al.* (2008) propose an ontology to map two quality standards, ISO 9001: 2000 and capability maturity model integration (CMMI). Pardo-Calvache *et al.* (2014) develop an ontology of process-reference models, named PrMO, which defines a typical structure of process elements to support the harmonization of structural differences of multiple reference models, through the homogenization of their process structures. They validate the ontology through the instantiation of the information contained in different models and standards, including ISO 9001: 2008.

3.4.5 Generic Life Cycle Stages

This subsection looks at the SE knowledge areas from the life cycle stages and processes perspective. Therefore, in this section, the role of ontologies in different life cycle stages can be well identified and analyzed. Note that some studies that are reported in the previous sections also provide evidence of using ontologies in different life cycle stages. There are also papers that cover more than one life cycle stage. This can be identified in Table 3.3.

E1. Concept stage

This subsection contains the application of ontologies in the process of business analysis and stakeholder requirements definition.

Bergholtz and Eriksson (2015) propose an ontology for the institutional domain that is used for supporting conceptual modeling in business analysis. Pfaff and Krcma (2018) present a system architecture for an integrated data management of distributed databases based on a domain-specific ontology. This ontology is linked to data sources and functions as the central concept for database access, which is an increase in knowledge and data sharing which will enhance existing business analysis methods. Thus, additional databases can be integrated by linking them to this domain-specific ontology and are directly available for further business analyses. Silega *et al.* (2016) suggest using ontologies for the transformation and mapping from abstract domain models (technology-independent) to technology-dependent models, through platform-independent models, which is a crucial issue of business process modeling. They define the process of generating and validating model transformations by using ontologies, and the description of business processes within an organization contributes to the automation and quality of the architectural design.

Ryan and Wheatcraft (2017) develop a cohesive set of definitions of the terms associated with stakeholders and requirements, such as entity, need, requirement statement, requirement expression, characteristics of a well-formed requirement statement, and attributes of a requirement. They argue that a much more precise ontology is needed for agreements on standard definitions across the full requirements engineering domain.

E2. Development stage

Here, a summary of the use of ontologies in systems development is provided, including processes such as architecture definition and design definition.

Hallberg *et al.* (2014) present the definitions, dependencies, and relationships of the most fundamental concepts in systems development in the form of an ontology. The ontology consists of four categories of concepts: general concepts, description concepts, realization concepts, and appearance concepts. The two core concepts in the ontology are systems and systems development.

Miller (2017) presents an ontology for developing airspace system architecture, which composed of a collection of entities, properties, and relationships representing the key system concepts. It separates the domain knowledge from the operational, thus enabling the development of architectural variations derived from a common language and understanding of the Airspace Systems. From the ontology, development and investigations can be made for many architectural variations founded in a common vocabulary and understanding of the airspace system domain.

Hatchuel *et al.* (2013) derive an ontology of design from a comparison between formal design theories developed in two different scientific fields: engineering and set theory. It clarifies six of the main features of design as rigorously as possible. Sim and Duffy (2003) identify and classify a generic set of design activities from published literature into what is referred to as design definition activities, design evaluation activities, and design management activities. They are considered as an ontology of the design activities that designers perform in the design process. A set of consistent and coherent definitions of these activities are deliberated and presented. Sarder *et al.* (2007) propose a methodology, named domain knowledge acquisition process (DKAP), for design ontologies modeling.

The synthesis of design solutions is a stage of the SE design process. Christophe *et al.* (2009) present a framework of conceptual design specific for SE by developing a mid-level ontology to integrate with other ontologies of engineering and the connections with lower taxonomies. The framework also integrates ontology search with SysML semantics supported by a computer application, OPAS, a guide for designers during the synthesis of conceptual solutions.

Flexibility is frequently hailed as a desirable system design characteristic. However, in the SE literature, flexibility remains an ambiguous concept. Ryan *et al.* (2013) employ an ontological framework for clarifying salient aspects of extant flexibility-related terminology. The proposed definitions of these fundamental system design principles can provide a baseline for improving analysis and communication among SE practitioners and academics.

Witherell *et al.* (2007) develop an ontology for engineering design optimization, which incorporates both standardized optimization terminology, formal method definitions, and often unrecorded optimization details. They utilize the ontology in a prototype computational knowledge-based tool, named ONTOP, and implement ONTOP in two engineering design optimization case studies. The results illustrate the potential value of an ontology in representing application-specific knowledge while facilitating both the sharing and exchanging of this knowledge in engineering design optimization.

E3. Production stage

In the systems production stage, it is found that ontologies are developed to support systems integration. Madni and Sievers (2014b) develop a systems integration ontology that captures the key issues and concerns in a standard language to enable efficient information transport within a system and the interactions among stakeholders. This ontology minimizes errors that cause systems integration failures. Specifically, it provides the basis for building a checklist that can be reasoned with in ways that allow many integration problems to be avoided or detected and circumvented. Furthermore, they present an SOS integration ontology (Madni and Sievers, 2014a). This ontology includes both artifacts (documentation) and metrics (measurements or tests used to assess integration success). The concepts are related to the SOS engineering domain through relationships (associations).

3.4.6 SE Management

Implementing SE requires the coordination of technical and managerial endeavors. This subsection is about managing the resources and assets allocated to perform SE, often in the context of a project or a service. However, each of these management disciplines has its own body of knowledge that can be discussed in detail; thus, this review tried to focus on the technical or engineering aspects to tighten up the scope. Moreover, these management disciplines are intertwined with project management and software engineering. This section does not provide an all-inclusive review but focuses on the research that is SE oriented.

F1. Planning

Bouras *et al.* (2016) develop an ontology to maintain the historical data during the project planning process to help project managers with an accurate and realistic estimation of timelines of the project. It consists of three major concepts of project management: project, employee, and task. Lee *et al.* (2008) develop a project planning ontology based on CMMI. The project planning ontology involves the following: developing the project plan, interacting with stakeholders appropriately, getting a commitment to the plan, and maintaining the plan. Líska and Návrat (2010) present an approach that enhances the software and systems process engineering meta-models by an ontology to support requirements specification activity in the project.

F2. Assessment and control

Hahn *et al.* (2008) argue that ontologies can be used in product design project performance assessment to describe the project output independent from data models used by the design tools to store their results and from the different engineering disciplines.

F3. Decision management

Wulandari *et al.* (2018) discuss the role of ontologies in developing decision support systems. They provide an SLR on ontology-based decision support systems. Interested readers can refer to the review to learn more about the state of the art of ontologies in decision management. Cruz *et al.* (2018) develop an ontology to describe design decisions. Based on this ontology, they propose an approach to represent design decisions. It is validated through checking its properties as a formal logical system and the applicability of the ontology to support the design decision process of a specific project. Gorshkov *et al.* (2016) propose a method of ontological representation of multiple viewpoints using named graphs to allow formulating the functional requirements for a multi-viewpoint decision-making support system.

F4. Risk management

Tserng *et al.* (2009) propose an ontology-based risk management framework to enhance risk management performance by improving the workflow and knowledge reuse. The study verifies that project risk ontologies can be developed by acquiring tacit knowledge and extracting explicit knowledge from the organization. Nota *et al.* (2010) develop a risk ontology whose aim is to capture the fundamental concepts of risk management together with a formal specification of rules to qualify operational aspects of risk management. Also, Sales *et al.* (2018) carry out an in-depth ontological analysis on the nature of risk and unclear notions in performing risk analysis under the principles of UFO. Lykourentzou *et al.* (2011) develop an ontology for operational risk management to facilitate information sharing across organizational unit boundaries and enable computational inferences over the heterogeneous applications of the organization. It provides a unified view of operational risk management information.

Ansaldi *et al.* (2012) develop two ontologies, named OntologyGuide73 and OntologyRATIS, for risk management domain to facilitate the reading and understanding of the guidelines for risk assessment and support the choice of the most suitable method for a given context among the available technologies.

Birkholz *et al.* (2012) develop an interconnected-asset ontology as a step towards a standardized representation of detailed asset information. It is a machine-readable representation that supports the automation of risk management processes, and the standardization of asset information reduces redundant acquisition processes that are often found in practice. Jiang and Zhang (2013) construct an ontology that can be used as the foundation of construction projects' risk events database. This ontology includes risk sources, risk events, and risk consequences. It provides a unified risk-related concept set through illustrating concepts and the relationships between them. This ontology can be utilized, shared, and reused by people and computers and lays a solid foundation for semantic retrieval.

Guo and Nunes (2009) develop a risk identification checklist for facilitating user companies to identify, organize, and manage potential risks associated with the post-adoption of enterprise resource planning (ERP) systems. A risk ontology is subsequently established to highlight these ERP risks, as well as to present their potential causal relationships. The risk ontology is a valuable tool and checklist to support risk identification, prevention, management, and control, as well as to facilitate strategic planning and decision making.

F5. Configuration management

Zhang (2014) presents a state-of-the-art review of product configuration ontologies. Eito-Brun (2018) reports a case study in the aerospace sector that uses ontologies to streamline the management of configuration management related data, including both the inventory of software and hardware components and their aggregation into higher-level configuration items. It is achieved by defining a configuration management ontology to allow the capture of all the configuration management data based on and requested in the European Space Agency standard. The ontology is designed using the RDF/OWL modeling language and modeled with the TopBraid software tool. Unfortunately, the full ontology is not published, but the taxonomy of configuration items can be inferred in the paper. The properties are defined to keep information about different aspects of both the configuration items and the recurrent units.

F6. Information management

Blanco *et al.* (2008) conduct an SLR on information security ontologies. They compare the studies by represented contents, language, methodology, software environment, and cost of using the ontology in new systems. Wimalasuriya and Dou (2010) provide an introduction to ontology-based information extraction (OBIE) and review the details of different OBIE systems. They discuss the implementation details of these systems, including the tools used by them and the metrics used to measure their performance. Grubic and Fan (2010) review the supply chain ontologies in terms of their contributions to attaining information systems interoperability. Mikroyannidis and Theodoulidis (2010) examine existing approaches in information management, as well as ontology management and evolution in business intelligence.

F7. Measurement

Honour and Valerdi (2014) build an ontology for SE to allow consistent measurement. It contains broad-based definitions of key terms such as systems engineering effort, amount of effort, type of effort, quality, success, and optimum.

Bertoa *et al.* (2006) present a software measurement ontology proposal. The concepts of the ontology and relationships among them are presented in detail and grouped according to the sub-ontology to which they belong. A running example based on a real case study is used to illustrate the ontology.

F8. Quality management

To build a knowledge repository for managing requirements quality, we need first clearly define the typology of requirements that are going to be covered by the knowledge system, as this will affect the requirements structure and vocabulary. Fraga and Llorens (2015) propose an ontology-based knowledge management process to enhance standardizing and normalizing the terminology used in requirements quality management.

Geisler *et al.* (2016) propose an ontology-based data quality framework for relational data stream management systems that includes data quality measurement and monitoring in a transparent, modular, and flexible way. The core of the data quality framework is an ontology that manages all data quality-related meta-data, such as the data sources, their data quality factors, and data quality metrics.

Kim and Fox (2002) present a TOVE measurement ontology used as data models to provide quality management services. An assessment system for measuring attributes of an entity, activities for measurement, and quality as conformance to requirements are the core concepts represented in the ontology.

3.4.7 Summary Remark for RQ1

In Section 3.4, a detailed report on what SE knowledge areas are supported by ontologies is presented. The results show that there are special focuses on applying ontologies in particular SE knowledge areas, such as the development stage of the SE life cycle, the cyber-physical systems, the SE related standards. However, there are also many knowledge areas that have only received limited attention. These areas need more exploration regarding the benefits of ontologies. Therefore, in the next section, the overall contributions of ontologies brought to the SE domain are synthesized to create an opportunity for digging up the potentials.

3.5 The Role of Ontologies in SE (RQ2)

RQ2 focuses on the contributions that ontologies make to SE. In other words, why ontologies are created and used in SE. This section answers RQ2 - for what purposes ontologies have been used in SE, i.e., benefits, value, and contributions. These effects are synthesized from the original statements in the reviewed papers and are summarized in Table 3.4.

Table 3.4 Contributions of ontologies for SE

#	Contributions of ontologies for SE
1	Enabling interoperability and communication among multiple disciplines or across different stakeholders
2	Integrating, mapping, exchanging and reusing knowledge
3	Describing concepts and their relationships explicitly and accurately to avoid incompleteness and ambiguity
4	Developing a domain knowledge representation
5	Unifying a controlled vocabulary or semantics for capturing declarative knowledge
6	Providing core and basic concepts as a reference to describe other concepts
7	Defining a homogeneous terminology to eliminate inconsistency
8	Sharing a common understanding of a domain
9	Capturing knowledge in a formal language
10	Allowing, expressing and reasoning about machine-readable programmable complex logical axioms
11	Visualizing and navigating knowledge repository

These benefits reflect the nature of the ontology itself as well as the various interpretations of the functions of ontologies by the research community. The contributions of ontologies vary depending on the nature and type of SE problems. Nevertheless, it is worth noting they are often linked. For example, when ontologies are used to enable interoperability (#1 of Table 3.4), they will generally contribute to forming a controlled vocabulary or semantics (#5), which can result in a homogeneous terminology to eliminate inconsistency (#7). These functions are not combined into one category, as they have a slightly different emphasis in terms of the ultimate impact.

To sum up, it is confirmed that ontologies contribute to SE problems in various ways. Most of the time, the contributions are often implicit or concealed in the articles, which makes the evaluation of the real functions of the ontologies hard to get. Therefore, in order to reveal the real contributions of ontologies, it is necessary to evaluate the existing ontologies that are artifacts to conceptualize the SE knowledge. In the next section, extant SE ontologies will be reviewed and reported.

3.6 Extant SE Ontologies and Adopted Techniques (RQ3 and RQ4)

RQ3 deals with the scope of the ontologies that are developed for SE as artifacts. This section summarizes the existing ontologies for SE and provides an answer to RQ3 regarding ontologies as final deliverables. Table 3.5 presents all the ontologies and their scopes. As many papers do not publish all the details about the ontologies, Table 3.5 resolves some key concepts and properties according to the content of the papers.

Table 3.5 Ontologies and their scope

References	Topics	Scopes	C[1]	P[2]	R[3]	Key concepts	Key properties
Dori (2016, 2002)		SE	√	√	√	system, subsystem, stakeholder, beneficiary, customer, user, supplier, product, service, function, structure, and behavior	procedural link, structural link
Honour and Valerdi (2014)		SE	√			systems engineering effort, amount, type, quality, success, optimum	
Madni *et al.* (2001, 1998)		SE	√	√	√	entity, enterprise, process, constraint	deploy, achieve, employ, set, own
Triantis and Collopy (2014)		SE	√			artifact, system, subsystem, component	
Aslaksen *et al.* (2011)	SE body of knowledge	SE	√	√	√	process, project, activity, design, implementation, operation, maintenance	perform, defined by, provide, incur, produce
Chourabi *et al.* (2010)		SE	√	√	√	entity, resource, process, product, domain, requirement	specialize, instantiate
Sarder *et al.* (2007); Sarder and Ferreira (2007)		SE	√			systems engineering function, systems engineering object, technical management, technical execution, actor, product	
van Ruijven (2015, 2013)		ISO/IEC/IEEE 15288	√	√	√	purpose, objective, stakeholder requirement, process, service	is derived from, consist of, is realized by
Yang *et al.* (2017)		ISO/IEC/IEEE 15288	√	√	√	process, object, input, output, control, enabler	
Hennig *et al.* (2016)	MBSE	MBSE	√	√	√	connector, contamination element, discrete model, discrete state, functional port	configure, consist of, contain element
Easterbrook (2014)		system behavior	√			stock, flow, emergent behavior, feedback loop	
Mason (2005)	behavior	organizational behaviors		√			decide, inform, monitor, evaluate, hold, exchange, transform, locate
Kaderka *et al.* (2018)		behavior and scenario	√	√	√	scenario, temporal constraint, system, schedulable behavior constraint, behaving element	has type, constrain, from, to, begin, end

References	Topics	Scopes	C[1]	P[2]	R[3]	Key concepts	Key properties
Borgo *et al*. (2009)		artifact behavior	√	√	√	artifact behavior, perdurant, endurant, behavior environment, behavioral constraint	participate, span, exist
Gero and Kannengiesser (2007)		object behavior	√			function, behavior, structure, input, transformation, output	
Herzig *et al*. (2011)	inconsistency	inconsistency	√	√	√	scientific data, idea, belief, preference, model, modeling language, mathematics	has, consistent with, observation of, informed by
Dogan *et al*. (2012)	capability engineering	capability engineering	√	√	√	activity, capability engineering, organization, system, service, outcome, perspective, resource	encompass, consider, generate, comprise
Eskins and Sanders (2011)	opportunity-willingness-capability	opportunity-willingness-capability	√			component, participant, process, task	
Ali and Hong (2018)	failure	failure	√	√	√	component, failure, severity criticality, recommended action	has failure, has criticality, has hazard
Balduccini *et al*. (2018)	trustworthiness	trustworthiness	√	√	√	concern, aspect, trustworthiness, security, cyber security	has sub concern satisfy
He *et al*. (2014)		SOS	√		√	history data, aim, capability index, functional characteristics, system, system function, system effectiveness, system cost, system quantity, system relation	
Ferreira and Tejeda (2011)	SOS	unmanned and autonomous SOS test and evaluation	√	√	√	test, test and evaluation plan, resource, test script, test object, test type	develop
Zhu *et al*. (2017)		SOS mission	√	√	√	system, system of systems, constitute system, system configuration, emergency behavior, task, mission	decompose, support, conflict
Madni and Sievers (2014a)		SOS integration	√			verification and validation, integration, certification and accreditation, tailoring and reuse, stakeholder, configuration management	
Guessi *et al*. (2015a)	architecture	ISO/IEC/IEEE 42010	√	√	√	architecture description element, environment, architecture, system	has concern, is interested in, frame

References	Topics	Scopes	C[1]	P[2]	R[3]	Key concepts	Key properties
Miller (2017)		airspace system architecture	√	√	√	system resource, aircraft, resource manager, resource authority, airspace system	use, support, manage, own, has
Sim and Duffy (2003)		design activities	√			abstracting, associating, composing, decomposing, defining, detailing	
Witherell *et al*. (2007)		design optimization		√			assumption, author, constraint, description
Lynch *et al*. (2017)	design	design and component	√	√	√	system, launch system, aerodynamic system, power system, mass measurement unit	has direct component, has mass unit
Cruz *et al*. (2018)		design decisions	√	√	√	architecture view, technology, pattern, communication interface	has style architectural service oriented, has communication interface web service, has structure component
Hallberg *et al*. (2014)	development	system development	√	√	√	general concept, description concept, realization concept, appearance concept, context, system, component, architecture, model, design	exist in, consist of, describe, support development of
Orellana and Madni (2014)	integration	human-system integration	√	√	√	requirement, human agent, behavior, structure, parametric, mechanism	basis for, document, represent, comprise, describe, conform to, extend
Madni and Sievers (2014b)		systems integration	√			integration, certification and accreditation, tailoring and reuse, stakeholder, configuration management	
Lee *et al*. (2008)	project planning	project planning	√	√	√	scope, work product and task attribute, life cycle, effort and cost, budget and schedule, risk	estimate, establish, define, determine, identify
Nota *et al*. (2010)		risk	√	√	√	environmental context, cluster, station, risk responsible, sensor, risk	is subdivided in, own, is managed by, is equipped with
Lykourentzou *et al*. (2011)	risk	operational risk management	√	√	√	process, process risk, risk event, treatment plan, risk root cause	has risk, has result, has impact, has treatment plan
Ansaldi *et al*. (2012)		risk management	√	√	√	risk assessment method, risk assessment, expected input, application sector, expected output	include, need input, provide output

References	Topics	Scopes	C[1]	P[2]	R[3]	Key concepts	Key properties
Jiang and Zhang (2013)		construction projects' risk	√			risk source, risk event, risk consequence, unexpected event, adverse change	
Guo and Nunes (2009)		enterprise resource planning risk	√			operational risk, analytical risk, organization-wide risk, technical risk	
Agrawal (2016)		ISO/IEC 27005: 2011	√	√	√	risk, consequence, control, vulnerability, organization	contain, modify, lead to, own, has
Bertoa *et al*. (2006)	measurement	measurement	√	√	√	measure, scale, unit of measurement, measurement, measurement result	has, expressed in, belong to, use
Ferchichi *et al*. (2008)	quality	ISO 9001: 2000	√	√	√	quality standard, recommendation set, practice, mapping, maturity level	include, group, mapping
Annamalai *et al*. (2011)	product service systems	product service systems	√			product service systems, stakeholder, product life cycle, business model, customer need	
Rese *et al*. (2013)	product service systems	business model	√			value, organization, risk distribution, revenue stream, property right	
Nardi *et al*. (2015)		service as commitment	√	√	√	agent, service provider, service offer, target customer, service offering claim	create, describe, member of, involve, part of, inhere in
Dong *et al*. (2011)	service	service	√	√	√	service element, port, property, constraint, function, supplier, resource	has port, has property, has constraint, has function, supplied by, has resource
Lemey and Poels (2011)		service science	√	√	√	entity, service, resource, stakeholder	enable, consist of, use, participate in
Trappey *et al*. (2018)		CPS technology and function	√			connection, conversion, computation, cognition, configuration	
Jeong *et al*. (2018)	CPS	CPS	√	√	√	physical object, object profile, object model, service, computational service, cyber service, input, output, request, acting	present, support, has input, has output, has request, has acting
Daun *et al*. (2016)		CPS context information	√	√	√	environment, irrelevant environment, context, context object, context subject, system, software, hardware	are separated, influence, constraint, interact with

References	Topics	Scopes	C[1]	P[2]	R[3]	Key concepts	Key properties
Brings *et al*. (2018)		CPS context information	√	√	√	environment, irrelevant environment, context, context object, context subject, system, software, hardware	are separated, influence, constraint, interact with
Teslya and Ryabchikov (2018)		socio-cyber-physical systems	√	√	√	supply chain operation, manufacturing machine and capability, product and material, structural relation	use, concretize, on, consume

Note:

1. C = concept, '√' if concepts are defined in the ontology
2. P = property, '√' if properties are defined in the ontology
3. R = relationship, '√' if relationships are defined in the ontology

From Table 3.5, it can be seen that many ontologies are developed for SE to fulfill different purposes. Most of them have a specific scope and provide concepts and relations within that scope to a certain extent. However, the formality of the ontologies is subject to further investigation. Therefore, in the next section, a review is conducted, especially from an ontology engineering perspective, since the techniques can reflect the degree of formality of the extant ontologies.

RQ4 deals with languages, methods, and tools that have been adopted to develop SE ontologies from an ontology engineering perspective. This section reports the results of the analysis of the ontologies. It follows a framework proposed by Scheuermann and Leukel (2014), including ontology engineering techniques such as languages, methods, and tools. The results are presented in Table 3.6.

Table 3.6 Analysis of ontologies from an ontology engineering perspective

References	Languages	Methods	Tools
Dori (2016, 2002)	OPM[1]	OPM	OPCAT[2] OPCloud[3]
Honour and Valerdi (2014)	NLD[4]	NS[5]	NS
Madni *et al.* (2001, 1998)	UML[6]	NS	NS
Triantis and Collopy (2014)	NLD, DD[7]	NS	NS
Aslaksen *et al.* (2011)	NLD, DD	NS	NS
Chourabi *et al.* (2010)	NLD, DD	NS	NS
Sarder *et al.* (2007); Sarder and Ferreira (2007)	IDEF5[8]	DKAP[9]	IDEF5
van Ruijven (2015, 2013)	RDF[10]	NS	Relatics[11]
Yang *et al.* (2017)	OWL[12]	AOM[13]	Protégé[14]
Hennig *et al.* (2016)	OWL 2[15]	NS	Protégé
Easterbrook (2014)	NLD	NS	NS
Mason (2005)	NLD	NS	NS
Kaderka *et al.* (2018)	NLD, DD	NS	NS
Herzig *et al.* (2011)	NLD, MD[16], DD	NS	NS
Dogan *et al.* (2012)	OWL	AOM	Protégé
Eskins and Sanders (2011)	NLD, MD	NS	NS
Ali and Hong (2018)	OWL	NS	Protégé
Balduccini *et al.* (2018)	NLD, MD	NS	NS
He *et al.* (2014)	NLD, MD, DD	NS	NS
Ferreira and Tejeda (2011)	UML, OWL	DKAP	Magic Draw[17], Protégé
Zhu *et al.* (2017)	OWL DL[18]	NS	NS
Madni and Sievers (2014a)	SysML[19]	NS	NS
Guessi *et al.* (2015a)	OWL	NS	Protégé
Miller (2017)	NLD, DD	NS	NS
Sim and Duffy (2003)	NLD	NS	NS
Witherell *et al.* (2007)	OWL	NS	Protégé
Lynch *et al.* (2017)	OWL	NS	Protégé
Cruz *et al.* (2018)	OWL	NS	Protégé
Hallberg *et al.* (2014)	NLD, DD	AOM	NS
Orellana and Madni (2014)	UML	NS	NS

References	Languages	Methods	Tools
Madni and Sievers (2014b)	SysML	NS	NS
Lee *et al.* (2008)	NLD, DD	NS	NS
Nota *et al.* (2010)	NLD, DD	NS	NS
Lykourentzou *et al.* (2011)	NLD, DD	NS	NS
Ansaldi *et al.* (2012)	NLD, DD	NS	NS
Jiang and Zhang (2013)	OWL	AOM	Protégé
Guo and Nunes (2009)	NLD, DD	NS	NS
Agrawal (2016)	NLD, DD	NS	NS
Bertoa *et al.* (2006)	UML	NS	NS
Ferchichi *et al.* (2008)	UML	NS	NS
Dong *et al.* (2011)	OWL	Uschold and King[20]	Protégé
Lemey and Poels (2011)	UML	NS	NS
Trappey *et al.* (2018)	NLD, DD	NS	NS
Jeong *et al.* (2018)	OWL	NS	NS
Daun *et al.* (2016)	UML	NS	NS
Brings *et al.* (2018)	UML	NS	NS
Teslya and Ryabchikov (2018)	NLD, DD	NS	NS

Note:
1. OPM = object-process methodology
2. http://esml.iem.technion.ac.il/opcat-installation/
3. https://www.opcloud.tech/
4. NLD = natural language description
5. NS = not specified
6. UML = unified modeling language
7. DD = diagrams demonstration
8. http://www.idef.com/idef5-ontology-description-capture-method/
9. DKAP = domain knowledge acquisition process
10. RDF = resource description framework
11. https://www.relatics.com/
12. OWL = web ontology language
13. AOM = authors' own method
14. https://protege.stanford.edu/
15. OWL 2 = version 2 of web ontology language primer
16. MD = mathematical definition
17. https://www.nomagic.com/products/magicdraw
18. OWL DL = web ontology language description logics
19. SysML = systems modeling language
20. Uschold, M., and King, M. (1995). Towards a Methodology for Building Ontologies

The analysis shows that the degree of formality of most of the ontologies is low. The reason is that very few studies report the methodology of how the ontology is built. Moreover, most of the ontologies are still at a conceptual stage, described by natural languages and informal diagrams, and are not created by professional tools, which hinder the ability to reuse and share them. The ontologies shown in Table 3.5 are published in various journals and conference proceedings. Surprisingly, only two ontologies are available online and to download. Thus, the problem with most ontologies is that from reading the paper, it is difficult to grasp the formal specification sufficiently. Very often, the level of description is rather high, and the detail is deficient.

3.7 Conclusion

This chapter presents a comprehensive literature review on the state of the art of OBSE. It focuses on what SE knowledge areas are supported by ontologies, what contributions that these ontologies have made to SE, what concepts and relations are covered by these ontologies, and what kinds of methods, developing tools, and languages are used to build these ontologies. Based on the results of the literature review, key conclusions can be drawn as follows.

- There is no ontology that tends to capture the whole SE body of knowledge.
- The methods of developing SE ontologies are all manual and in need of automation.
- The extant ontologies remain at a general level and lack a detailed representation.
- The development of the ontologies requires using sophisticated languages and tools to increase the formality.
- Little work has been done regarding the visualization of the ontologies and their potential application scenarios.

Next, a research methodology is presented in Chapter 4. It is designed for addressing the research gaps identified from the literature and guiding the research towards targeted deliverables.

CHAPTER 4 RESEARCH METHODOLOGY

4.1 Introduction

This chapter presents the research methodology for developing a formal ontology for the SE body of knowledge. It begins with a brief review of the research framework (Section 4.2). Then, the proposed ontology learning approach is presented in detail. The proposed approach contains three key stages, document collection and pre-processing, lexical analysis through NLP, and ontology components extraction. They are respectively elaborated by the three subsections in Section 4.3.

4.2 Research Framework

This section summarizes the research framework from the highest level. Figure 4.1 is the top-level framework also modeled by IDEF0.

IDEF methods are a suite or family of methods that support a paradigm capable of addressing the modeling needs of systems and software engineering fields. IDEF0 is the function modeling method that is designed to model the decisions, actions, and activities of an organization or system. The activities are represented by squares. Arrows contain four roles, which are inputs (from left boarder to activity), outputs (from activity to the right border), controls (from up border to activity), and mechanisms (from down border to activity). Activities can be zoomed into sub-diagrams, which include sub-activities and participants.

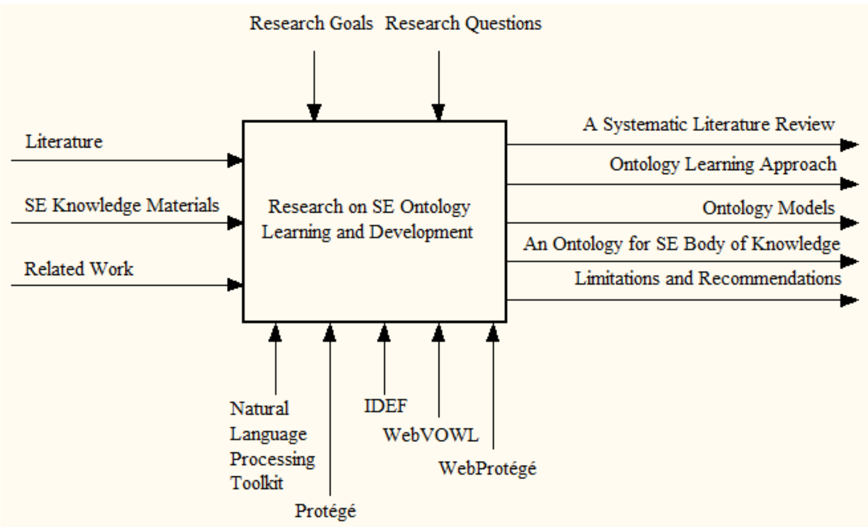


Figure 4.1 Top-level research framework

In Figure 4.1, research questions and goals control the research process in general. Key deliverables are yield from different stages of the research. The mechanisms show supportive tools and platforms, which are indispensable technological means for this research. As external inputs, literature, SE knowledge materials, and other related work are imported into the ontology learning and development process. The outputs from this research are

- an SLR on OBSE,
- a novel ontology learning approach for learning SE ontological knowledge from textual resources,
- a set of SE ontology models that frame the SE ontology,
- a formal ontology that represents the SE body of knowledge, and
- a summary of the research contributions, limitations, and future work.

The design of the study is based on and controlled by the research questions and research goals. As this research also concerns the development of an engineering artifact, an SE ontology, as one of the final deliverables, the development ought to be supported by sophisticated tools and platforms.

Figure 4.1 is the top-level diagram that represents the overall research framework. Figure 1.1 is a break-down of the research methodology that shows more details of the research framework. Figure 1.1 illustrates more details about the research activities that consist of the research framework. The eight activities are respectively to

- conduct a literature review on ontologies in SE,
- design a research methodology,
- propose an ontology learning approach,
- develop ontology models,
- apply the learning approach,
- populate data in the model,
- discuss the results and,
- identify limitations and future work.

First, an SLR was conducted to identify the gaps and understand state of the art. The results have been presented in Chapter 3. The gaps identified then guided the design of the research methodology to achieve the stated goals and solved research problems. Then, the study developed an SE ontology based on the research framework. According to the key deliverables, activities were implemented one after another, including proposing the ontology learning approach, developing ontology models, and applying the approach and the models. The learning approach yielded the expected ontological primitives. These primitives acted as the primary elements that consist of the ultimate ontology. The ontology was then edited, presented, and visualized using sophisticated tools. Finally, the ontology was compared with related work to discuss the strength and limitations.

Among the research activities, one of the critical steps is to propose an ontology learning approach. This approach is also realized by a case study to demonstrate the ontology learning process. In the next section, a more detailed discussion is provided on the existing ontology learning and development process.

Within the entire research, key deliverables are

- an SLR,
- a novel ontology learning approach,
- a set of SE ontology models
- a case study for demonstrating the proposed learning approach and validating the developed models, and
- a formal SE ontology presented by hierarchical visualization tools.

The SLR has already been presented in Chapter 3. The next section presents the proposed ontology learning methodology.

4.3 The Proposed Ontology Learning Methodology

As the major outcomes of the research shown in Figure 4.1, this section presents one of the significant outcomes, that is, the proposed ontology learning methodology.

An analysis of the literature shows that ontology development methodologies in SE are all manual and not making the most of existing SE standards. In order to improve the current research outcomes, this study used emerging ontology learning techniques. To generate a big picture of the proposed ontology learning approach utilized in this study, Figure 4.2 presents a top-level overview of the stages in the ontology learning approach. It was created based upon best practices from previous ontology learning efforts and depicted by Object-Process Methodology (Rosa *et al.*, 2019).

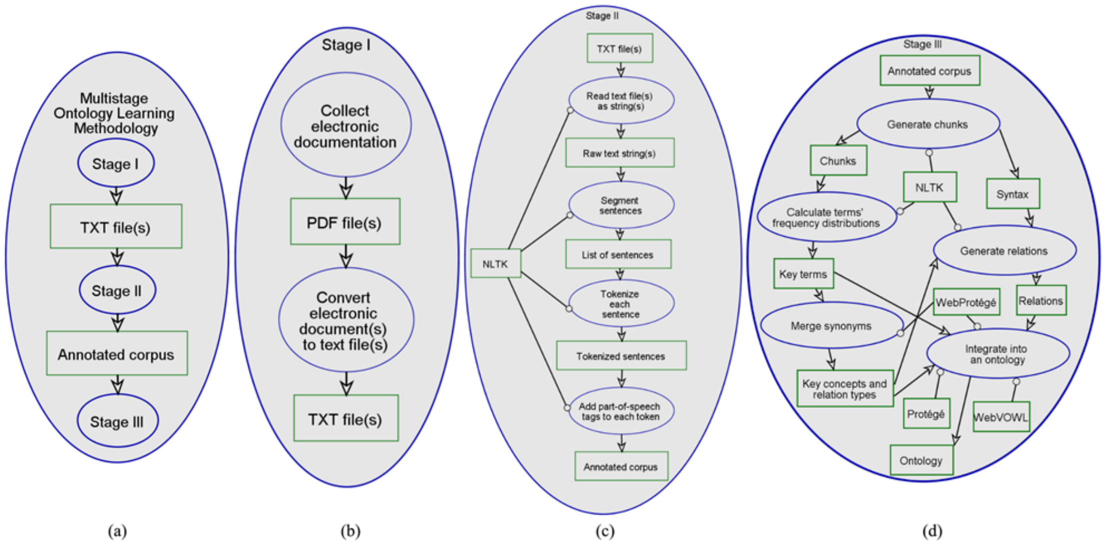


Figure 4.2 Multistage ontology learning approach

The methodology aims at learning ontological primitives from SE knowledge materials such as SE standards to develop SE ontologies in a semi-automatic fashion. Figure 4.2 (a) illustrates the top-level stages. Figure 4.2 (b) to (d) zoom into each stage and also shows the toolchain. The following sections detail each stage by describing the activities and steps with their expected outputs.

4.3.1 Stage I: Document Collection and Pre-processing

To begin with, SE standards in electronic documentation format were collected to create an SE domain of discourse. The files were authoritative reference materials, such as international standards, since they contain formal English terminologies and have high-quality content. These documents consist of an SE corpus. The length of the standards determines the size of the corpus and how much data is available in the later training and testing stage. Therefore, it is ideal to have sufficient data.

SE standards are commonly found in PDF format. It is good practice to keep as much information found in the standard when transforming the original PDF file into the TXT text document and cancel 'noises' such as figures, tables, page numbers, formatted texts as they may interfere with the logical flow of the text.

After the pre-processing steps above, a plain and tidy TXT file was brought into the next stage.

4.3.2 Stage II: Lexical Analysis through NLP

Advanced analysis tools were used to aid the execution of the second stage. It is especially important to conduct a scientifically rigorous Natural Language Processing (NLP) on the pre-processed documents. For this study, the Natural Language Toolkit (NLTK) platform was used for symbolic and statistical analysis, such as tokenization, stemming, tagging, parsing, and semantic reasoning. The detailed application of this toolkit is presented in Chapter 6: Case Study.

Stage II started with the text file obtained from stage I. The file was correctly encoded by character encodings such as Unicode, UTF-8, and ANSEL. It can ensure the text in a not messy or unreadable format. Initially, the text was loaded and read as a large string. The next step was to segment the large string into mutually independent sentences according to the sentence boundary. The function offered by NLTK or other NLP tools is not one-size-fits-all for all circumstances. Therefore, evaluation of the segmented sentences was conducted to improve the accuracy of the sentence boundary. In SE standards, paragraphs are often organized by chapters, sections, and headings; thus, it is necessary to check whether the titles are split from the following sentences. It is common to find the use of bullet points in SE standards to set forth parallel statements, therefore ensuring each of the bullet points is also separated is critical. A full stop was attached to each line break to obtain better results and improve the performance of the NLTK sentence tokenizer to ensure a thorough sentence segmentation further.

Following the sentence segmentation, a word tokenization was conducted on each sentence. With the help of NLTK word tokenizer, an SE corpus was obtained with tokenized words, which is better for lexical analysis. However, it is essential to evaluate the correctness of word tokenization since SE standards use abbreviations and unique punctuations, which might not be correctly recognized. A batch processing of special occasions was carried out as correct tokenization influenced the statistics in the later stage.

The last activity of Stage II was to add part-of-speech tags to each token. There are various ways to add word classes to the SE corpus automatically. It is, therefore, important to compare the performance of different taggers. A training corpus was designed to improve the adaptive ability of the tagger to acquire a tagger with a high accuracy score. The data were split into two, training the tagger on some data and test it on the remaining.

At the end of Stage II, an annotated SE corpus was developed, which is a major undertaking for the next stage.

4.3.3 Stage III: Ontology Components Extraction

In most ontology development, the layers conceptually build upon each another in the sense that higher layers rely on the output situated at lower layers. Consequently, the activities in Stage III were defined following this rule.

First, the entities were segmented and labeled, which are typically nouns, proper names, and definite noun phrases. Meanwhile, it is also useful to consider indefinite nouns or noun chunks since some of them might be frequently used terms in SE standards. In order to extract key terms in the SE domain, the frequency distributions of different kinds of terms were used.

As different terms may refer to the same concept, the merging of synonyms is an essential activity in this stage, making ontologies one of the best solutions to language ambiguity. The key terms extracted from the SE corpus were not only nouns or noun phrases, but also verbs and verb phrases. Generally, nouns and noun phrases are concepts in a relation, whereas verbs and verb phrases describe what relations exist between the concepts (they are called "relation types" in Figure 4.2). Therefore, when merging synonyms, not only different noun terms were merged, but also the verbs or verb phrases of the same relation. A data model was predefined to store these synonyms. Two other models (conceptual and logical models) were also created to aid the development of the SE ontology. The next chapter provides more details about these models and how they are developed.

The next task was to deal with abbreviations. In this study, they were automatically considered as synonyms to the full specified name of a concept. Certain language patterns were also used to detect terms that have the same semantics. Label annotations were used to present different names representing the same concept. Then, a set of key concepts with their possible synonyms were obtained, as well as a set of fundamental relations types.

The last activity was to assembly the concepts with relevant relations. There are two kinds of relations in ontologies, taxonomic and non-taxonomic. The taxonomic relation is the key to construct the concept hierarchy. Non-taxonomic relations describe how the concepts are linked together. In this study, the SE ontology was built with the help of an ontology editor called Protégé and its web-based version WebProtégé. They ensure that the logical axioms are established in formal ontology language (e.g., OWL). The tools also enable argument and inference. They can significantly enhance queries and automated reasoning (Dibowski *et al.*, 2018).

In summary, this approach converted natural language descriptions in the SE standards into rigorous computable definitions.

4.4 Conclusion

This chapter contains two parts. The first part presents the overall research framework that this study employed. The second part focuses on the ontology learning approach that this study proposed to learn an SE ontology from authoritative SE standards.

Regarding the research methodology, the key deliverables are presented in the research framework. So far, the first three activities, along with their outcomes, have been presented in Chapters 3 and 4. They are conducting a systematic literature review, designing a scientific research methodology, and proposing a novel ontology learning approach. Especially, as one of the key deliverables in the methodological aspect, the proposed ontology learning approach is described in detail in three stages. The next chapter presents the ontology models that are developed for formalizing the conceptual, logical, and data facets of the SE ontology.

CHAPTER 5 ONTOLOGY MODELING

5.1 Introduction

This chapter describes the ontology models that were developed for this study. It is the third output in the IDEF0 model in Figure 4.1. The ontology models can be seen as the overarching structure in which the SE knowledge can be gathered in a systematic and formal way. To be specific, three models are created for modeling the SE ontology. They are conceptual model, logical model, and data model, which are described respectively in the following subsections.

5.2 Conceptual Model

When developing ontologies, the conceptual model represents the structure that clarifies the expected artifact to shape the final deliverables. The conceptual model also serves as the meta-model of the structure of the ontology under construction. It concerns how the ontological primitives are arranged in the ontology. Based on best practices, the conceptual model of this research is presented in Figure 5.1.

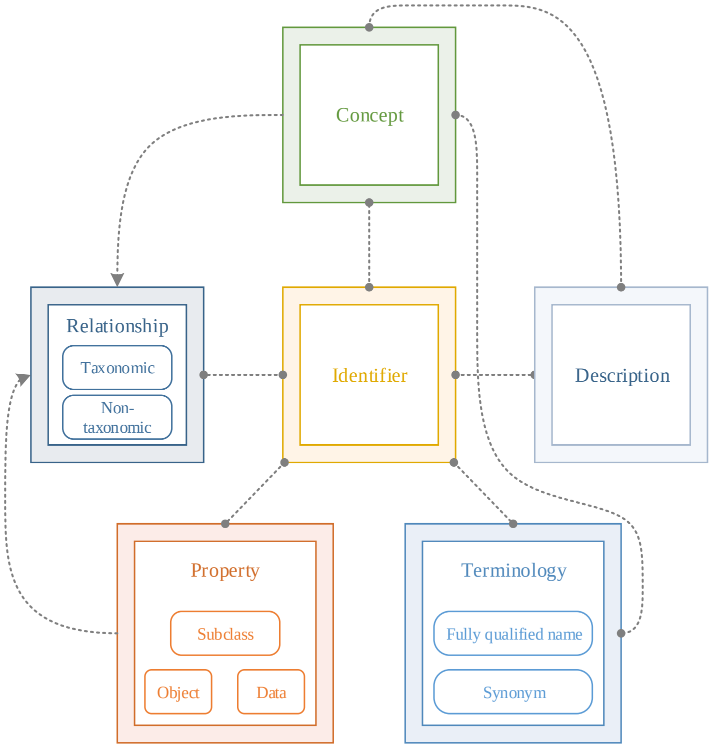


Figure 5.1 Conceptual Model

A unique internationalized resource identifier (IRI) is allocated for each concept. Therefore, a concept is uniquely identified by its IRI but not by its name. Also, for each term, property, relationship, and description, a unique identifier will be assigned to differentiate every item. A detailed specification of various identifiers is presented in Table 5.1.

Table 5.1 Specifications of identifiers

Identifier types	Specifications
Identifier	The only and unique numeric code for identifying an item
Concept identifier	The identifier of any concept
Description identifier	The identifier of any description
Destination concept identifier	A relationship consists of a source concept, a relation type, and a destination concept. When a concept in a relationship is a destination concept, its identifier is referred to as destination concept identifier
Terminology identifier	The identifier of any terminology
Relation type identifier	The identifier of any relation type
Relationship identifier	The identifier of any relationship
Source concept identifier	A relationship consists of a source concept, a relation type, and a destination concept. When a concept in a relationship is a source concept, its identifier is referred to as source concept identifier

A concept is a class, a category, a collection, a type of object, or a kind of thing. Concept depicts the fundamental elements of a knowledge domain. The term used to express a concept is terminology. Different terminologies of the same concept are synonyms. The description is used for providing a textual explanation of a concept. A relationship consists of a source concept, a relation type (or property), and a destination concept. The relation type in a relationship refers to the construct whereby concepts can be linked. Subclass property, object property, and data property are three kinds of relation types. Relationship expresses how two concepts are related, including taxonomic relationships and non-taxonomic relationships. Figure 5.2 illustrates the conceptual model in a graphical form.

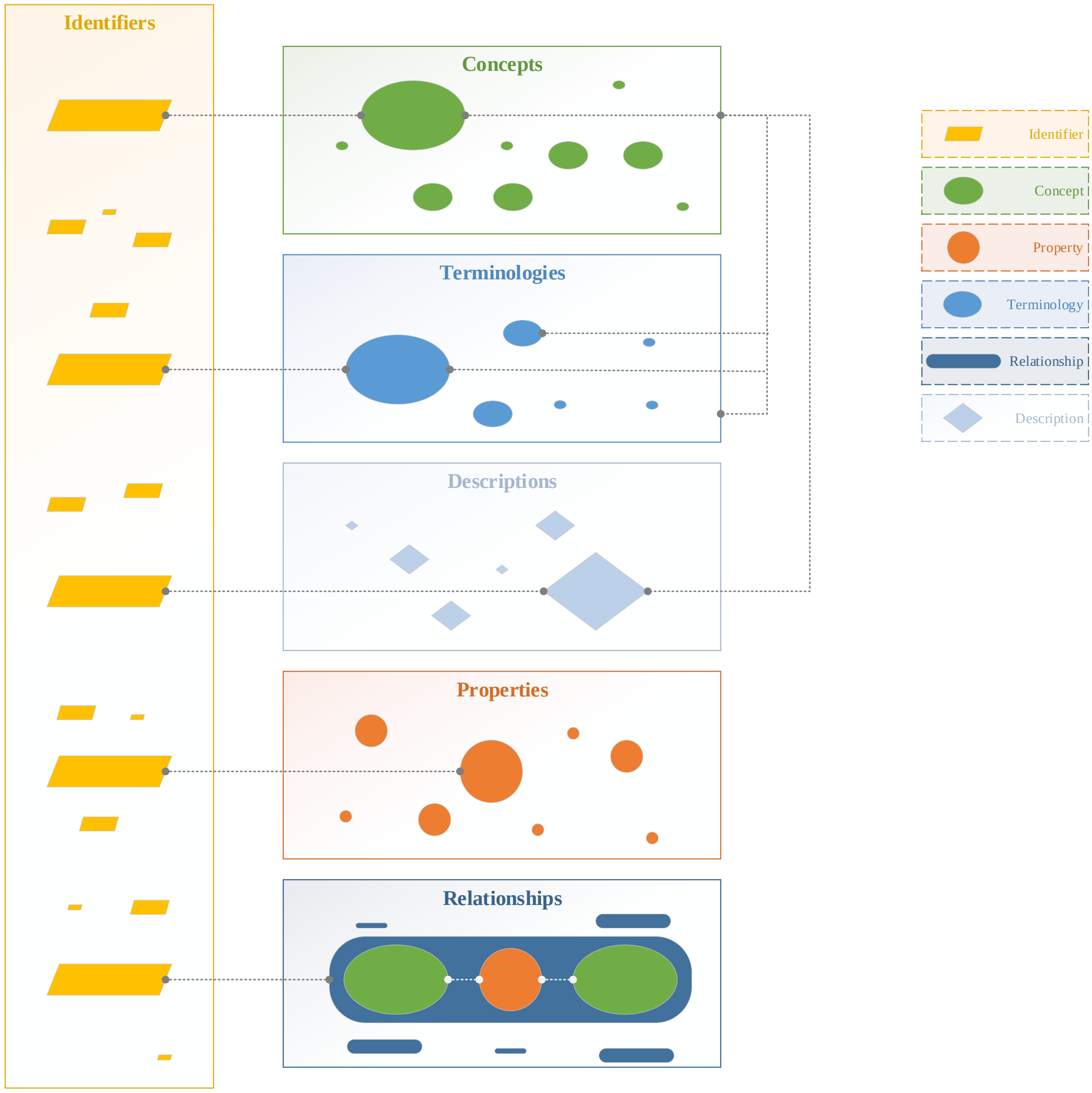


Figure 5.2 A graphical representation of the conceptual model

There are six pools in Figure 5.2, representing identifiers, concepts, terminologies, descriptions, properties, and relationships, respectively. Each yellow rhomboid is an identifier, which has one to one correspondence between identifiers and the other five elements. Concepts, presented in green ellipses, link with terminologies (blue ellipses) and descriptions (Grey diamonds). Properties are marked as orange circles. As can be seen from the navy rounded rectangle (a relationship), a relationship consists of two green ellipses and one orange circle, which means two concepts linked by one property. The graphical representation in Figure 5.2 presents the abstract conceptual model in a more vivid way.

5.3 Logical Model

Unlike the conceptual model, the logical model of an ontology focuses on explaining the logical restrictions between the different elements of an ontology. Figure 5.3 shows the logical model of the ontology that was built.

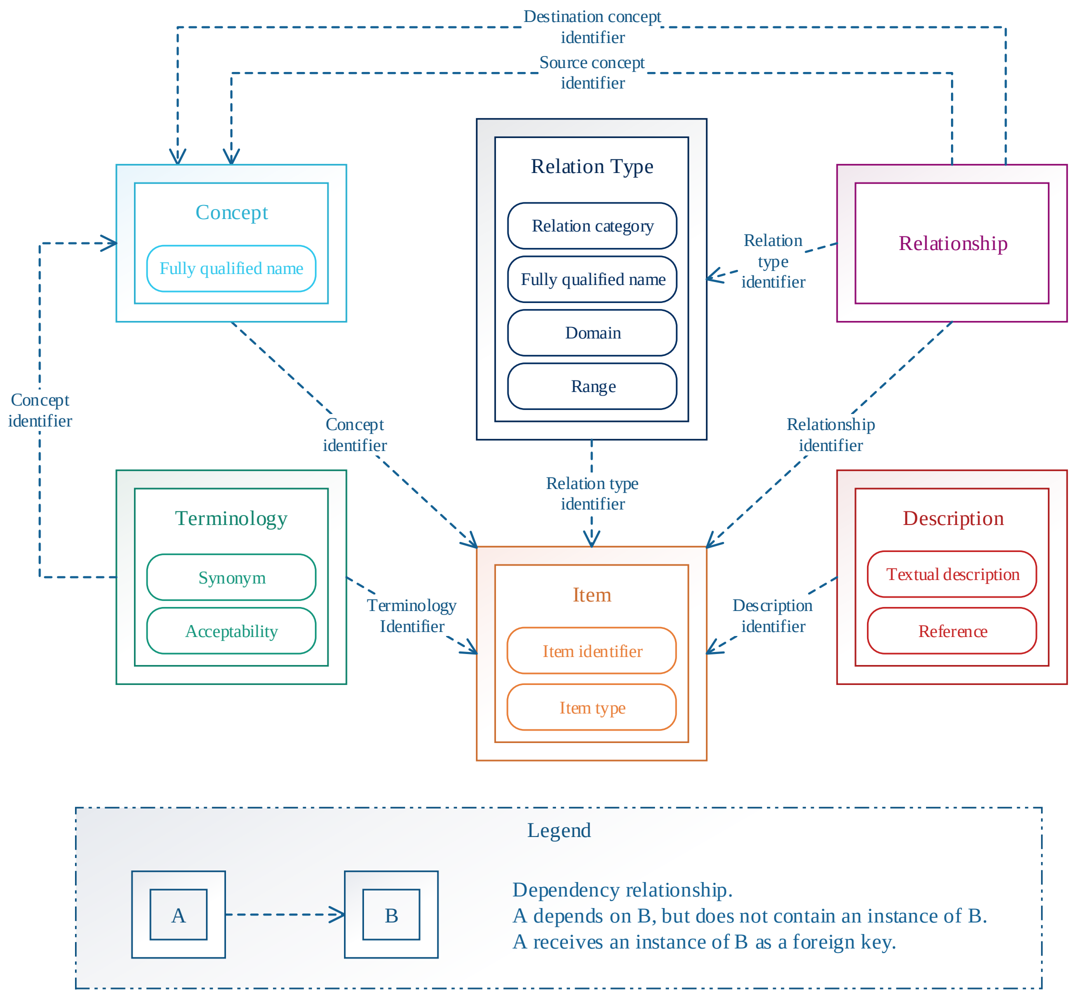


Figure 5.3 Logical model of the ontology

As can be seen from the logical model, all the elements of an ontology expect for the identifier are further abstracted into a class called item. The identifiers become the bond that links all the other elements together. In each of the classes, for example, the relation type (or property), primary attributes are listed, such as relation category, fully qualified name, domain, and range. These attributes are indispensable for developing the SE ontology. They also play a supporting role when looking at the ontology from the data model perspectives, which will be discussed in the next section. The legend in Figure 5.3 shows the dependency relationship that is very commonly used by UML when generating logical models. Take terminology and concept as an example. The concept identifier indicates that terminologies depend on concepts and receives an instance of concepts as a foreign key. That is to say, all the terminologies must link with at least one concept, with the concept identifier being the bridge connecting these two.

Concept hierarchy is an additional output, but a critical element in an ontology, when developing the taxonomic relationships. It can be automatically generated when adding new concepts to the ontology by using subclass properties. The concept hierarchy is also used to demonstrate the taxonomy of a knowledge domain. However, Figure 5.3 cannot express this important characteristic directly. Therefore, an independent notation is portrayed in Figure 5.4 to describe the logical model of the taxonomic relationship, especially the multiple parenting feature in ontologies.

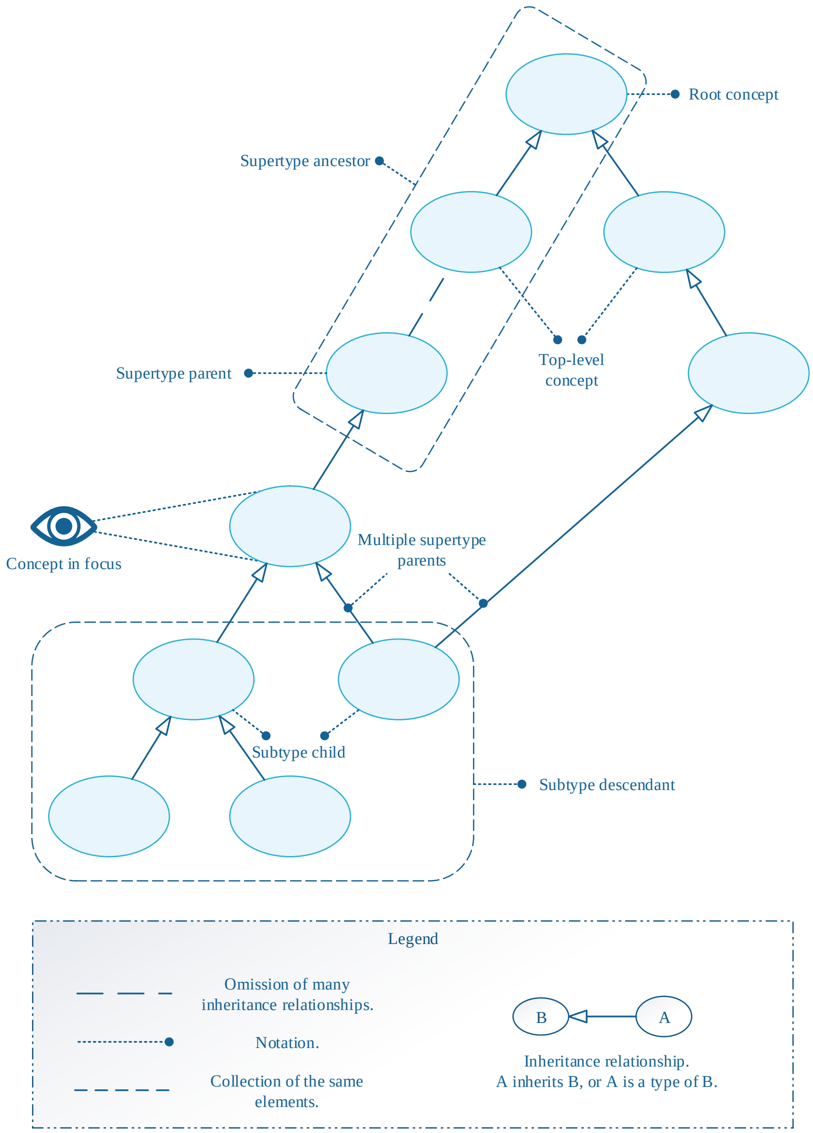


Figure 5.4 Logical model of multiple parenting feature of ontologies

One of the significant features of ontologies is to enable multiple inheritance relationships between concepts (see Figure 5.4). Looking from the perspective of the eye, Figure 5.4 describes all the possible taxonomic relationships that a concept in focus could have. First, this concept is directly linked by its supertype parent to the concept hierarchy. Its direct descendant is called subtype child. All of the subtype descendants inherit the relationships that their supertype parent has. With the level of the concept hierarchy increasing, the subtype child depicts the knowledge domain towards a more and more specialized degree, which is commonly understood as specialization in UML. It is worth noting that a concept can have multiple supertype parents. All the concepts are under a group of top-level concepts that are the fundamental categories or the top-level generalization of a knowledge domain.

In this study, one of the primary objectives is to define the top-level concepts of the SE body of knowledge, which needs to be comprehensive to contain all kinds of SE knowledge and differ the boundaries of each classification.

5.4 Data Model

The data model of the ontology is to clarify the data structure for storing and presenting all the information possessed by the ontology. First of all, as the core function of the ontology, the terminology must be captured in a systematic way. Table 5.2 shows the pre-defined data structure for storing multiple terminologies for a concept.

Table 5.2 Data model of terminologies corresponding to a concept

	<Concept identifier>			
Concept and terminology	<Fully qualified name>			
		<Terminology identifier>	<Synonym>	Preferential
	Terminology	<Terminology identifier>	<Synonym>	Optional
		<Terminology identifier>	<Synonym>	Optional

A concept can be uniquely identified by its concept identifier. However, showing identifiers in ontologies is not ideal in terms of good interoperability. Therefore, a fully qualified name is designed to represent the concept and increase readability. Different terms for expressing the concept are captured as synonyms of the aforementioned concept, each of which is assigned with a unique terminology identifier. In addition, a preferential level of usage is attached to each terminology. Some of the terminologies are preferential in conveying the concept, while others are optional with lower preference.

A fully qualified name is the full, formal, and specified name of a concept, which is not necessarily the most commonly used one. However, it must be re-recorded in the terminology, which can be marked as preferential or optional. No abbreviation can appear in the fully qualified name.

All the names appearing in the terminology are called synonyms to the fully qualified name, which presents the same concept in different terminologies.

The data model provides a mechanism that enables SE knowledge to be represented, even when a single concept does not capture the required level of detail. This is important as it enables a wide range of SE knowledge to be captured in a record, without requiring the terminology to include a separate concept for every detailed combination of ideas that may potentially need to be recorded.

Precoordinated expressions are expressions that represent the meaning of individual concepts that are predefined in the ontology. Besides the unique concept identifier and descriptions, each concept also has a formal logic definition represented by a set of defining relationships to other concepts. Table 5.3 illustrates that an item can be represented by a single identifier, with or without an accompanying human-readable term. It also illustrates the defining relationships of the concept identified in the expression. This is the precoordinated definitional knowledge that is conveyed by this expression.

Table 5.3 also shows expressions that contain two or more concept identifiers, which are referred to as postcoordinated expressions. Postcoordination combines concepts and allows more detail to be added to the meaning represented by a single concept. A postcoordinated expression is not just a list of concept identifiers; it follows a set of rules that mimic the way attributes and values are used to define the SE concepts.

Table 5.3 Precoordinated expression and postcoordinated expression rules and grammar

Grammar	Symbol		Explanation	Example
<identifier> "<label>"	Double quotation marks	""	The label quoted in the double quotation marks is the label of the identifier, such as a fully qualified name, a synonym, a property name.	R7RnJE2bqOxvcOWbQ7oxtqA "validation process"
<identifier> "<label>": <refinement>	Colon	:	A refinement can be attached to an item, detailing the definition, description, or relationships of the item.	R7RnJE2bqOxvcOWbQ7oxtqA "validation process": <refinement>
<identifier 1> "<label 1>": <identifier 2> "<label 2>" = <identifier 3> "<label 3>"	Equal sign	=	In a refinement of a relationship, the source concept is linked by a property to a destination concept. The destination concept can be understood as the value of the property, which is after an equal sign in the data model.	R7RnJE2bqOxvcOWbQ7oxtqA "validation process": RRX5rediZc8okKYSepNubM "has input" = RzTb9NXhEJFCxjAI0Dp5vY "system to be validated"
<identifier 1> "<label 1>": <identifier 2> "<label 2>" = <identifier 3> "<label 3>"; <identifier 4> "<label 4>" = <identifier 5> "<label 5>"	Semicolon	;	When adding more than one refinement to the same concept, each refinement can be separated from other, divided by a semicolon to mean "and".	R7RnJE2bqOxvcOWbQ7oxtqA "validation process": RRX5rediZc8okKYSepNubM "has input" = RzTb9NXhEJFCxjAI0Dp5vY "system to be validated"; R9EZzi0FswU1YZEyY6aXBpi "outputting" = RCIPzJDujiBBU6YdSb1a749 "validation report"
<identifier 1> "<label 1>": {<identifier 2> "<label 2>" = <identifier 3> "<label 3>"; <identifier 4> "<label 4>" = <identifier 5> "<label 5>"}	Curly braces	{}	All the refinements can be regarded as a whole when operating the concepts.	R7RnJE2bqOxvcOWbQ7oxtqA "validation process": {RRX5rediZc8okKYSepNubM "has input" = RzTb9NXhEJFCxjAI0Dp5vY "system to be validated"; R9EZzi0FswU1YZEyY6aXBpi "outputting" = RCIPzJDujiBBU6YdSb1a749 "validation report"}
<identifier 1> "<label 1>": {<identifier 2> "<label 2>" = <identifier 3> "<label 3>"; (<identifier 4> "<label 4>" = <identifier 5> "<label 5>"; <identifier 4> "<label 4>" = <identifier 6> "<label 6>")}	Round braces	()	When assigning different values to the same property,	R7RnJE2bqOxvcOWbQ7oxtqA "validation process": {RRX5rediZc8okKYSepNubM "has input" = RzTb9NXhEJFCxjAI0Dp5vY "system to be validated"; (R9EZzi0FswU1YZEyY6aXBpi "outputting" = RCIPzJDujiBBU6YdSb1a749 "validation report"; R9EZzi0FswU1YZEyY6aXBpi "outputting" = RDoP67VSJ80OsqdfTbSLZpY "validation record")}

There are several valid ways to represent and store ontological knowledge. However, to support interoperability, this section has specified a standard data model with compositional grammar form that is both human-readable and computer processable.

5.5 Conclusion

In this chapter, three novel models are developed and presented to depict the SE ontology from three facets, namely conceptual model, logical model, and data model. These models are created based on ontology definitions and theoretical foundations. They seem universal; however, they are designed to specify the SE body of knowledge. The definition of formal ontologies should consider these three facets.

These models serve as the overarching structure in which the SE knowledge can be gathered and structured. The conceptual model illustrated by Figure 5.1 and Figure 5.2 specifies the key elements in an ontology, i.e., the definitions of identifiers, concepts, terminologies, descriptions, properties, and relationships. As for the logical level, Figure 5.3 lays the foundation of the logical relations between the different elements in the conceptual model, and Figure 5.4 further elaborates on the multiple parenting hierarchical relations in the SE ontology. The data model includes the definition of precoordinated and postcoordinated expression rules and grammar that enable the ontology represented by compositional grammar form that is both human-readable and computer processable.

The proposed ontology learning approach and the developed ontology models have been introduced. The next chapter focuses on applying the proposed approach and the models in a real case study to show the feasibility of these research endeavors.

CHAPTER 6 CASE STUDY

6.1 Introduction

This chapter presents a complete case study by using the approach and models proposed in Chapter 4 and Chapter 5. It deals with two outputs of the research as defined in the IDEF0 model, namely a formal ontology deriving from authoritative SE standards, and a visualization of the SE ontology by sophisticated ontology tools.

The case study is conducted to show how the ontology learning approach and the ontology models are used in practice. As the description of the ontology learning approach in the last chapter is not presented by an actual SE standard, some details may be lost. Therefore, in this chapter, the case study chooses a real SE standard and practices all the stages of the ontology learning approach to generate an SE ontology.

Section 6.2 records the entire process of the application of the ontology learning approach using the INCOSE SE handbook. Furthermore, the ontological primitives generated from the ontology learning approach are populated into the ontology models.

Instead of describing how the model was populated, Section 6.3 focuses on the description of the actual learned SE ontology, detailing its terminology, concepts, and relations. Moreover, the developed ontology shows its powerful ability to re-structure and represent the SE life cycle processes in a dynamic way.

Section 6.4 provides an application scenario of the SE ontology in obtaining an IDEF0 model of the SE life cycle processes. It is a more accurate and dynamic model than the N^2 diagram provided by the INCOSE SE handbook. It presents the A-0 diagram and the A0 diagram in the IDEF0 model. The detailed decomposition of the IDEF0 model is attached in Appendix 1.

6.2 Application of the Ontology Learning Methodology

This section presents a complete application of the proposed ontology learning methodology for learning an SE ontology from the INCOSE SE handbook. To elaborate on the details, it first provides a brief introduction to the INCOSE SE handbook. Then, it presents how each of the stages in the methodology is applied through the case study.

6.2.1 Introduction to the INCOSE SE Handbook

The INCOSE SE handbook (INCOSE, 2015a) provides an authoritative reference to the SE discipline in terms of theory and practice. It covers the SE core body of knowledge and relates to other international standards.

The INCOSE handbook describes critical process activities performed by systems engineers and other engineering professionals throughout the life cycle of a system. The handbook itself is a tool and portfolio of system concepts, aiding practitioners by providing a solid background on system thinking, life cycle concepts, the system of systems management and complex systems, etc. The handbook also acts as a reference for the discipline of SE in general, including standards, models, life cycle stages, processes, and their built-in tools and methods. According to Forsberg and Roedler (2011), the series of the INCOSE SE handbook all become the primary reference to many organizations for creating internal SE process documents.

However, there are also some limitations that restrict the application of SE best practices. It contains sufficient terminologies that describe the SE domain but are often intertwined and ambiguous (Chourabi *et al.*, 2010). With continuous revisions, the same concepts are referred to by different terms, and the linkages between the two editions are complicated to identify (Di Maio, 2011). The handbook has been translated into different languages and adapted for different cultures. The translation also causes misunderstanding and misinterpretation of the meanings of important SE concepts.

Thereby, although it is a world-widely accepted standard, it needs improvement in terms of the level of interoperability. However, due to its rich terminologies, it is ideal for acting as the SE corpus to derive an SE ontology. The following sections will show how the knowledge in the INCOSE SE handbook is extracted to form an SE ontology to address the problems.

6.2.2 Stage I: Document Collection and Pre-processing

To begin with, the latest English edition (4th version) of the INCOSE SE handbook was obtained. It was published in 2015. Therefore, the electronic file of the English version in PDF was obtained. However, when using this handbook as the SE corpus, a pre-processing of the original text is necessary. The table of contents, Chapter 1, and the appendices were eliminated to clean the text since they do not contribute much to the corpus; in contrast, they might affect the results.

The rest of the INCOSE SE handbook has a conventional layout, as shown in Figure 6.1 and Figure 6.2. They show the general layout of the handbook, which contains titles, subtitles, paragraphs, bullet points, tables, figures, pages, headers, and footers.

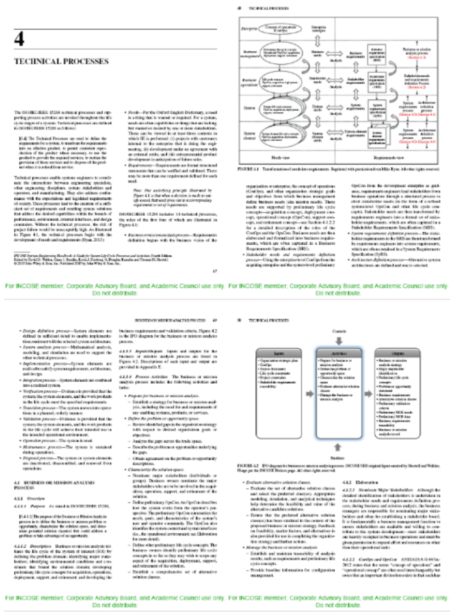


Figure 6.1 An epitome of the INCOSE SE handbook

In order to obtain a clean text, this electronic document (PDF) was converted into a text file (TXT). It is worth noting that case sensitivity was retained since some of the abbreviations are important to the SE community. A sample of the TXT file is shown in Figure 6.3.

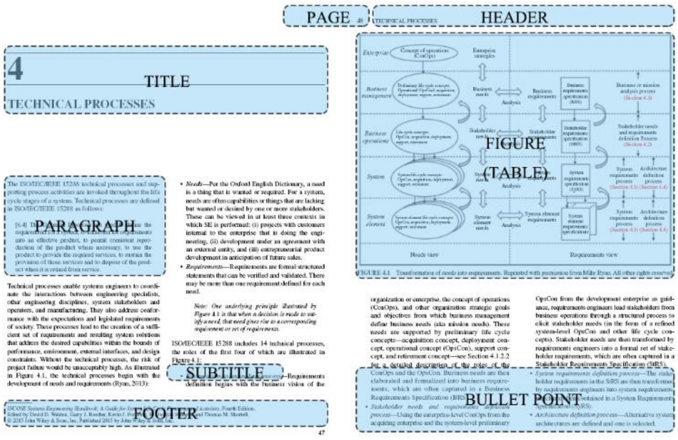
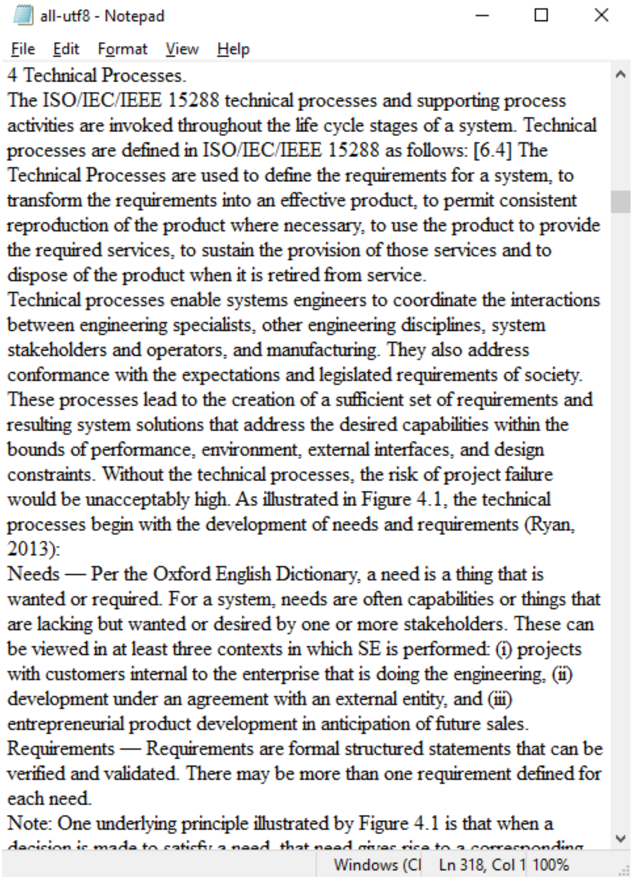


Figure 6.2 General structure of a page in the INCOSE SE handbook



4 Technical Processes.
The ISO/IEC/IEEE 15288 technical processes and supporting process activities are invoked throughout the life cycle stages of a system. Technical processes are defined in ISO/IEC/IEEE 15288 as follows: [6.4] The Technical Processes are used to define the requirements for a system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary, to use the product to provide the required services, to sustain the provision of those services and to dispose of the product when it is retired from service.
Technical processes enable systems engineers to coordinate the interactions between engineering specialists, other engineering disciplines, system stakeholders and operators, and manufacturing. They also address conformance with the expectations and legislated requirements of society. These processes lead to the creation of a sufficient set of requirements and resulting system solutions that address the desired capabilities within the bounds of performance, environment, external interfaces, and design constraints. Without the technical processes, the risk of project failure would be unacceptably high. As illustrated in Figure 4.1, the technical processes begin with the development of needs and requirements (Ryan, 2013):
Needs — Per the Oxford English Dictionary, a need is a thing that is wanted or required. For a system, needs are often capabilities or things that are lacking but wanted or desired by one or more stakeholders. These can be viewed in at least three contexts in which SE is performed: (i) projects with customers internal to the enterprise that is doing the engineering, (ii) development under an agreement with an external entity, and (iii) entrepreneurial product development in anticipation of future sales.
Requirements — Requirements are formal structured statements that can be verified and validated. There may be more than one requirement defined for each need.
Note: One underlying principle illustrated by Figure 4.1 is that when a

Figure 6.3 The electronic file after conversion to plain text

6.2.3 Stage II: Lexical Analysis through NLP

Next, the text file was brought to the lexical analysis stage, with the help of NLP techniques. The NLP drew support from the platform Spyder 3.3.2 to run Python 3.7.1 to invoke the NLTK packages 3.4. The TXT file was loaded first to the Python environment and then segmented into sentences (nltk.sent_tokenize(raw text)). Next, a loop command was made upon each sentence to further segment the sentence into words (nltk.word_tokenize(sentence)). Afterward, the Penn Treebank tagset was used to assign part-of-speech tags to each token in a sentence (nltk.pos_tag(sentence)), resulting in a list of tagged tokens. The changes made to a sample sentence along with the above operations are illustrated in an example as Table 6.1.

Table 6.1 Changes in the text after applying relevant NLTK packages

NLTK package	Changes in the text
nltk.sent_tokenize(raw text)	'The systems engineer must continually distinguish between systems in the real world and system representations.'
nltk.word_tokenize(sentence)	['The', 'systems', 'engineer', 'must', 'continually', 'distinguish', 'between', 'systems', 'in', 'the', 'real', 'world', 'and', 'system', 'representations', '.']

NLTK package	Changes in the text
nltk.pos_tag(sentence)	[('The', 'DT'), ('systems', 'NNS'), ('engineer', 'NN'), ('must', 'MD'), ('continually', 'RB'), ('distinguish', 'VB'), ('between', 'IN'), ('systems', 'NNS'), ('in', 'IN'), ('the', 'DT'), ('real', 'JJ'), ('world', 'NN'), ('and', 'CC'), ('system', 'NN'), ('representations', 'NNS'), ('.', '.')]

The results in Table 6.1 are the final outcomes of the repeated training of the off-the-shelf tagger. In the beginning, the tagger had a relatively low accuracy score, meaning it did not always assign the proper tag to the token. Then, the tagger was trained by comparing the outcomes to the tags that a human expert would assign to enhance its performance. Also, a gold standard test was used to evaluate the tagger to improve its accuracy further. The rule is that the tagger is regarded as being correct if the tag's guess for a given word is the same as the gold standard tag. This principle was implemented on a set of randomly chosen sentences from the corpus. After they reached the standard level, the tagger was trained through the gold standard to tag the rest of the corpus. The output from this stage is an SE corpus with part-of-speech tags annotated to each token.

6.2.4 Stage III: Ontology Components Extraction

Once the annotated SE corpus was obtained, the ontology components extraction was performed. A series of initial findings were obtained, such as the identification of the high-frequency terms and the detection of a set of key verbs or verb phrases. The high-frequency terms are the foundation of making up the ontology concepts (or classes). The verb and verb phrases are usually considered as the relations that link such concepts.

In Figure 6.4, a word cloud of the high-frequency terms that appear more than 100 times is shown.

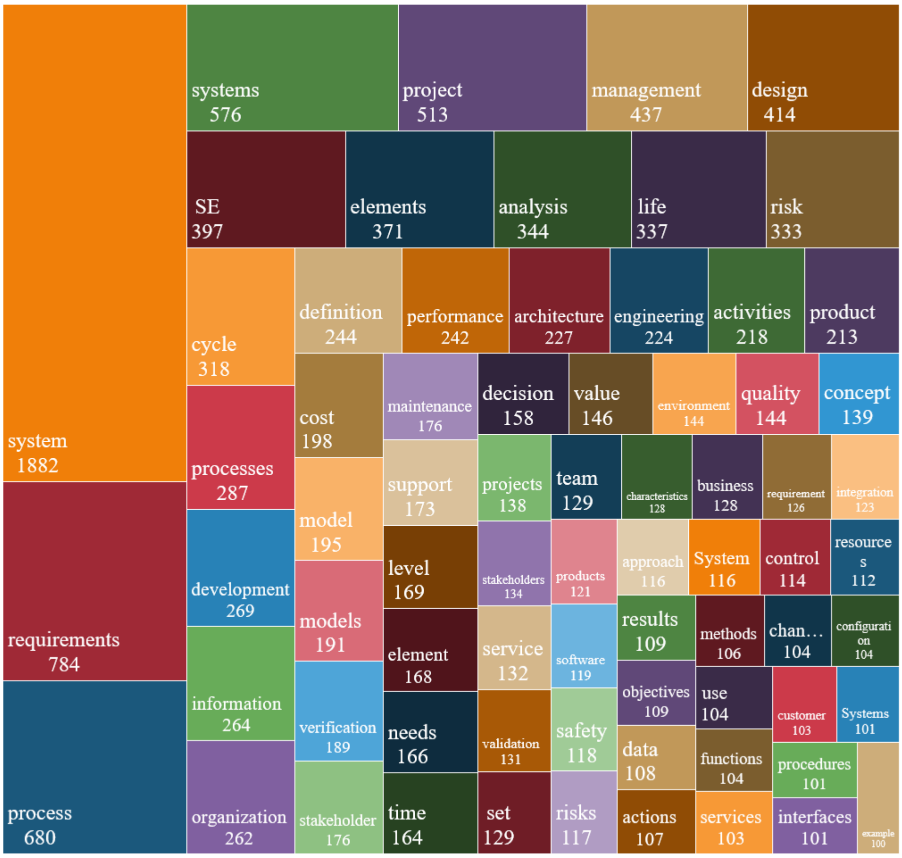


Figure 6.4 Word cloud of high-frequency terms

The size of the area in Figure 6.4 represents how often each term appears. In fact, more than 5,000 different kinds of noun terms were discovered. Although it is not surprising that the term ‘system’ is mentioned the most often, its occurrence only accounts for 4.3% percent of the total number of nouns. When taking all the noun terms that have more than 100 occurrences into account, their accumulative occurrence adds up to 16,327 times, which makes up 37.2% of the total number of nouns. This finding reveals that although the handbook uses many kinds of terms (more than 5,000 for noun), some are used only once or twice, and others are used repeatedly. If a suitable threshold is chosen, it is possible to use the least number of terms to describe the complete SE body of knowledge, decreasing confusion, and ambiguity.

For this case study, the initial threshold was set at 100 and then gradually reduced to include more terms in the core term set, i.e., the core set of terms for SE knowledge. Through repeated experiments, the final threshold was set as 20, meaning that all the terms whose occurrences reach 20 were included in the core term set. This set contains the most key terms for SE and the least useless terms. Figure 6.5 shows the inclusions of the terms that are used to define SE life cycle processes. The size of the bubble represents the frequency of occurrence. By contrast, the term system appears nearly 2,000 times, whereas tailoring only occurs 22 times. The position of the bubble from right to left reflects its level of increasing importance. It can be seen that the key terms cluster on the left with larger bubble sizes, which verifies that by setting up an appropriate threshold, the core term set will contain the least number of terms but will represent the complete SE body of knowledge.

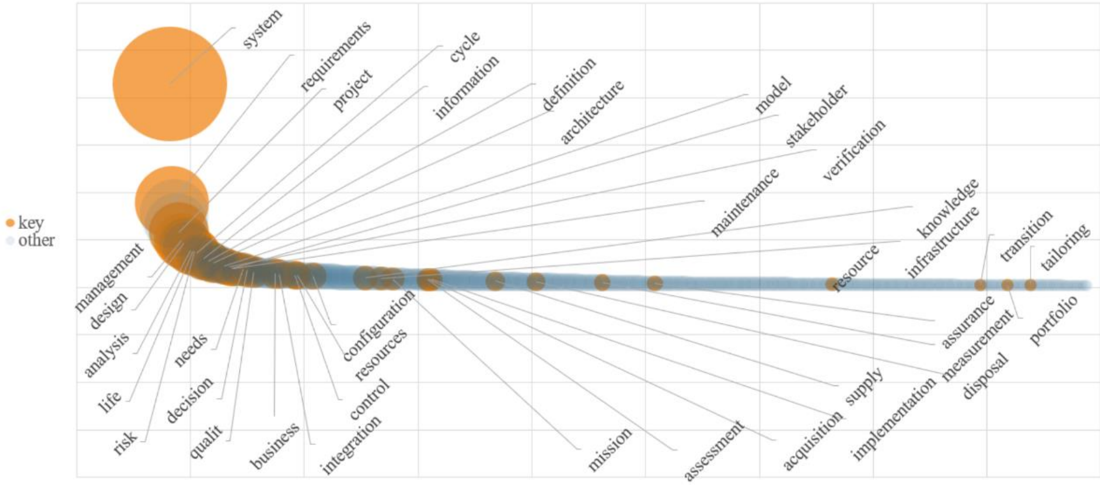


Figure 6.5 Key terms obtained with a threshold of 20

Moreover, essential terms were used to seek key noun phrases that are usually as important concepts in this domain. By using regular expressions in Python, a multistage chunk grammar was developed. It contains recursive rules, not only for chunking noun phrases, but also for prepositional phrases, verb phrases, and sentences. The extraction of key phrases provides the creation of ontologies with ontological primitives, i.e., the components of an ontology. For noun phrases to become concepts, direct one-to-one mapping was performed. Figure 6.6 shows an excerpt of the most frequent noun phrases.

It can be found from Figure 6.6 that the phrases that comprise the SE knowledge domain are mostly fundamental concepts, such as system elements, system requirements, system engineers. Nevertheless, it can also be found that some high-frequency phrases are used to ensure clarity in writing, such as process activities, appendix E, and following activities. Therefore, the extracted list of noun phrases was filtered to make it contain only fundamental SE concepts.

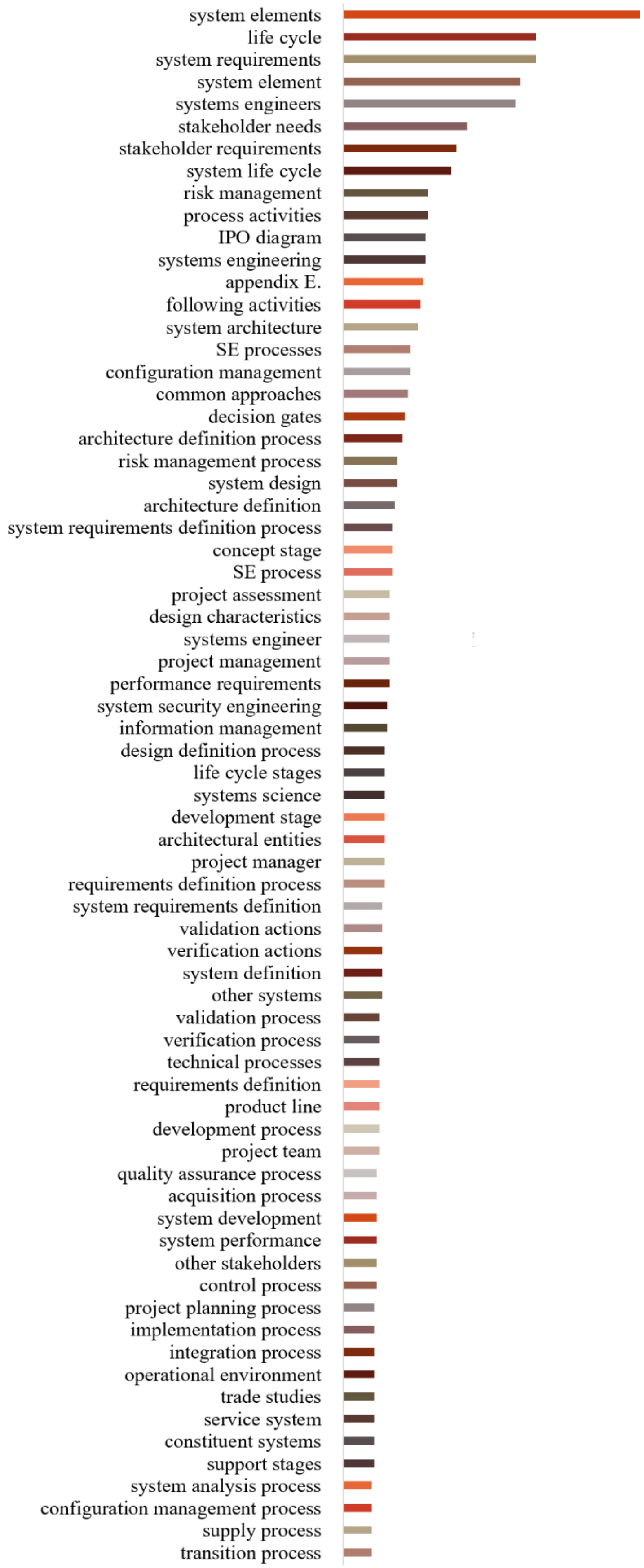


Figure 6.6 High-frequent noun phrases

Verb and verb phrases were also detected by the chunk grammar. Therefore, each sentence was analyzed and chunked to split the different parts of the sentence structure. Eventually, each sentence can be presented in a tree view, as shown in Figure 6.7, as an example. This sentence is also the one in Table 6.1. However, unlike before, the SE corpus was structured with further semantics, leveraging a foundation for ontology creation in the next phase.

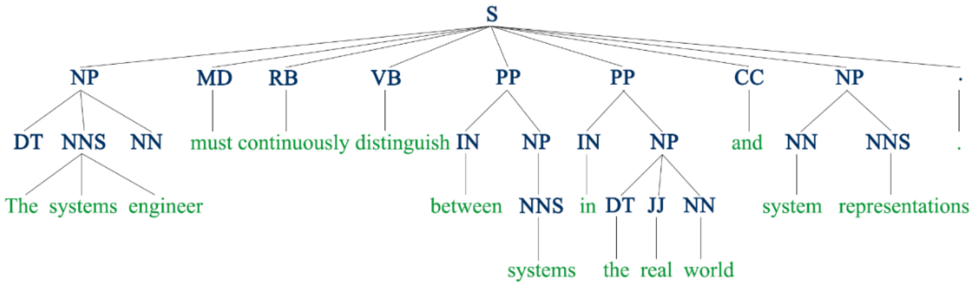


Figure 6.7 A tree view of sentence structure

6.3 The SE Ontology

This section elaborates on the SE ontology in terms of the terminology it includes, the top-level concepts that form the ontology, the multiple-parent taxonomic relations within the ontology, and the concept hierarchy that the SE ontology contains.

6.3.1 Terminology

From the ontology learning process, several ontological primitives were generated. They constitute the ontology by consisting of the concepts, terminologies, properties, relationships, and axioms. Noun terms and phrases constitute the basics of ontology concepts. From the ontology learning processes, various SE terminologies were extracted with frequency distributions. Starting from the terms and phrases with high frequency, terminologies were imported into WebProtégé to build the concept hierarchy. The concept hierarchy was built based on the taxonomic relationships detected in the SE corpus. For each concept, synonyms were merged based on the data model presented in Section 5.4. WebProtégé offers a function to edit concepts' names with labels.

Take the 'concept of operations' as an example (Figure 6.8).

IRI

http://webprotege.stanford.edu/RFf582I3O4QQyqQXjg7DK6

Annotations

rdfs:label	concept of operations	en
skos:prefLabel	ConOps	en
skos:definition	The ConOps is a verbal and/or graphic statement prepared for the organization's leadership that describes the assumptions or intent regarding the overall operation or series of operations of the enterprise, to include any new capability (ANSI/AIAA, 2012; ISO/IEC/IEEE 29148, 2011).	en
skos:note	The ConOps, at the organization level, addresses the leadership's intended way of operating the organization. It may refer to the use of one or more systems, as black boxes, to forward the organization's goals and objectives. The ConOps document describes the organization's assumptions or intent in regard to an overall operation or series of operations of the business with using the system to be developed, existing systems, and possible future systems. This document is frequently embodied in longrange strategic plans and annual operational plans. The ConOps document serves as a basis for the organization to direct the overall characteristics of the future business and systems, for the project to understand its background, and for the users of [ISO/IEC/IEEE 29148] to implement the stakeholder requirements elicitation.	en
skos:altLabel	concept of operations document	en

Figure 6.8 Merging synonyms

The 'concept of operations' is a commonly referred concept within the SE domain. However, most of the time, SE practitioners are inclined to refer to it as ConOps. In fact, the three different labels of 'concept of operations' are all captured in the ontology as they are synonyms. They are annotated as rdfs:label, skos:prefLabel and skos:altLabel. This annotation specifies the various names used to describe a concept, i.e., fully qualified name, preferred name, and acceptable name.

6.3.2 Top-level Concepts

Similar concepts were grouped together, and taxonomic relationships were developed between concepts to build concept hierarchy. The process is done in a bottom-up manner, as the concept hierarchy starts with the leaf nodes and branches of the lower and middle layers. The groupings of concepts were not performed arbitrarily. In fact, they are formed by bottom-up clustering according to good practices (Gangemi *et al.*, 2002; OBO Technical WG, 2019). The top-level concepts are presented in Figure 6.9.

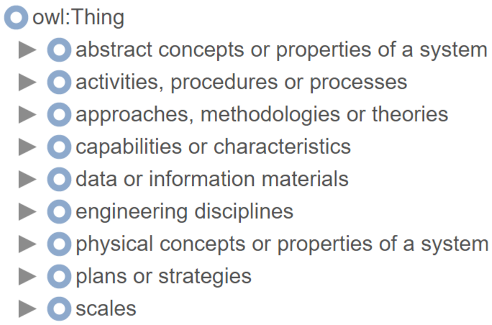


Figure 6.9 Top-level concept hierarchy in WebProtégé

6.3.3 Multiple-parent Taxonomic Relations

There are nine top-level classes that depict the SE body of knowledge. In other words, the SE concepts generated from the INCOSE SE handbook were all grouped under at least one class. In fact, an SE concept can appear under multi-parent classes. Therefore, the class hierarchy is an acyclic net structure, not a simple tree, but a poly-hierarchy. This is pre-defined in the logical model of the ontology in Section 5.3.

Figure 6.10 shows an example of multiple-parent taxonomic relationships.

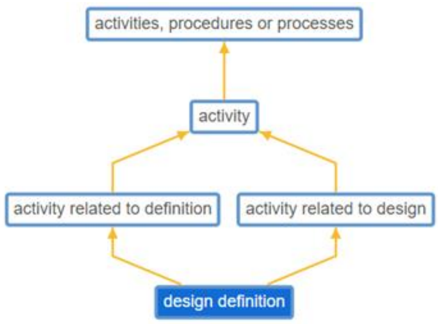


Figure 6.10 An example of the multiple-parent taxonomic relationships

'Design definition' is a concept in the SE knowledge domain. It belongs to both the 'activity related to definition' and the 'activity related to design'. The 'activity related to definition' and the 'activity related to design' all belong to the class 'activity', which is under the top-level hierarchy 'activities, procedures or processes'.

The top-level classes with their descendants will be briefly described from Section 6.3.4.1 to Section 6.3.4.9.

6.3.4 Concept Hierarchy

The following subsections present the concept hierarchy that frames the SE ontology.

6.3.4.1 Abstract Concepts or Properties of a System

As the default sorting of concepts was set as the alphabet order, the first top-level class is 'abstract concepts or properties of a system'. Figure 6.11 shows an excerpt of this class and its descendants.

Figure 6.11 An excerpt of abstract concepts or properties of a system

Take 'measurement needs' as an example. 'Measurement needs' include 'measure of effectiveness needs,' 'measure of effectiveness needs', 'technical performance measures needs', 'project performance measures needs', and 'organizational process performance measures needs'. The INCOSE SE handbook differentiates needs from requirements. However, through the development of the SE ontology, it is found that sometimes needs are called requests under certain circumstances. For example, 'acquisition need' is the identification of a need that cannot be met within the organization encountering the need or a need that can be met in a more economical way by a supplier. A 'request for supply' is the request to an external supplying organization to propose a solution to meet a need for a system element or system (product or service). In essence, they are both a kind of needs. Therefore, they are both a subclass of needs.

6.3.4.2 Activities, Procedures or Processes

The second top-level class is related to 'activities, procedures, or processes' within SE. Figure 6.12 shows the secondary and primary decomposition of the classification of the procedure and process.

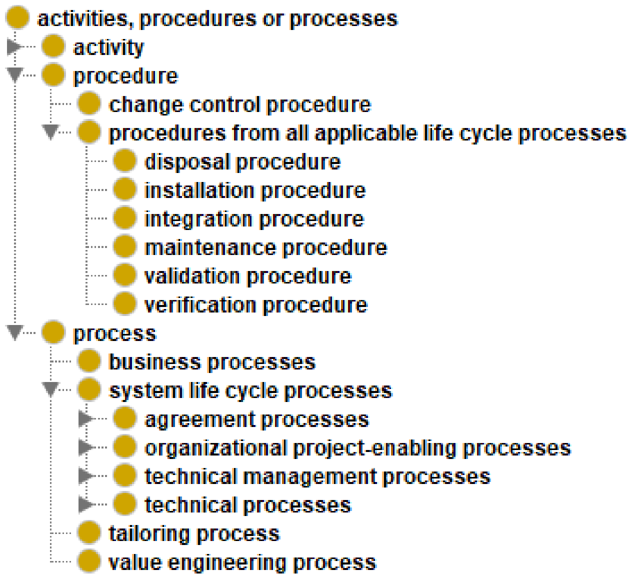


Figure 6.12 An excerpt of activities, procedures or processes

Procedures in the SE domain represent a set of actions and techniques performed with specific enablers. The SE life cycle processes contain 'disposal procedure', 'installation procedure', 'integration procedure', 'maintenance procedure', 'validation procedure', and 'verification procedure'. However, through the ontology learning process, a concept called 'change control procedure' is found in the INCOSE SE handbook. The handbook does not provide a clear definition of this procedure, but it is introduced by ISO/IEC 26514:2008, meaning the actions that are taken to identify, document, review, and authorize changes to a software or documentation product that is being developed. Therefore, it can be concluded that the ontology learning approach can detect knowledge that is implicitly presented in the SE standards. However, the INCOSE SE handbook alone cannot portray the entire universe of the SE domain. The SE ontology developed in this research does not intend to provide a complete picture of the SE body of knowledge either, as ontologies can always be enriched or refined. The aim of the research is to provide a universal approach that can take advantage of and extract ontological knowledge from existing SE standards. The goal of the developed ontology is to serve as a foundation to contain more SE concepts.

6.3.4.3 Approaches, Methodologies or Theories

The third top-level class is 'approaches, methodologies or theories' in the SE domain. Figure 6.13 shows the hierarchy within this classification. This class can be enriched with more concepts, as the current extracted hierarchy is only based on the INCOSE SE handbook. The handbook does not provide much information on SE approaches, methodologies or theories. However, as an essential constituent of the SE body of knowledge, this top-level class is reserved for further investigation.

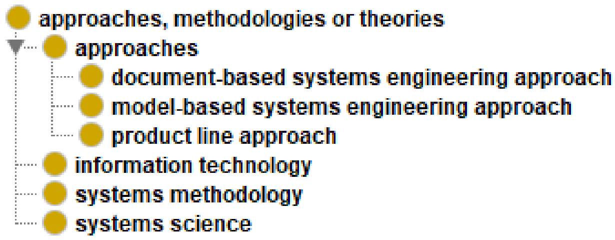


Figure 6.13 An excerpt of approaches, methodologies or theories

6.3.4.4 *Capabilities or Characteristics*

The fourth top-level class is ‘capabilities or characteristics’. Figure 6.14 presents the ‘capabilities or characteristics’ found in the INCOSE SE handbook.

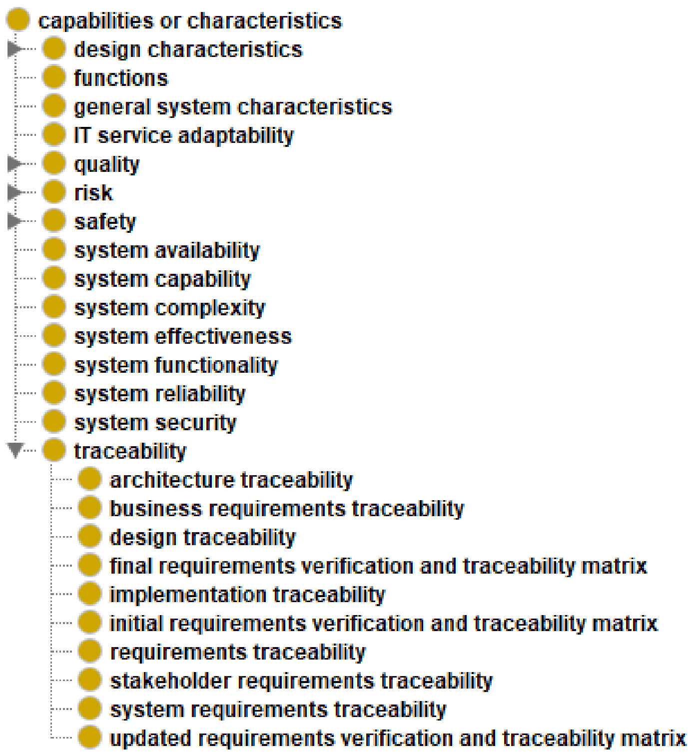


Figure 6.14 An excerpt of capabilities or characteristics

One of the capabilities that the SE life cycle processes emphasize is traceability. Various kinds of traceability were detected from the ontology learning process, such as ‘architecture traceability’, ‘business requirements traceability’, ‘design traceability’, etc.

6.3.4.5 *Data or Information Materials*

The fifth top-level class is ‘data or information materials’. This is the largest classification among the nine top-level classes. Figure 6.15 presents an excerpt of this top-level class.

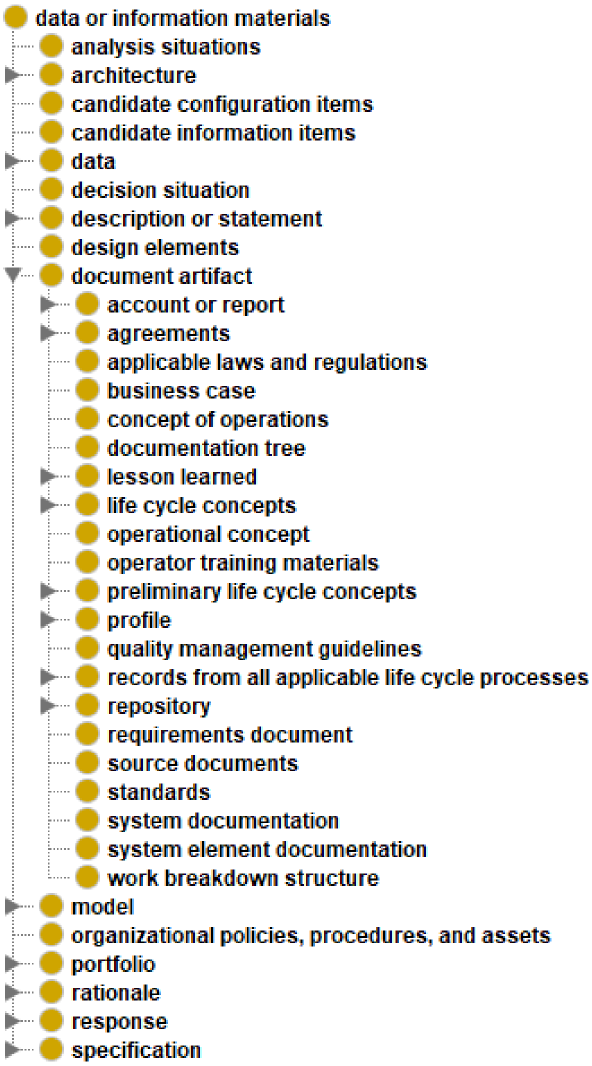


Figure 6.15 An excerpt of data or information materials

As known by many SE practitioners, when practicing SE processes, many documents are generated and transferred between processes. The variety of these document artifacts makes SE novices hard to manage. In the SE ontology, different kinds of documents are systematically classified. Figure 6.16 provides an excerpt from the 'document artifact' class in the SE ontology.

In Figure 6.16, the 'account or report' in the SE domain can be classified into reports that are from all applicable life cycle processes and other processes. Agreements are divided into two categories, 'acquisition agreement' and 'supply agreement'.

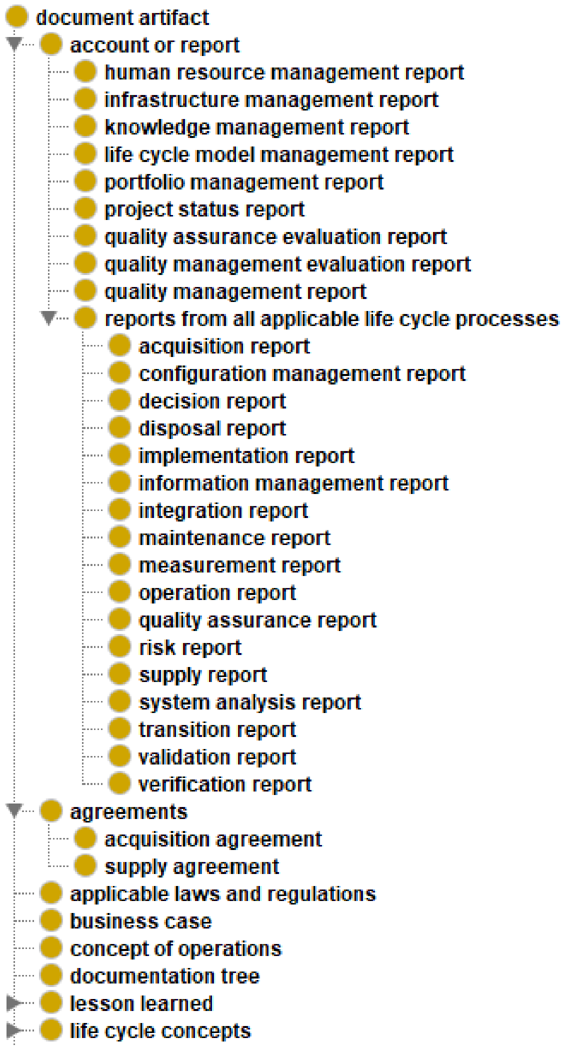


Figure 6.16 An excerpt of document artifacts

6.3.4.6 Engineering Disciplines

The sixth top-level class is ‘engineering disciplines’. Figure 6.17 shows an overview of this category. Since the INCOSE SE handbook provides general SE domain knowledge, it does not provide details about specific engineering disciplines. However, it is vital to establish an independent classification for distinguishing different engineering disciplines related to SE. Therefore, the subclasses currently under this top-level class are rather limited.

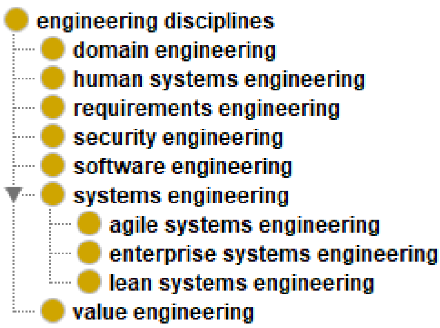


Figure 6.17 An excerpt of engineering discipline

6.3.4.7 Physical Concepts or Properties of a System

The seventh top-level class is 'physical concepts or properties of a system'. It is another important set of concepts that corresponds to 'abstract concepts or properties of a system'. Figure 6.18 shows an excerpt of this top-level class.

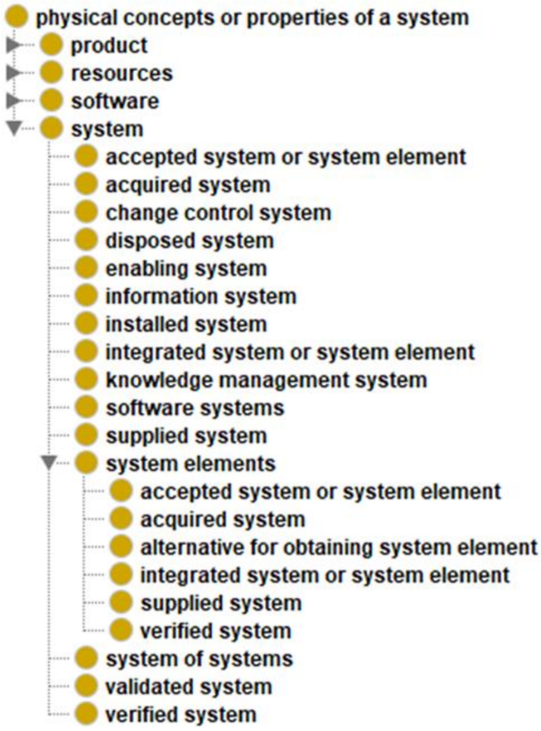


Figure 6.18 An excerpt of physical concepts or properties of a system

As can be seen from Figure 6.18, a 'system or system element' can have different statuses, such as accepted, acquired, integrated, supplied, and verified.

6.3.4.8 Plans or Strategies

The eighth top-level class is 'plans or strategies' in the SE body of knowledge. Figure 6.19 shows an excerpt of this class.

Figure 6.19 An excerpt of plans or strategies

For example, the ontology shows 23 types of strategies that appear in the INCOSE SE handbook. It can also be found that sometimes a strategy is named as a policy or plan. Summarizing them together can help unify the terminology.

6.3.4.9 Scales

The last top-level class is ‘scales’. Figure 6.20 shows an excerpt of scales found in the INCOSE SE handbook. For example, criteria belong to one kind of scale.

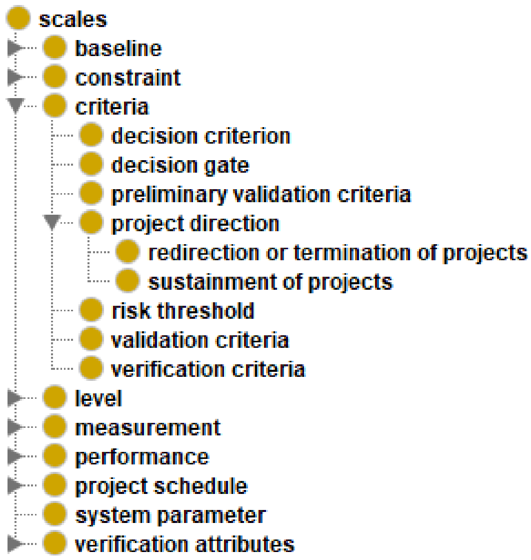


Figure 6.20 An excerpt of scales

6.4 Application of the Developed Ontology

This section presents an application scenario where the developed ontology for the SE body of knowledge is applied. The scenario is how the ontology is used to enable a dynamic and robust representation of the system life cycle processes. It begins by presenting an overview of the system life cycle processes. Then, a semantic network is proposed for connecting all the processes. Third, the ontology for the system life cycle processes is demonstrated. Next, by using the reasoning and inferring features of ontologies, the interrelations within the system life cycle processes are revealed. Fifth, it provides a reorganization of the processes by the IDEF0 process modeling method.

6.4.1 System Life Cycle Processes Overview

ISO/IEC/IEEE 15288:2015 is the international standard for systems and software engineering – system life cycle processes. This international standard establishes a defined set of processes for describing the life cycle of systems created by humans in order to facilitate communication among different stakeholders in the life cycle of a system. It also provides a normative direction regarding the tailoring of these system life cycle processes to fit different SE projects. As jointly described in the INCOSE SE handbook and ISO/IEC/IEEE 15288, the SE life cycle processes are categorized into four groups, technical processes, technical management processes, agreement processes, and organizational project-enabling processes. The four process groups and the sub-processes included in each group are depicted in Figure 6.21.

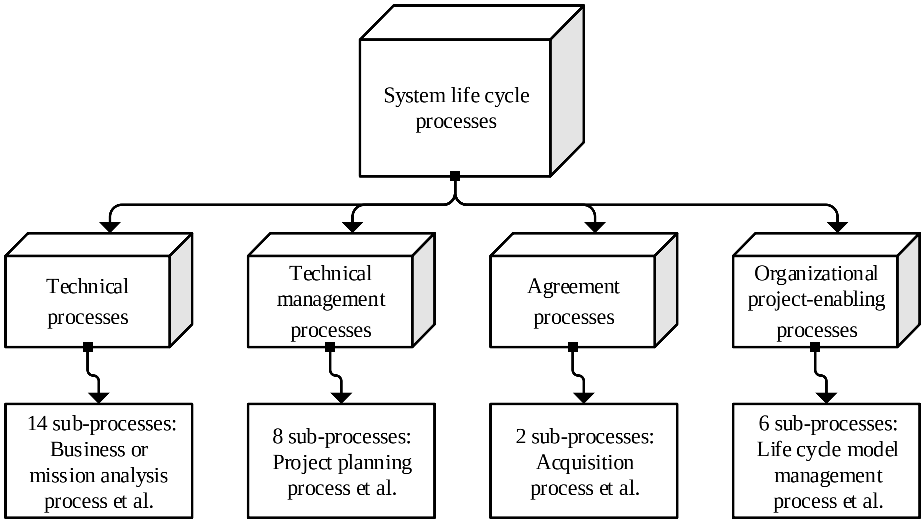


Figure 6.21 System life cycle processes per ISO/IEC/IEEE 15288:2015

While ISO/IEC/IEEE 15288:2015 provides a generic top-level process description, the INCOSE SE handbook further elaborates on the practices and activities necessary to put the processes into practice consistent with the international standard. A common format is used to describe the system life cycle processes, which is illustrated in Figure 6.22.

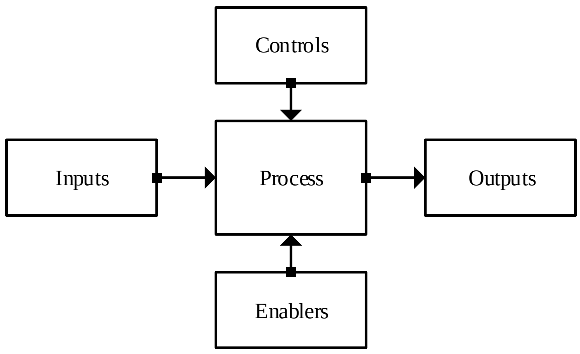


Figure 6.22 Sample of an IPO diagram

This format is named as an IPO diagram, which stands for the input-process-output diagram, and it shows the key inputs, resulting outputs, necessary controls, and essential enablers of a system life cycle process. Inputs, outputs, controls, and enablers can be regarded as indispensable elements when describing a process. Therefore, in this research, they are defined as a generalized class named system life cycle object. In other words, the input, output, control, or enabler can be regarded as specialized roles of an object playing in a system life cycle process.

6.4.2 Proposing a Semantic Network for System Life Cycle Processes

There are some system life cycle objects that act as both outputs and inputs in terms of two or more processes when the objects have sequential relationships. However, one IPO diagram can only show one life cycle process at a time. It cannot represent the whole network of which all the objects consist. A possible situation is mentioned that some objects can act both as an output and an input, which gives an example of one possible relation that is missing, unrecognized, and undefined in the processes network.

Therefore, important relations are missing in the IPO diagrams, and the current knowledge representation lacks the capability to describe the system life cycle processes as a semantic network. Specifically, the relations between sub-processes are missing, as shown in Figure 6.23.

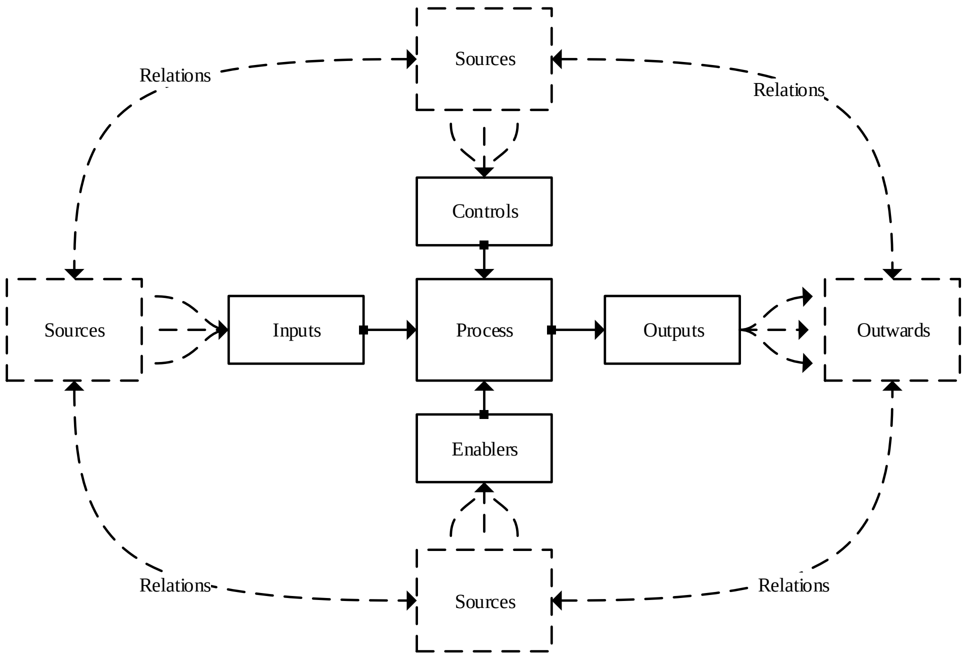


Figure 6.23 Extension of IPO diagram with potential relations

In the middle, one IPO diagram stands alone. The boundary of this IPO diagram is limited to what is presented as its objects. In fact, outside the boundary, all inputs, controls, and enablers must have some sources that import them into the process of interest. Moreover, every output must have some destination, no matter if they are exported to another process or a few processes or even outside the whole life cycle processes.

Based on the above analysis, improvement and optimization can be made to change the stand-alone IPO diagrams into a network. It will expose what relations are implicitly contained in the process. According to the precious achievement in this research, the SE ontology can realize this and even make the processes computer-readable.

6.4.3 Development of System Life Cycle Processes Ontology

The SE ontology presented in Section 6.3 is derived from the INCOSE SE handbook, capturing the entire SE body of knowledge. System life cycle processes are one of the core elements of the SE domain knowledge. As the extant SE standards fail to represent the processes in an explicit and interoperable way, it is promising to take advantage of the SE ontology to reorganize and represent the system life cycle processes in ontological knowledge representation. This endeavor can make the isolated life cycle processes become an interconnected network. In order to achieve such a network, a 9-step method is adopted, illustrated in Figure 6.24. Each step is briefly introduced as follows.

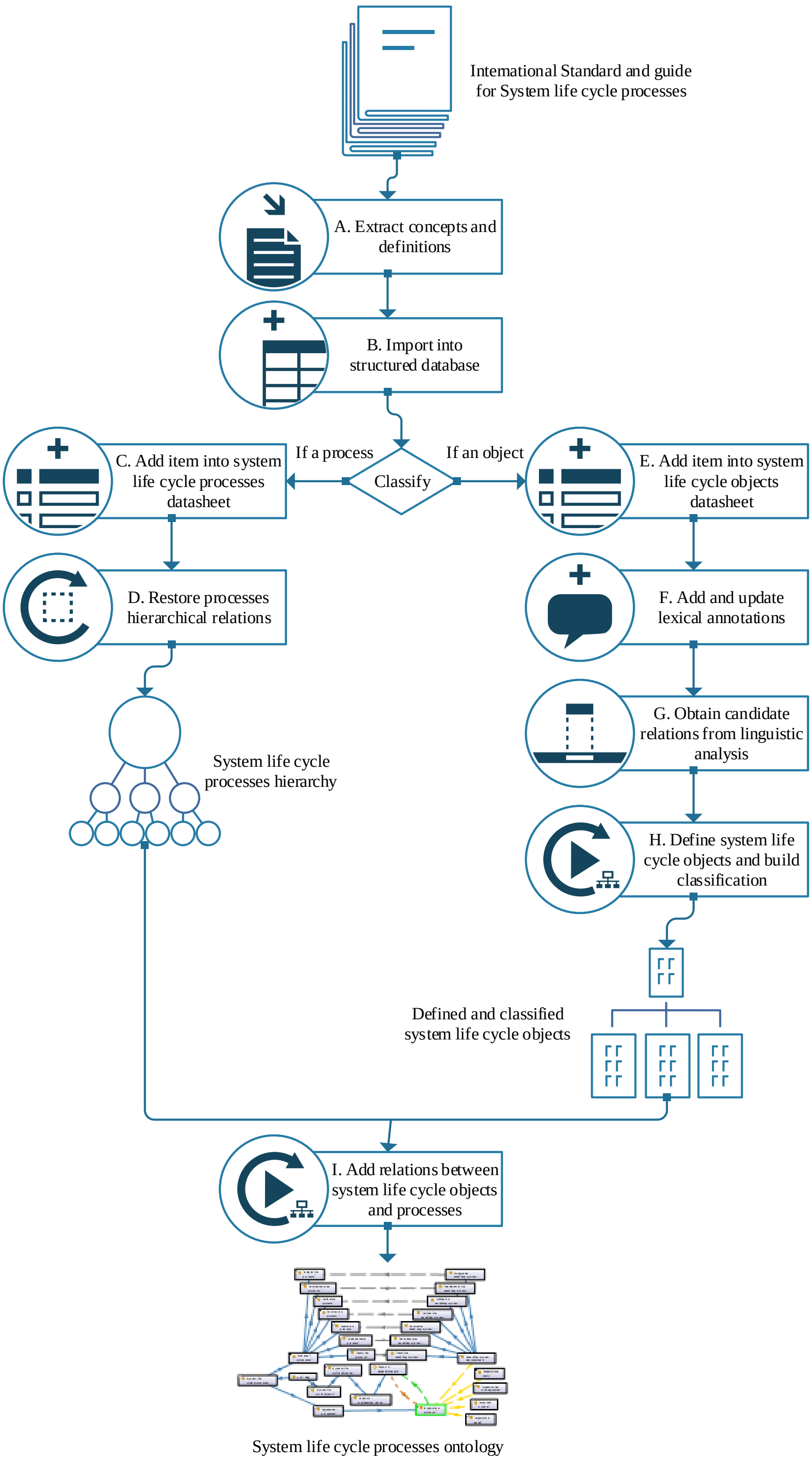


Figure 6.24 Flowchart of generating system life cycle processes ontology

A. Extract Concepts and Definitions

ISO/IEC/IEEE 15288:2015 adopts a common structure to describe each system life cycle process, its purpose, outcomes, as well as activities and tasks. The Input-Process-Output (IPO) diagram in the INCOSE guide contains the inputs, outputs, controls, and enablers of each system life cycle process. Both of the standard and the handbook are available in electronic format, especially with the handbook already formulating an SE corpus. In the first step, the activity is to extract all the life cycle processes, objects, and their textual definitions from the international standard and the INCOSE SE handbook. This has been achieved in the previous research activities when learning the SE ontology from the handbook. As for life cycle processes, their purposes were extracted from ISO/IEC/IEEE 15288:2015 because it is important to specify the objectives when executing a process. As for the life cycle objects, the INCOSE SE handbook has summarized a list of all the objects with their respective descriptions arranged alphabetically. However, it does not distinguish which role or roles (i.e., as inputs, outputs, controls, enablers) a life cycle object plays.

B. Import into Structured Database

The extracted data were firstly stored in datasheets. The terms and their definitions were also extracted in the learning process of the SE ontology. The datasheet structures to store life cycle processes and objects are shown in Figure 6.25 (a) and (b).

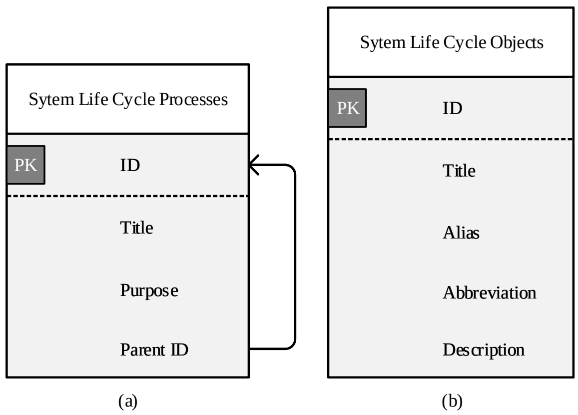


Figure 6.25 Data model of the datasheets

In Figure 6.25 (a), 'ID' is a set of automatically generated numbers without repetition, as the primary key (shown as PK in the figure) to identify each life cycle process. 'Title' is the name of a process consistent with the international standard. 'Purpose' is a short textual description to explain the goal of a process. 'Parent ID' is used to mark up the hierarchical relations between processes. In Figure 6.25 (b), the properties for life cycle objects are more than processes. Apart from 'ID' and 'Title', 'Alias' was used to save other names of a system life cycle object, 'Abbreviation' to store acronym, and 'Description' to note the meaning.

C. Add Item into System Life Cycle Processes Datasheet

This step is to populate the data into the datasheet. With the developed SE ontology, it is effortless to exchange desired information between OWL (format of an ontology) and SQL (format of a database).

D. Restore Processes Hierarchical Relations

After obtaining the structured data, the SE ontology on WebProtégé was exported to desktop Protégé as reasoners are needed to run to infer class hierarchy. Figure 6.26 shows the ontological expression for a system life cycle process, taking the ‘acquisition process’ as an example. Note that the model includes owl:Class to mark the concept of interest by using an unique ID (in this example, ‘#OWLClass_265’ refers to ‘acquisition process’); rdfs:subClassOf to build the processes hierarchy (in this example, ‘#OWLClass_264’ refers to ‘agreement processes’ which means ‘acquisition process’ is a subclass of ‘agreement process’); rdfs:label to indicate the name of the process; and rdf:purpose to provide the objectives included in the original standard.

```
<owl:Class rdf:about="#OWLClass_265">
    <rdfs:subClassOf rdf:resource="#OWLClass_264"/>
    <rdfs:subClassOf>…</rdfs:subClassOf>
    <rdfs:label xml:lang="en">acquisition process</rdfs:label>
    <rdfs:purpose xml:lang="en">
        The purpose of the acquisition process is to obtain a product or service
        in accordance with the acquirer’s requirements.
    </rdfs:purpose>
</owl:Class>
```

Figure 6.26 Example of the ontological model for system life cycle processes

E. Add Item into System Life Cycle Objects Datasheet

This activity was done in parallel with step C. Computer programs were written to facilitate the operation to add system life cycle objects into the datasheet according to the data structure. The datasheet contains 222 records that are the entire set of system life cycle objects. In the meantime, the ontological model was created for life cycle objects. A class named ‘system life cycle object index’ was used to include every object. There is no manually added hierarchy in the index because the classification of objects can be generated automatically in the later steps.

F. Add and Update Lexical Annotations

There is a pretreatment to the names of the life cycle objects. When generating the SE ontology, synonyms have been considered to store full names, alias and abbreviation as the INCOSE SE handbook use them in a mixed way. In order to unify the usage, full names were employed as the label of each object. Aliases and abbreviations were updated in the lexical annotations. The reason to keep these aliases and abbreviations is that many of them are commonly used than the full names. Table 6.2 shows some examples.

Table 6.2 Unifying terminology

Original term	**Unified Term**	**Notes**
Candidate configuration items (Cis)	Candidate configuration items	Updates 'Cis' to synonym as an alias
Initial RVTM	Initial requirements verification and traceability matrix	Update 'Initial RVTM' to synonym as an alias
Operator /maintainer training materials	Operator training materials	Update 'maintainer training material' to synonym as an alias
SEMP	Systems engineering management plan	Update 'SEMP' to acronym as an abbreviation

Figure 6.27 shows the ontological expression for the life cycle objects, taking 'candidate configuration items' as an instance. Note that the model further includes rdfs:definitionEn to provide the English definition of the object and rdfs:synonym to indicate the alias.

```
<owl:Class rdf:about="#OWLClass_60">
    <rdfs:subClassOf rdf:resource="#OWLClass_37"/>
    <rdfs:subClassOf>
        <owl:Restriction>...</owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:definitionEn xml:lang="en">
        Items for configuration control. Can originate from any life cycle
        process
    </rdfs:definitionEn>
    <rdfs:label xml:lang="en">candidate configuration items</rdfs:label>
    <rdfs:synonym xml:lang="en">CIs</rdfs:synonym>
</owl:Class>
```

Figure 6.27 Example of ontological expression for system life cycle objects

G. Obtain Candidate Relations from Linguistic Analysis

In order to obtain the candidate relations within the system life cycle processes, it is necessary to undertake a linguistic analysis of the original standard. This can be done in a semiautomatic way by applying NLP techniques. As this has been described in previous chapters in detail, these activities will only be briefly summarized here. An important aspect of this process is natural language comprehension. For this reason, several different kinds of programs were employed, including lemmatization (which implements stemming algorithms to extract the lexeme or root of a word), morphological analysis (which gleans sentence information from their constituent elements: morphemes, words, and parts of speech), syntactic analysis (which group sentence constituents to extract elements larger than words), and semantic modeling (which represent language semantics in terms of concepts and their relations, using abstraction, logical reasoning, organization, and data structuring capabilities).

From a linguistic analysis, it is possible to determine a hierarchical relationship when the name of a term contains the name of another one (for example, the term agreements and the terms acquisition agreement and supply agreement), or when expressions such as 'is a' or 'including' linked to the name of another term included in the standard appear in the text of the term definition.

Apart from hierarchical relations, there are many more complex relations between life cycle processes and life cycle objects, or just among objects. These relations are usually expressed in verb forms, such as the base form of a verb, the past tense, the gerund, the past participle, etc. Therefore, NLP was applied for processing the raw text in the standard by sentence segmentation, word tokenization, and part-of-speech tagger. For example, the description of qualified personnel is that the right people with the right skills are assigned at the right time to projects per their skill needs and timing, which can be illustrated in the constituency tree, as shown in Figure 6.28.

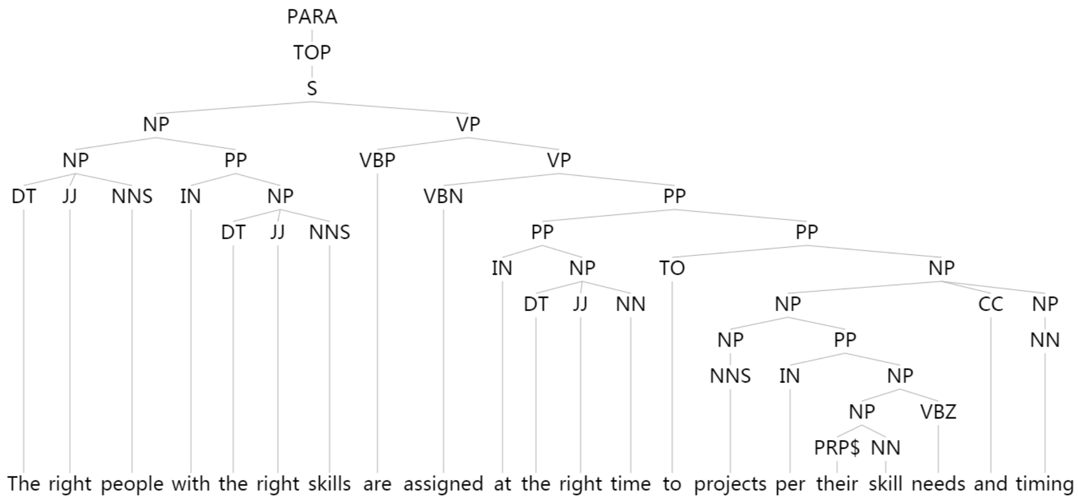


Figure 6.28 Example of constituency tree

Moreover, there are four special relations between life cycle processes and life cycle objects recognized and then added to the ontology model, which are 'is input of', 'is output of', 'is control of', and 'is enabler of'. These four relations are hidden in the original text and cannot be analyzed by NLP but are widely used in the standard, especially in the IPO diagrams. Therefore, they are nominated as candidates' relations as well.

H. Define System Life Cycle Objects and Build Classification

After obtaining the candidate relations, they were then used to define life cycle objects. A well-defined class is capable of being used for automatically classification by logical reasoning. The class hierarchy defined in Section 6.3.4 is also used for generating concept classifications of life cycle processes.

I. Add Relations between System Life Cycle Objects and Processes

The final step and the previous one overlap and are iterative as they both involve the creation of relations in the life cycle processes. Therefore, in this step, specific groupings are defined to provide different perspectives, viewpoints, and purposes to analyze the life cycle processes and objects. The generation of groupings is important to build the classification. Groupings such as systems life cycle objects as inputs (or outputs, controls, enablers) enrich and complete the classifications by adding new insights to the system life cycle processes.

Through the process described above, an ontology for system life cycle processes is created.

The ontology presents the system life cycle processes in both human-readable and computer-readable way. It has been proved that the computer can ‘understand’ what a life cycle process and what a life cycle object is by executing logical reasoning or queries to return the answers correctly. The ontology can also restore the functionality of IPO diagrams, as shown in Figure 6.29, which takes the ‘acquisition process’ as an example.

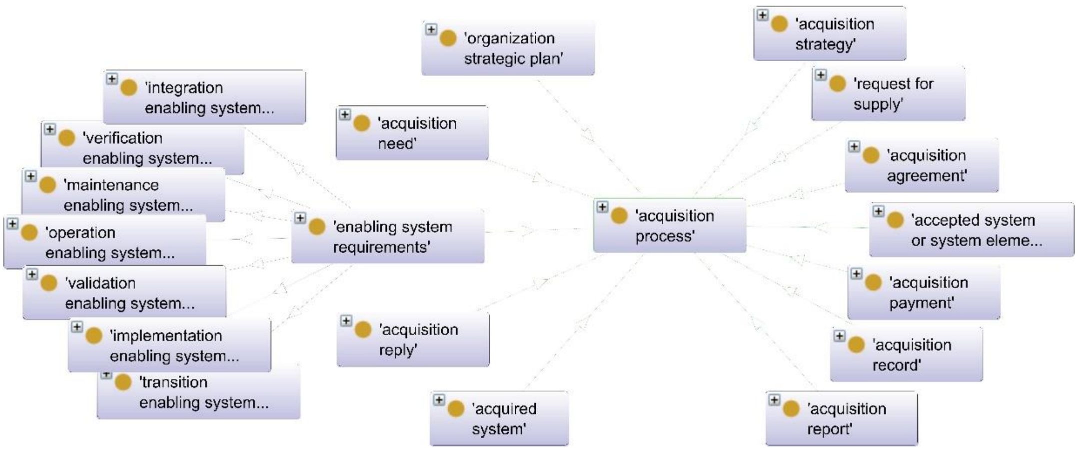


Figure 6.29 Restoration of the functionality of IPO diagrams

In Figure 6.29, inputs are on the left, among which the input ‘enabling system requirements’ further includes seven kinds of detailed enabling system requirements. The outputs are displayed on the right.

6.4.4 Reasoning and Inferring

There are 222 system life cycle classes, and by applying NLP techniques, 162 properties or relations between these classes are recognized and defined. The top 20 properties with their frequency appearing in the definitions of system life cycle objects are shown in Figure 6.30.

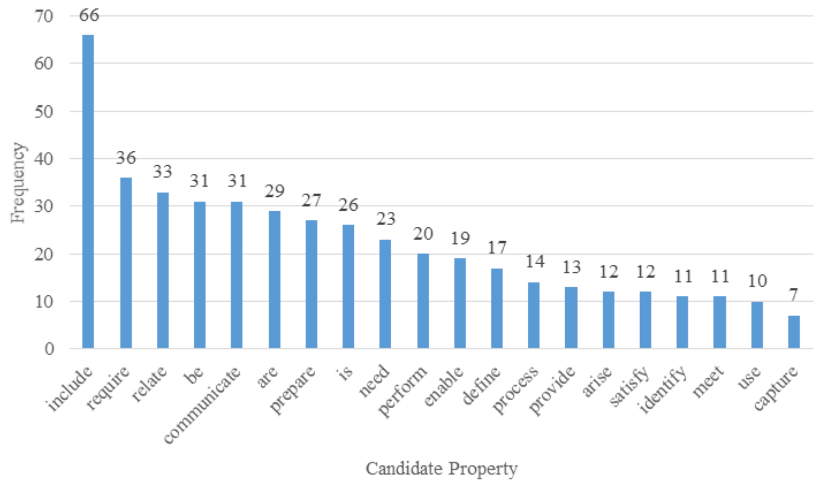


Figure 6.30 Candidate properties (top 20)

The most frequently used property is 'include', which announces the 'is a' relation. Note that the verb form is transformed through lemmatization processing, therefore 'include' represents the different forms that appear in the raw text such as 'including', 'includes', 'included', etc.

A classification of concepts in the life cycle processes is also presented in the ontology. Different groupings for different perspectives, viewpoints, and purposes to analyze and apply the processes are defined. Figure 6.31 shows all possible intersections of inputs (I), outputs (O), controls (C), and enables (E) by different colors, which are also the possible groupings that the ontology can create.

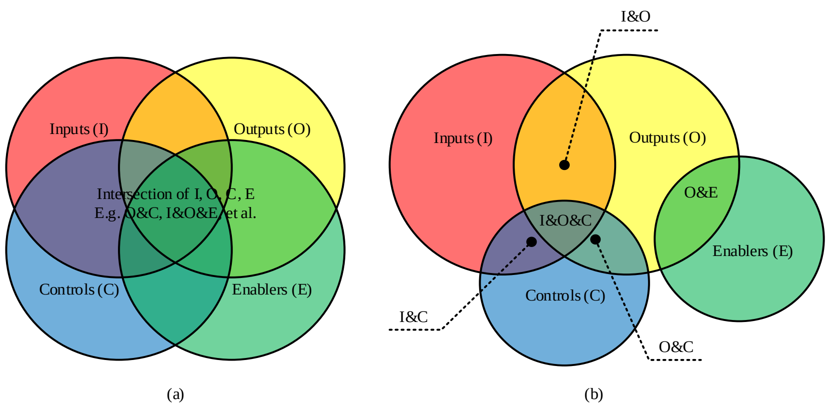


Figure 6.31 Possible groupings in theory vs. meaningful groupings in reality

The analysis starts with examining the intersection of every pair of two different kinds of objects, i.e., I&O, I&C I&E, O&C, O&E, and C&E. When getting an empty set from the intersection of two kinds of objects, there is no point adding a third one since it will also be an empty set. After the first round of reasoning, the ontology returns the following results. There are two empty set, I&E, and C&E, which means there are no life cycle objects that both play the role of input and enabler, and control and enabler. Therefore, there is no point examining the intersections that are smaller than I&E and C&E. The second round is to examine if there is a nonempty set in I&O&C and O&C&E. The reasoning shows that O&C&E turns out to be an empty set, but there is a common element among inputs, outputs, and controls, which is ‘project direction’. A sketch diagram to demonstrate the reasoning result of intersections of life cycle objects is shown in Figure 6.31 (b).

6.4.5 Reorganization of the Life Cycle Processes in IDEF0

The ontology for system life cycle processes has a direct impact on the reorganization of the processes. In order to keep similar formatting to the IPO diagrams, the IDEF0 model is selected to present the reorganization of the life cycle processes.

Note that the IDEF0 diagrams created for the life cycle processes have only referred to the latest English version (4.0, published in 2015) of the INCOSE SE handbook. Other versions, such as the 2011 version, are not considered. It is worth noting this since they have a few inconsistencies. Also, the IDEF0 diagrams only contain the system life cycle processes, not including the tailoring process. Therefore, the top-level diagram (Diagram A-0) is named system life cycle processes. It includes the four groups of processes defined in Diagram A0, which are technical processes, technical management processes, agreement processes, and organizational project-enabling processes.

In order to simplify the names, Table 6.3 provides a specification of the abbreviations of the four groups of processes and their sub-processes. The colors in the IDEF0 diagrams are consistent with the table. At the same time, this thesis will also use these abbreviations to simplify the expression. Because some abbreviations have duplicate names, special letters have been added to duplicate names for deduplication.

Table 6.3 Specifications of the abbreviation of the names of the system life cycle processes

Title	**Abbr.**	**Title**	**Abbr.**	**Title**	**Abbr.**	**Title**	**Abbr.**
Technical Processes	TPs	Technical Management Processes	TMPs	Agreement Processes	APs	Organization Project Enabling Processes	OPPs
Business or Mission Analysis Process	BMAP	Project Planning Process	PPP	Acquisition Process	AP	Life Cycle Model Management Process	LCMMP
Stakeholder Needs and Requirements Definition Process	SNRDP	Project Assessment and Control Process	PACP	Supply Process	SP	Infrastructure Management Process	IMP
System Requirements Definition Process	SRDP	Decision Management Process	DMP			Portfolio Management Process	PMP
Architecture Definition Process	ADP	Risk Management Process	RMP			Human Resource Management Process	HRMP
Design Definition Process	DDP	Configuration Management Process	RMP			Quality Management Process	QMP
System Analysis Process	SAP	Information Management Process	InfoMP			Knowledge Management Process	KMP
Implementation Process	IP	Measurement Process	MeaP				
Integration Process	InteP	Quality Assurance Process	QAP				
Verification Process	VP						
Transition Process	TP						
Validation Process	VaP						
Operation Process	OP						
Maintenance Process	MP						
Disposal Process	DP						

Since the four groups of processes generally include a large number of sub-processes, content-related sub-processes are integrated upward during the reorganization process, and mergers are established. The characteristics of these mergers are:

- The name is the abbreviated arrangement of the merged sub-processes.
- The shape uses rounded rectangles, and the color is gray.

In order to make the numerous arrows in the diagrams have a more obvious type distinction, the arrows are colored intentionally. The use and definition of the specific colors are shown in Table 6.4. The identification of these special classes of life cycle objects is based on the SE ontology.

Table 6.4 Specifications of the arrow colors

Arrows	Colors	Explanations
unsorted title		All arrows that have not been purposely colored remain black, which means that these arrows have not been particularly classified.
life cycle record		The concept named records is colored according to its definition (that is, which sub-classes it includes).
life cycle strategy		The concept named strategy documents is colored according to its definition (i.e., which sub-classes it includes).
life cycle report		The concept named reports is colored according to its definition (that is, which sub-items it includes). Note: In the SE ontology, it especially recognized that some reports are not included in this class (the original text in the handbook is that other reports go to other process areas and are not aggregated here), and these reports are not colored.
procedure		The concept named procedures is colored according to its definition (that is, which sub-items it includes).
enabling system requirement		The concept named enabling system requirements is colored according to its definition (i.e., which sub-items it includes).
life cycle concept		The concept named the life cycle concept is colored according to its definition. Note: The definition of the life cycle concept indicates which sub-classes it includes. However, the sub-classes do not appear in the IPO diagrams. In order to specify this concept, the color is reserved.
measurement data		The concept named measurement data is colored according to its definition (that is, what it includes).
measurement needs		The concept named measurement needs is colored according to its definition (that is, what it includes).
system (flow)		The arrows in this color mark the flow that a system or system element evolves in life cycle processes. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
life cycle constraint		The concept named life cycle constraints is colored according to its definition (that is, what it includes).
traceability		All concepts that are traceability are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
traceability matrix		All concepts that are traceability matrix are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
description		All concepts that are as descriptions are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.

Arrows	Colors	Explanations
rationale		All concepts that are rationale are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
definition		All concepts that are as a definition are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
identification		All concepts that are as identification are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
requirement		All concepts that are as a requirement are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
criteria		All concepts that are criteria are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
preliminary measurement data		All concepts that are preliminary measurement data are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
preliminary measurement needs		All concepts that are preliminary measurement needs are marked. Note: This concept is not defined in the handbook but is inferred from the SE ontology.
preliminary life cycle concept		The concept named the preliminary life cycle concept is colored according to its definition. Note: The definition of a preliminary life cycle concept indicates which sub-classes it includes. However, the sub-classes do not appear in the IPO diagrams. In order to specify this concept, the color is reserved.

To present the reorganized life cycle processes, a set of IDEF0 diagrams were developed. These diagrams have a feature that they can be decomposed from the top-level processes. Figure 6.32 presents the top-level diagram of the system life cycle processes.

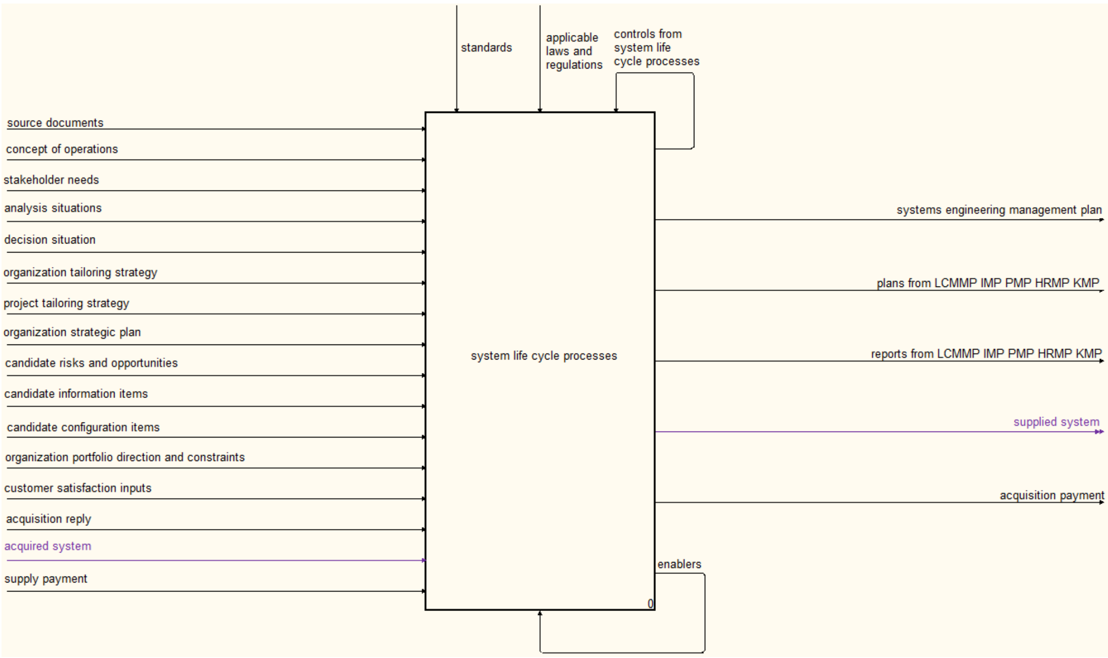


Figure 6.32 Top-level (A-0) diagram of the reorganized life cycle processes

As can be seen from the left, inputs that are not internally created from the processes are presented one by one. On the right, outputs can be classified into seven. They are controls from system life cycle processes, a unique plan called 'systems engineering management plan', plans generated from specific life cycle processes, reports produced from specific life cycle processes, supplied system, acquisition payment, and enablers. Among the inputs and outputs, it can be seen that two colored arrows named 'acquired system' and 'supplied system' are highlighted according to the specification in Table 6.4. Moreover, from this top-level diagram, it can be seen that all the enablers are generated from life cycle processes internally and work upon the entire processes. As for controls, standards, and applicable laws and regulations come from the external environment. The rest of the controls all come from the processes internally.

Figure 6.33 shows a decomposition of the top-level (A-0) diagram further broken down into the four groups of life cycle processes, A0 diagram.

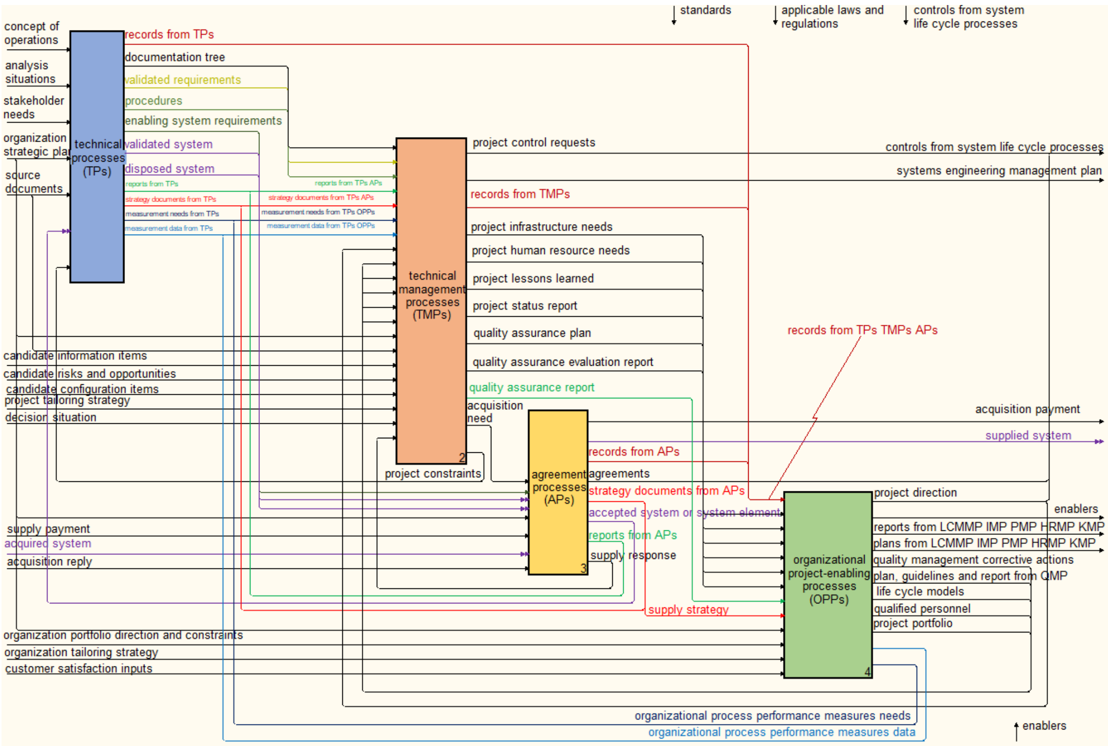


Figure 6.33 A0 diagram of the four groups of life cycle processes

The original life cycle processes defined in the ISO/IEC/IEEE 15288 do not provide the relationships between the technical processes, technical management processes, agreement processes, and organizational project-enabling processes. Through the establishment of the SE ontology, these relations are identified and can be explicitly presented. Figure 6.33 depicts the links through the arrows between the boxes. Colored arrows are special groupings in the SE ontology, obtaining from reasoning and inferring. The rules are presented in Table 6.4.

Each of the four groups can be further decomposed. Due to the length limit, they are attached in Appendix 1.

6.5 Conclusion

Chapter 6 presents a comprehensive and complete case study that applies the ontology learning approach and the ontology models in practice. Moreover, it also details the developed SE ontology that represents the entire SE body of knowledge with the terminologies that the ontology contains, the top-level concepts that the ontology is structured, and the concepts and relations that are derived from the INCOSE SE handbook. Additionally, the developed ontology is used to restructure the system life cycle processes, as illustrated by the new process models presented by Figure 6.29, Figure 6.32, and Figure 6.33, as well as the specifications in Table 6.3 and Table 6.4.

Through the case study, the following conclusions can be drawn.

- The ontology learning approach shows excellent advantages in acquiring SE knowledge primitives, yielding more than 5,000 different kinds of terms with their degree of importance in the SE body of knowledge.
- The ontology conceptual, logical, and data models are powerful for organizing the SE domain knowledge into a well-defined multiple parenting hierarchy.
- The SE ontology depicts the SE body of knowledge from nine dimensions, i.e.,
    - abstract concepts or properties of a system,
    - activities, procedures or processes,
    - approaches, methodologies or theories,
    - capabilities or characteristics,
    - data or information materials,
    - engineering disciplines,
    - physical concepts or properties of a system,
    - plans or strategies, and
    - scales.
- The developed ontology shows a high degree of interoperability and formality.
- The SE ontology has great potential in explicitly specifying the SE body of knowledge as well as restructuring and inferring new knowledge for the SE discipline.

CHAPTER 7 SUMMARY OF THE WORK, CONTRIBUTIONS, LIMITATIONS, AND RECOMMENDATIONS FOR FUTURE WORK

7.1 Introduction

This chapter provides a summary of the research undertaken. First, it presents a summary, including a review of the rationale for the study, a recap of the research gaps and questions, and a summary of the research achievements. Then, this chapter highlights the research contributions from the theoretical, methodological, and application perspectives, as well as the impacts on academia and industry. Next, the limitations of the research are discussed. Finally, it gives future research recommendations, including a possible roadmap for the future OBSE research implementation.

7.2 Thesis Summary

This section provides a summary of the work undertaken. The research is dedicated to developing a formal ontology for the SE body of knowledge via the emerging ontology learning approaches. To this end, several endeavors were made in the following aspects.

7.2.1 Review of the Rationale for the Study

So far, there is no single standard to unify the SE domain knowledge (Ward *et al.*, 2018b). In fact, domain knowledge is still treated primarily as heuristics, learned by each practitioner (Friedman and Prusak, 2008). The differences between various standards lead to disagreement and confusion, making information exchange increasingly difficult (Gonzalez-Perez *et al.*, 2016). Miscommunication and lack of mutual understanding of key concepts in the SE domain can potentially lead to dire consequences (Dori and Sillitto, 2017).

Moreover, current SE standards still remain document-centric, making its model-based transition very difficult (Ernadote, 2017). The prerequisite of model-based SE (MBSE) is to achieve a logically rigorous specification of the entire SE processes, i.e., what artifacts are generated, via which activities, by means of who, and how do they evolve? However, current SE standards fail to provide these relations explicitly, since their textual descriptions often have terminological ambiguity and relational inconsistencies (Yang *et al.*, 2017). Therefore, SE, especially for MBSE, is in need of a formal, explicit, and shared instrument to specify its body of knowledge. Nowadays, such knowledge representation is an ontology.

Ontologies are increasingly used by systems engineers to solve SE problems (Yang *et al.*, 2019a). They are employed to create a shared conceptualization (Forsberg *et al.*, 2010; Ring and Troncale, 2014), enable interoperability and communication among multiple disciplines or across different stakeholders (Fraga *et al.*, 2015; Gutierrez, 2018), and define key SE concepts explicitly to avoid incompleteness and ambiguity (Dong and Hussain, 2014; Guo *et al.*, 2012).

However, extant SE ontologies are either developed for partial SE subdomains or have low degrees of formality (Yang *et al.*, 2019a).

In order to improve the current research status and fill the gaps, this research was proposed and carried out. The specific research questions are reviewed in the next section.

7.2.2 Review of the Research Questions

This research begins by identifying the gaps and problems currently existing in the SE domain. Among all the problems, four glaring gaps were identified, which demand prompt solutions. They are synthesized as follows.

- The understanding of the nature of SE is implicit and different, as shown by the fragmented and fractured development of the SE standards (Di Maio, 2011; Honour, 2004; Hutchison *et al.*, 2017).
- Current SE standards and meta-models lack interoperability and computer interpretability to support the MBSE transition (Eito-Brun, 2016; Engel *et al.*, 2018; Ernadote, 2017; Giachetti, 2015; Givehchi *et al.*, 2017; Madni and Sievers, 2018a; Rosa *et al.*, 2019).
- There is an absence of a widely accepted and consistent terminology for fundamental SE concepts, which results in miscommunication, misunderstanding, and misinterpretation among SE stakeholders (Dori and Sillitto, 2017; Ernadote, 2015; Haskins and Ruud, 2018; Lin and Harding, 2007; Rousseau *et al.*, 2016; Sarder *et al.*, 2007).
- There still lacks a cohesive knowledge representation to conceptualize the entire SE body of knowledge (Adcock *et al.*, 2016; Di Maio, 2010; Hallberg *et al.*, 2014; Martin *et al.*, 2013; van Ruijven, 2013; Ward *et al.*, 2018a).

In the SE community, more and more researchers call for an SE ontology to fix the problems. They call for formal ontologies, advocate ontology-based SE (OBSE), and argue that the SE standards and SE body of knowledge will benefit from such cohesive SE ontology (Aslaksen *et al.*, 2011; Engel *et al.*, 2018; Ernadote, 2015; Giachetti, 2015; Givehchi *et al.*, 2017; Hallberg *et al.*, 2014; Martin *et al.*, 2013; Mezhuyev, 2014).

As ontologies and SE are two different domains, it is necessary to investigate what ontologies mean to SE and what benefits that ontologies can bring to SE. Consequently, a review on the extant ontology literature was carried out, focusing on the current definitions, formalisms, and classifications of ontologies. Also, the state of the art of ontology engineering techniques was understood, primarily focusing on the emerging ontology learning approach.

The review of SE literature helps to identify the existing gaps. The review of ontology literature contributes to finding possible solutions. There is a third angle, which is the literature that shows the extant use and development status of ontologies in SE or simply OBSE.

To understand the state of the art of OBSE, this research has conducted a comprehensive literature review through the systematic literature review (SLR) mechanism. It is the first SLR in the OBSE domain. The aim of the review is not only understanding the state of the art but also identifying gaps and needs existing in the OBSE domain. In the state-of-the-art review, four questions have been thoroughly analyzed.

- What SE knowledge areas are supported by ontologies, and to what extent?
- Why ontologies are used in SE?
- What are the existing SE ontologies?
- What ontology engineering methods, languages, and tools are applied to develop these ontologies?

As there is no previous literature review on this topic, the four review questions are elaborately and specifically designed to understand the most important issues in the state of the art of OBSE.

The first question focuses on exploring the width and depth of the application of ontologies in SE. As SE covers a vast knowledge domain and contains a variety of knowledge areas, a clear definition of the scope of SE in this study must be made first. Therefore, this research has defined a typology for the SE knowledge areas (Figure 2.1), based on a synthesis and understanding of currently widely accepted and authoritative standards. This typology presents a clear classification of SE knowledge areas and demonstrates it by six subjects, i.e., systems fundamentals, representing systems with models, engineered system contexts, SE standards, generic life cycle stages, and SE management. Each subject group further contains smaller subjects that deal with a specific knowledge area. Per each knowledge area, a literature review has been thoroughly conducted. The results are summarized in Figure 3.2 and detailly presented in Section 3.4. The results show how ontologies are applied in various SE knowledge areas.

The second question looks at the literature from another perspective, which is why ontologies are created and used in SE. The aim of the question is to reveal what kind of benefits that ontologies bring to SE. Table 3.4 provides a summary of the contributions of ontologies. However, most studies do not present the purposes for using ontologies explicitly. Therefore, the role of ontologies in solving SE problems is obtained by a careful analysis of the literature.

The third and fourth questions are both built on the fact that ontologies are an artifact. Therefore, each ontology has its own scope, a methodology to build, language to develop, and tools to present. They determine the knowledge domain that the ontology captures and the degree of formality that the ontology possesses. These two questions also reflect the state of the art of ontology engineering techniques being adopted in the SE community.

Through the literature review on the four questions, a few conclusions are drawn.

- There is no ontology that captures the entire SE knowledge domain.
- The methods of developing SE ontology are manual and need to be automated.
- The existing ontologies are still in a high-level description, lacking a detailed specification.
- Ontology development in SE requires the use of formal ontology engineering languages and tools to enable the sharing and reusing.
- Little work has been done on the visualization of the ontologies.

7.2.3 Summary of the Research Achievements

Based on the above points, combined with the latest studies in ontology engineering, this research proposed to use the ontology learning approach and technique to develop a formal ontology for the entire SE body of knowledge. Specifically, the following achievements were made.

- This research proposed a 3-stage ontology learning methodology based on the best practices using various natural language processing (NLP) statistical and linguistic methods.
- Three ontology models were formulated to represent the conceptual, logical, and data facets of the SE ontology.
- A case study was conducted applying the proposed ontology learning methodology to an authoritative SE standard - the INCOSE SE handbook to extract ontological primitives.
- An SE ontology was built based on the ontology models populated by the ontological primitives extracted from the ontology learning process.
- The created ontology was detailed presented in terms of the terminology, top-level concepts, taxonomic relations, and concept hierarchy.
- Through the reasoning and inferring features of formal ontologies, the developed ontology was applied to reorganize the system life cycle processes, which were refined in a set of IDEF0 models.
- A roadmap for future OBSE was generated to provide a clear direction on the research objectives.

7.3 Contributions

This research contributes to the body of knowledge in many ways.

From a theoretical perspective, it provides the first state-of-the-art review on the development and application of ontologies in the SE domain, which has never been conducted before. Within this review, an overall state of the distribution of the extant research is presented systematically, based on a dedicated classification of the SE knowledge areas. Through the review, gaps and inadequacies of the exiting work are clearly exposed. Moreover, a roadmap is generated for OBSE to steer future research directions.

From a methodology perspective, a novel ontology learning approach is proposed for learning SE ontologies from textual SE standards. The ontology learning approach can resolve the knowledge acquisition bottleneck in traditional manual construction. Also, this research is the first that uses the ontology learning approach to generate SE ontologies. It provides new possibilities, new ideas, and new solutions for future OBSE and MBSE.

From the application perspective, this study demonstrates one of the application scenarios of the SE ontology. It utilizes the features of ontologies in reasoning relations and inferring new knowledge to reorganize the life cycle processes. The original static and isolated IPO diagrams are linked into a more dynamic, robust, and interrelated knowledge representation through the interrelations derived from the SE ontology. Moreover, this new knowledge representation can make a big difference in process tailoring and project management of complex systems.

For the academic researcher, this research adds to the OBSE body of knowledge. It establishes a systematic framework to understand what is currently known and what could be explored in the future. It highlights a comprehensive strategy to design, develop, and deploy ontologies and makes sure that gaps in the knowledge domain are identified and can be closed as needed.

For practitioners, the research is designed to help developers to create an ontology, based on the best practice. It also bridges the gap between theory and practice to help set more competitive and realistic goals on the final delivered functions of the ontology. It also serves as a guide for the practitioners during their journey, allowing them to recognize and act on their current models that require an improvement of change.

In terms of deliverables, a complete and formal ontology for SE has been developed. Compared with the existing ones, this ontology has the following three advantages. It is developed based on authoritative SE standards, which ensures the content is consistent with the most widely accepted benchmarks. It is developed in a formal language and presented by sophisticated tools, which contributes to the share and reuse in a variety of scenarios. It covers the entire scope of the SE body of knowledge, which has not been created in any of the previous studies.

7.4 Limitations and recommendations for future work

All studies come with limitations, and this research is no exception. In particular, this study proposes an ontology learning methodology to extract fundamental concepts and relations from SE authoritative standards.

- However, only one comprehensive case study is conducted to present and explain the approach. It is acknowledged that it is good practice to apply the approach to many standards in order to unify terminologies and maintain a consistent vocabulary to suit different cultures.
- Further studies need to be carried out in order to validate that the learned ontology does not omit implicit taxonomic relations.
- Considerably more work will need to be done to automate the process of acquiring non-taxonomic relations and data properties.

In future work, researchers must incorporate more SE standards into the ontology and enable a robust mechanism to update the ontology in order to enrich the axioms to create a more heavyweight SE ontology. This will require further investigations on creating class restrictions, such as equivalent classes to specify the various names for the description of a concept, collaborative ontology development, and public accessibility to validate the ontology.

Based on the analysis of the extant literature, it can be concluded that there are many research gaps within OBSE. Therefore, this section discusses the limitations while proposes a roadmap for potential research directions. This roadmap, as illustrated in Figure 7.1, incorporates vital learnings and insights to help both academic researchers and practitioners implement a comprehensive strategy to create ontologies for SE. For this particular research, this roadmap also plays an important role in guiding the research directions.

The roadmap comprises four critical steps designed to help developers to identify and implement ontologies based on the extant literature. These steps are (1) targeting a SE knowledge area, (2) clarifying functions of ontologies, (3) improving ontology engineering techniques, and (4) becoming part of the solution to SE challenges. Each step serves as a checklist to ensure good practice is upheld. Together, they consist of a comprehensive, integrated strategy for formal ontology modeling and effective implementation.

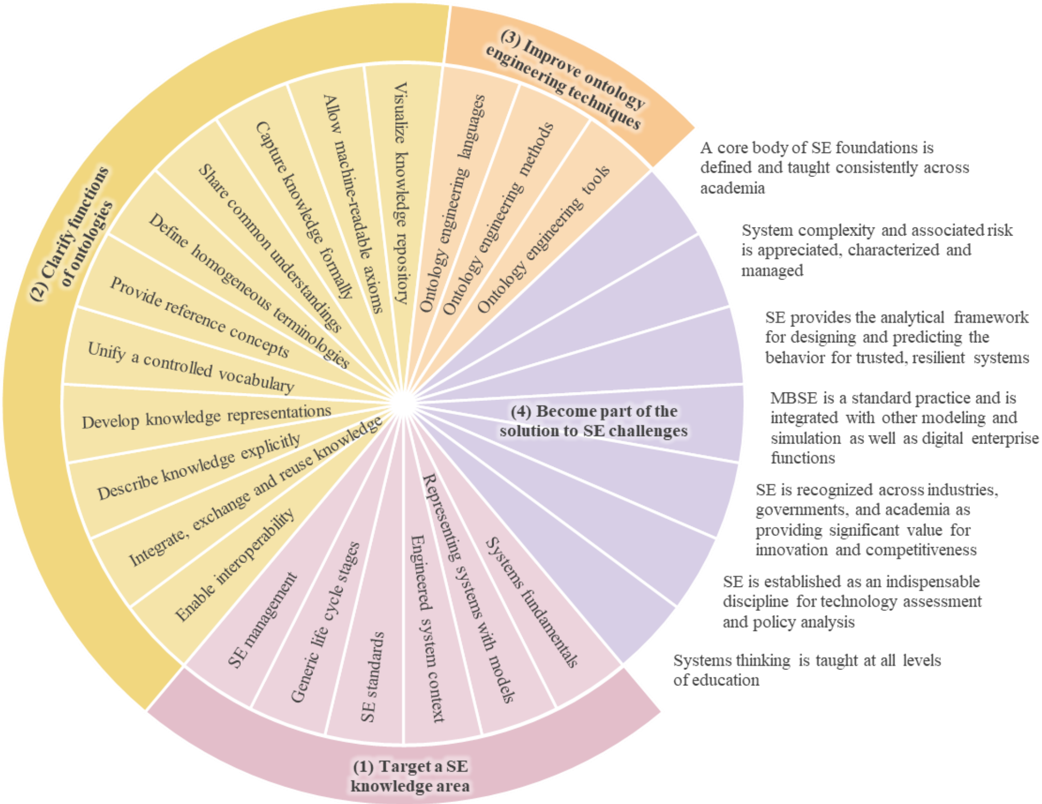


Figure 7.1 A roadmap for OBSE research directions

(1) Target a SE knowledge area

For ontology development, it is critical to choose a specific SE knowledge area, because defining the domain of an ontology is the first step to building an ontology. It is suggested to use the classification created in the study (Figure 2.1) and the bubble chart (Figure 3.2) to look for gaps or areas that have received little attention in the literature. The reason is that the terminologies, concepts, and definitions of these areas are the most ambiguous. From our investigation, much work is done for SE management, especially for risk management. However, there is a need to develop ontologies for systems fundamentals, including the knowledge areas not reviewed in this research. Another area that requires further attention is the later stages in the generic life cycle, as they are directly related to the performance of SE practice. With the support of an ontology during the whole system life cycle, a systems engineer will be able to communicate with various stakeholders more effectively.

(2) Clarify functions of ontologies

Table 3.4 identifies 11 types of contributions that ontologies make to SE from the literature. The result reflects the purposes and rationale of OBSE. However, these purposes should be made clearer so that the functions of the ontology can be highlighted. To address this deficit, the contributions of ontologies should be referred to, as listed in the roadmap. These 11 functions will help a developer to formulate clear goals and objectives for ontology development, which is essential for practical ontology engineering. The ability to clarify the functions is also crucial to strategic planning because they help to turn the objectives into specific, measurable targets and are concrete to help translate the missions into reality.

(3) Improving ontology engineering techniques

There is a dearth of maturity and formality in the existing SE ontologies. This situation can be changed by improving the ontology engineering techniques, including using formal languages, methods, and tools to develop SE ontologies. Therefore, it is recommended that future ontology development should adopt a well-designed methodology rather than heuristics to ensure that best practice is followed. The ontology learning approach can also be improved in many perspectives, such as extracting non-taxonomic relations. Some current ontologies described in natural languages can be formalized by ontology modeling languages such as OWL to check their logic consistency. The adoption of ontology development tools (e.g., Protégé) comes with many advantages, including providing ontology editing functionality, multiple inheritances, visualization, reasoning, and reuse.

(4) Becoming part of the solution to SE challenges

According to the INCOSE Systems Engineering Vision 2025 (INCOSE, 2014), the future state of SE is facing new challenges. These challenges are merged into the roadmap to enable OBSE to become part of the solution.

The roadmap presents a comprehensive strategy for OBSE based on the interrelated steps. This strategy is closely related to the findings of the research. It adheres to the SE knowledge areas followed throughout this article and is directly based on the conclusions drawn from the literature review. In conclusion, the four subsections in the outer ring of the roadmap should be integrated - one area should not be neglected at the expense of the other.

7.5 Conclusion

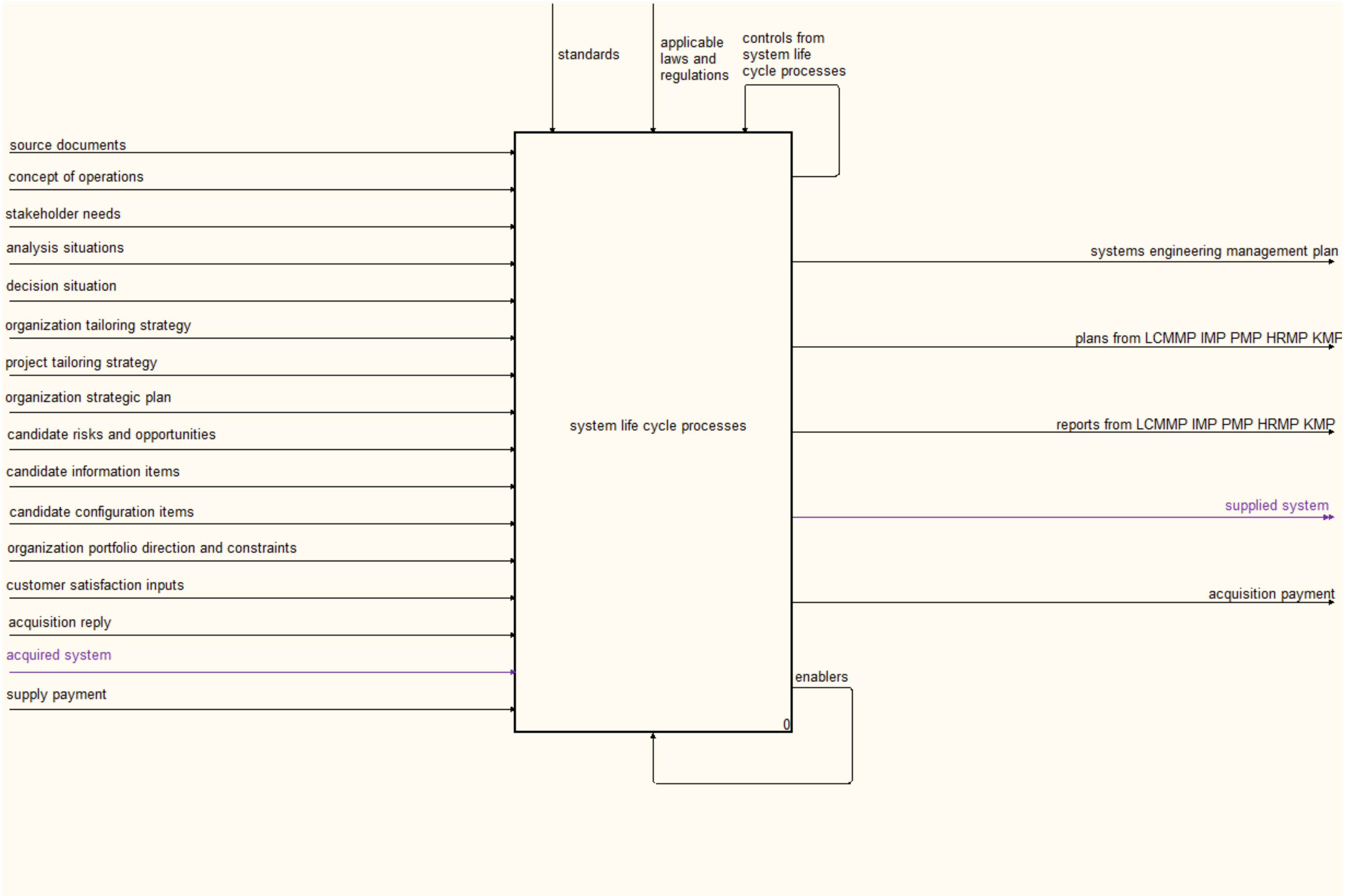
This thesis proposed an ontology learning methodology to develop an SE ontology. It also presents the developed ontology in a formal language by superior tools. The ontology conceptualizes the SE body of knowledge and enables the knowledge beings better shared and reused. It increases harmonization in standards and decreases language conflicts between stakeholders. The ontology learning approach breaks through the knowledge acquisition bottleneck, making ontology development more efficient, accurate, and economical. The classification in the ontology can be used as pre-defined containers to help bring the fragmented standards together to create a cohesive SE body of knowledge. The ontology is machine-readable; thus, it enhances the interoperability in MBSE. It also enables a digital engineering ecosystem and helps to trace application ontologies to an SE reference ontology. Furthermore, the terminologies, syntax, and semantics of fundamental SE concepts are extracted from the authoritative SE standard developed by INCOSE and populated by sophisticated developing languages and tools, which significantly increase the formality of the ontology. This ontology is a new way to evaluate SE terminologies and provide recommendations on the future revisions and diffusions of SE standards.

APPENDICES

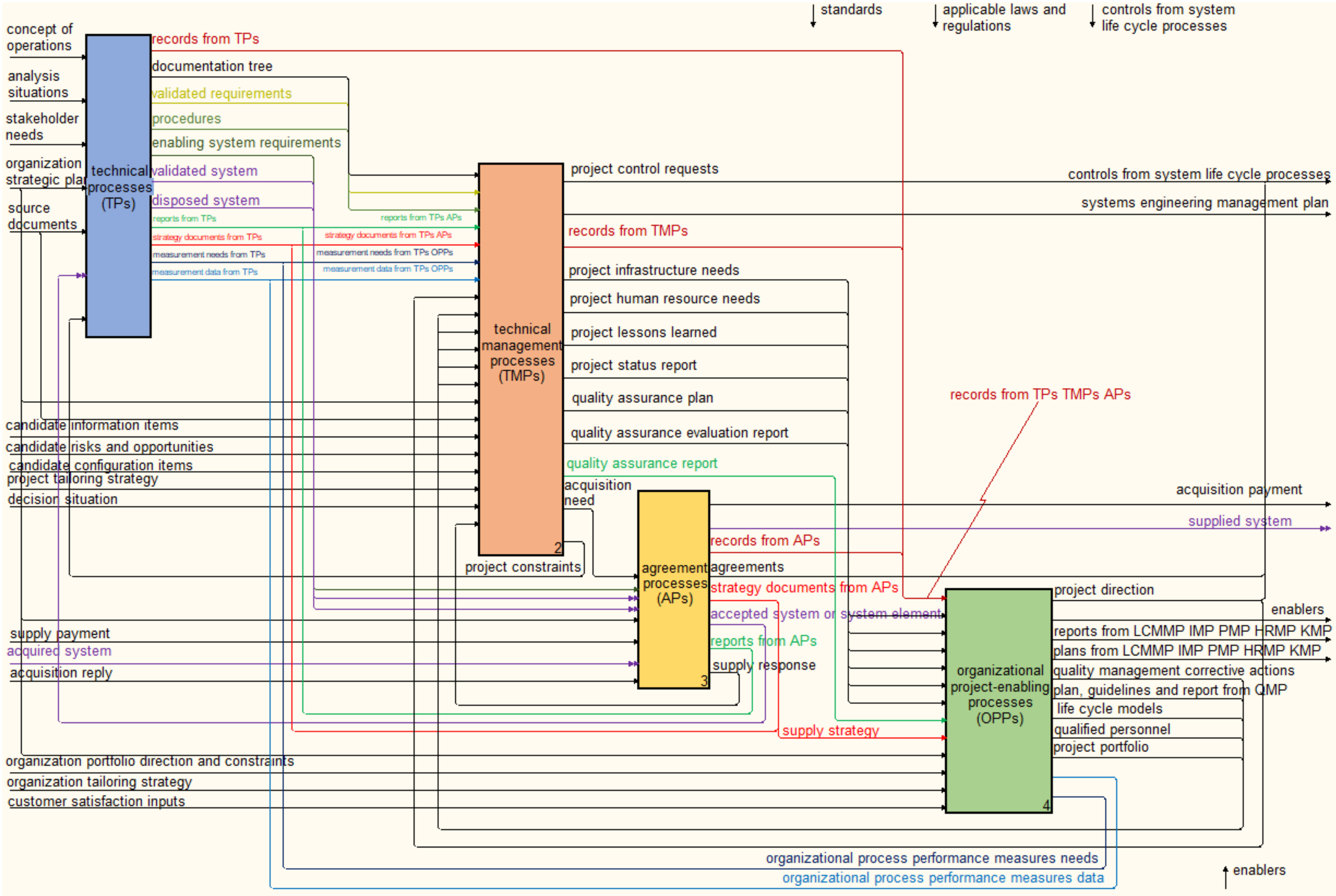
Appendix 1

This appendix includes the set of IDEF0 diagrams that show the interrelationships between the system life cycle processes. The diagrams have a feature of decomposition, meaning that the upper-level processes are broken down into lower-level subprocesses. This decomposition is reflected by the labeling rules of the IDEF0 model. For example, the top-level diagram is labeled by A-0. Its decomposition is numbered as A0. The A0 diagram is made up of A1, A2, A3, etc. Then, A1 can be further broken down into A11, A12, etc.

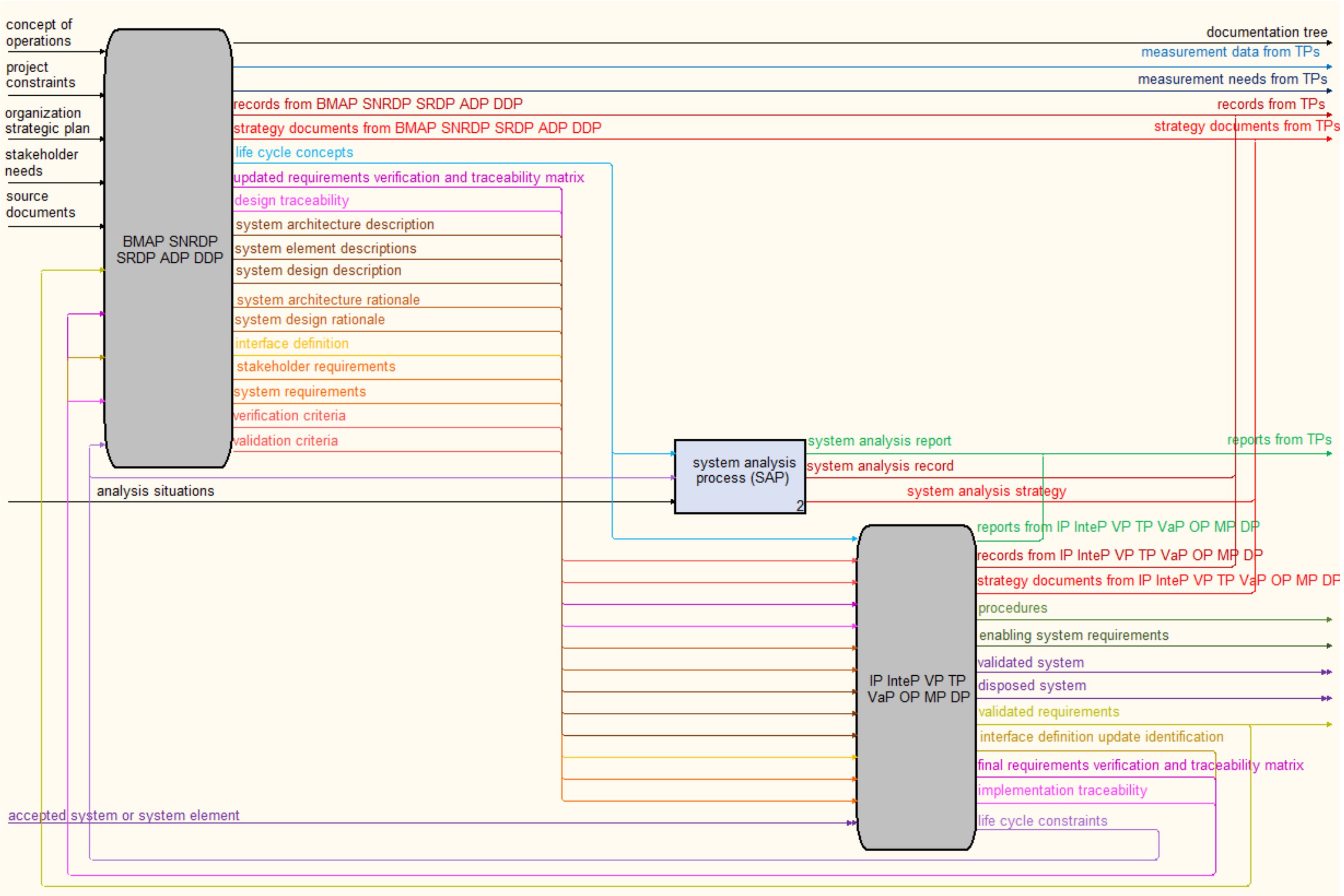
The table of appendix figures is as follows.



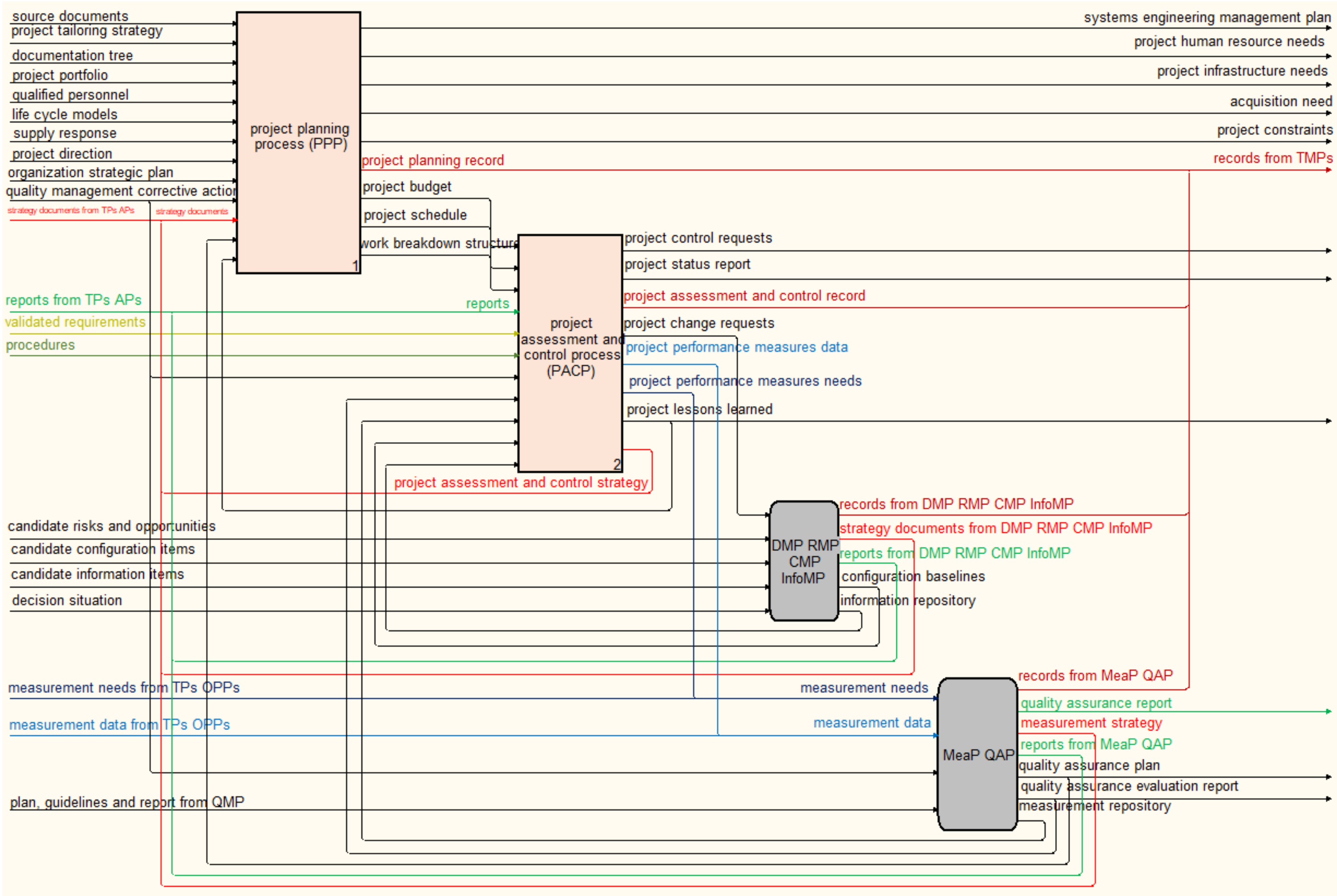
Appendix Figure 1 (A-0) System life cycle processes



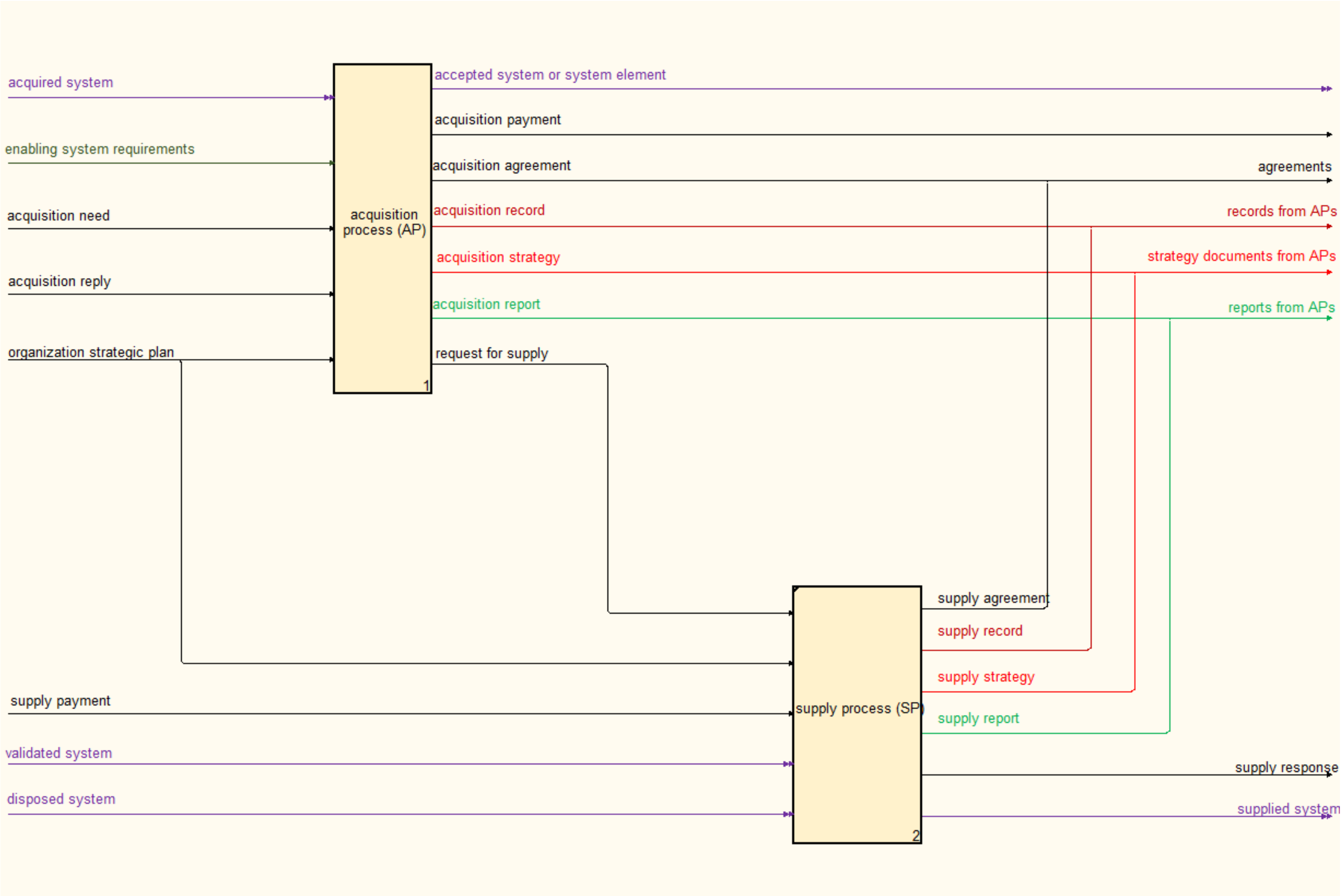
Appendix Figure 2 (A0) System life cycle processes



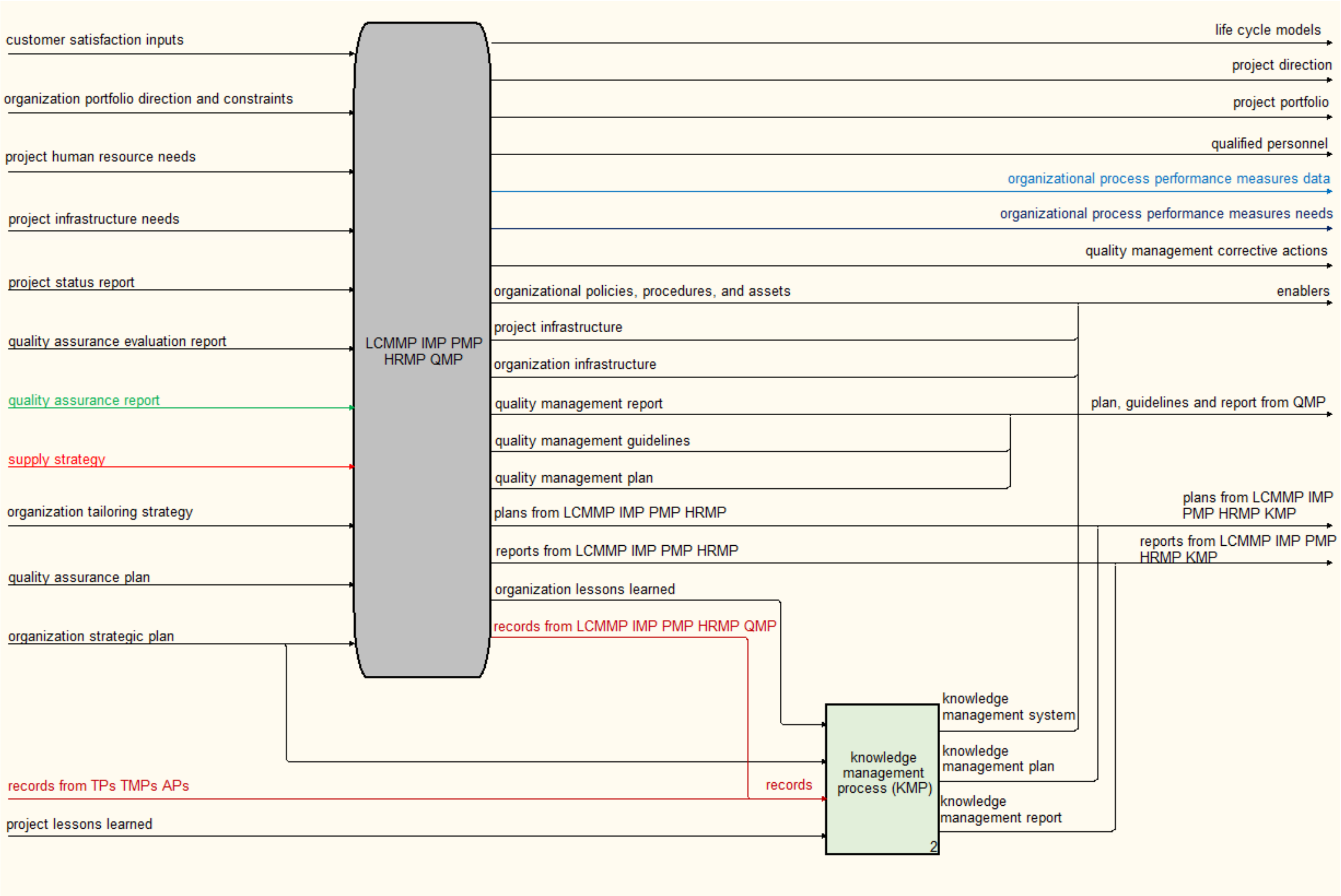
Appendix Figure 3 (A1) Technical Processes (TPs)



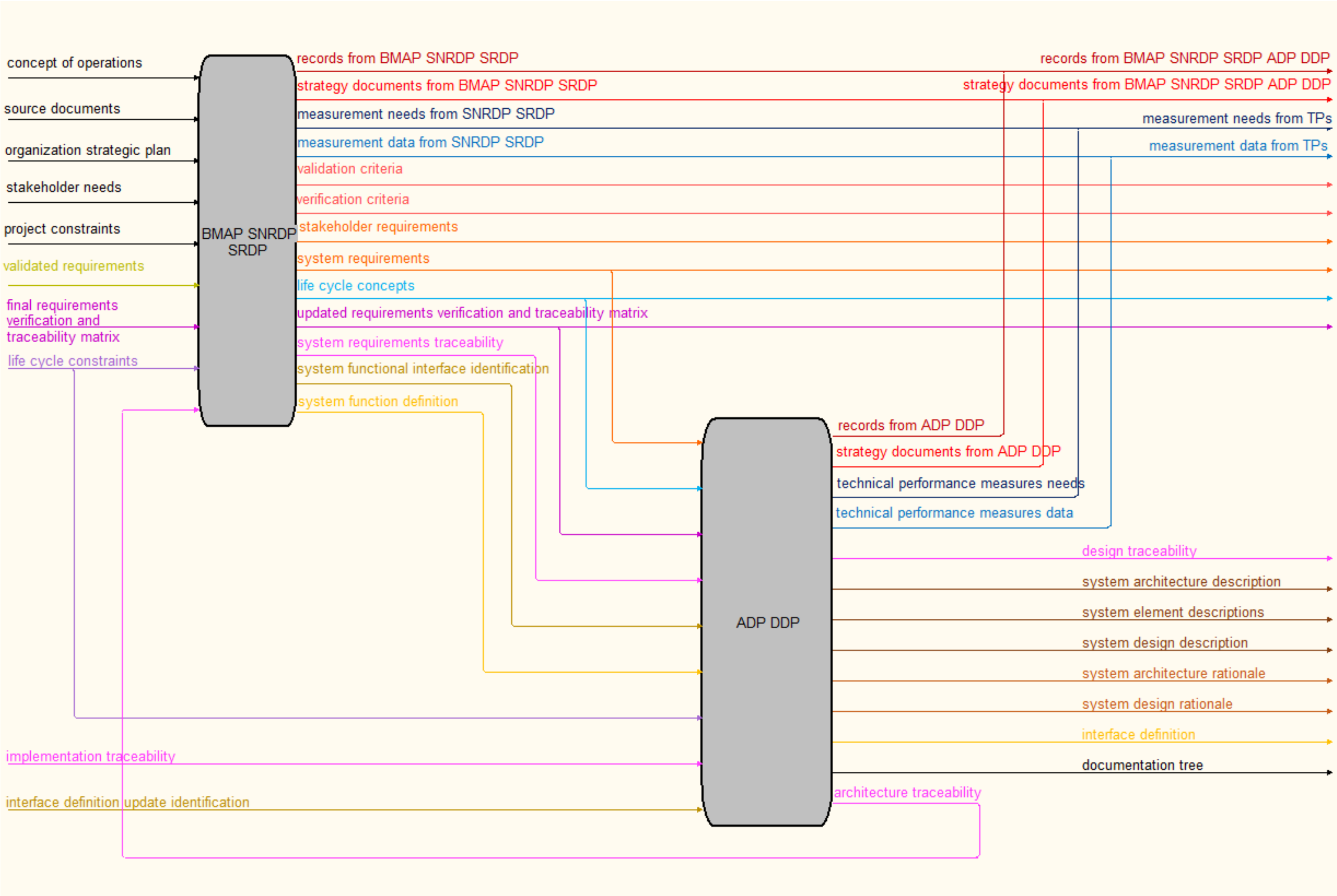
Appendix Figure 4 (A2) Technical Management Processes (TMPs)



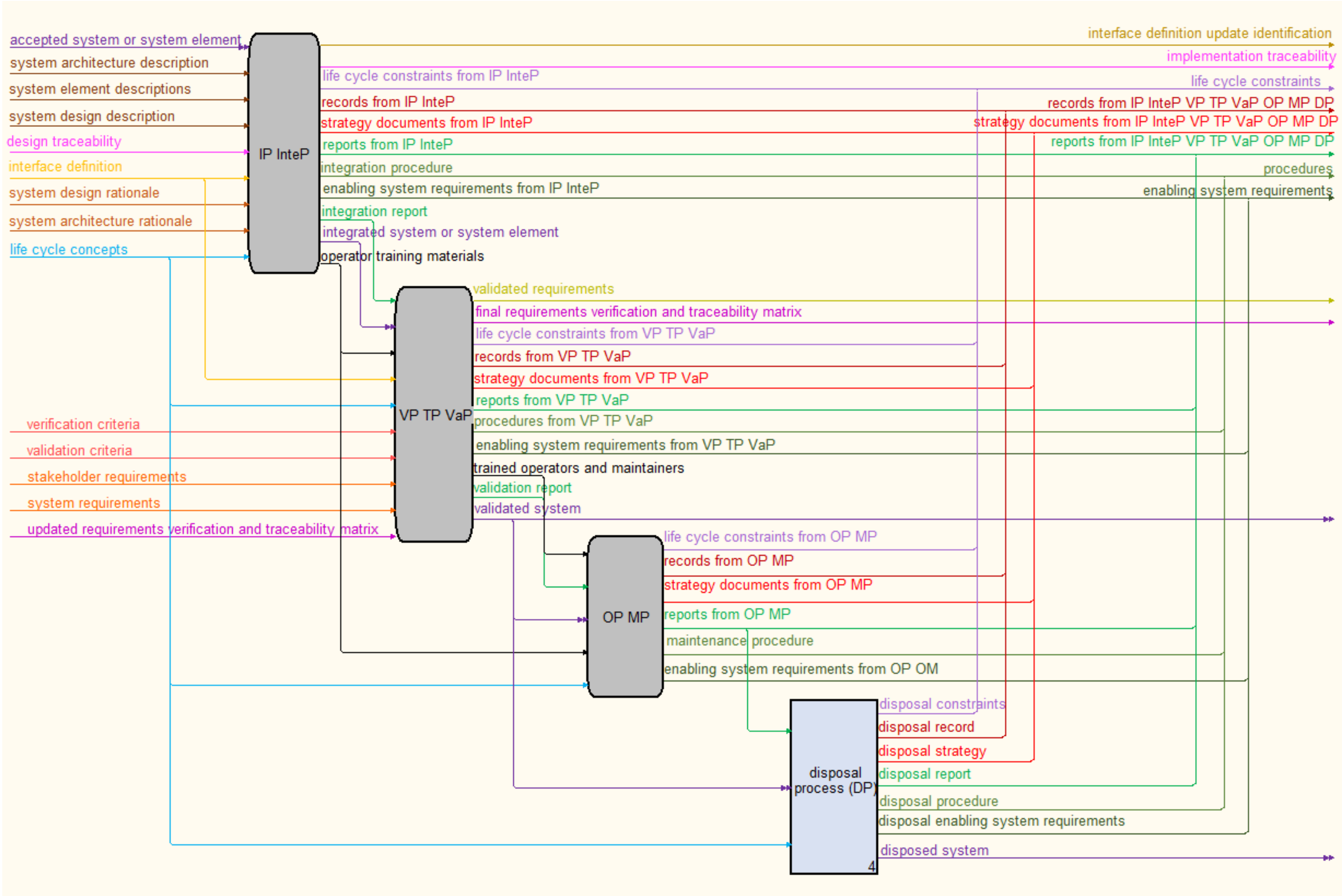
Appendix Figure 5 (A3) Agreement Processes (APs)



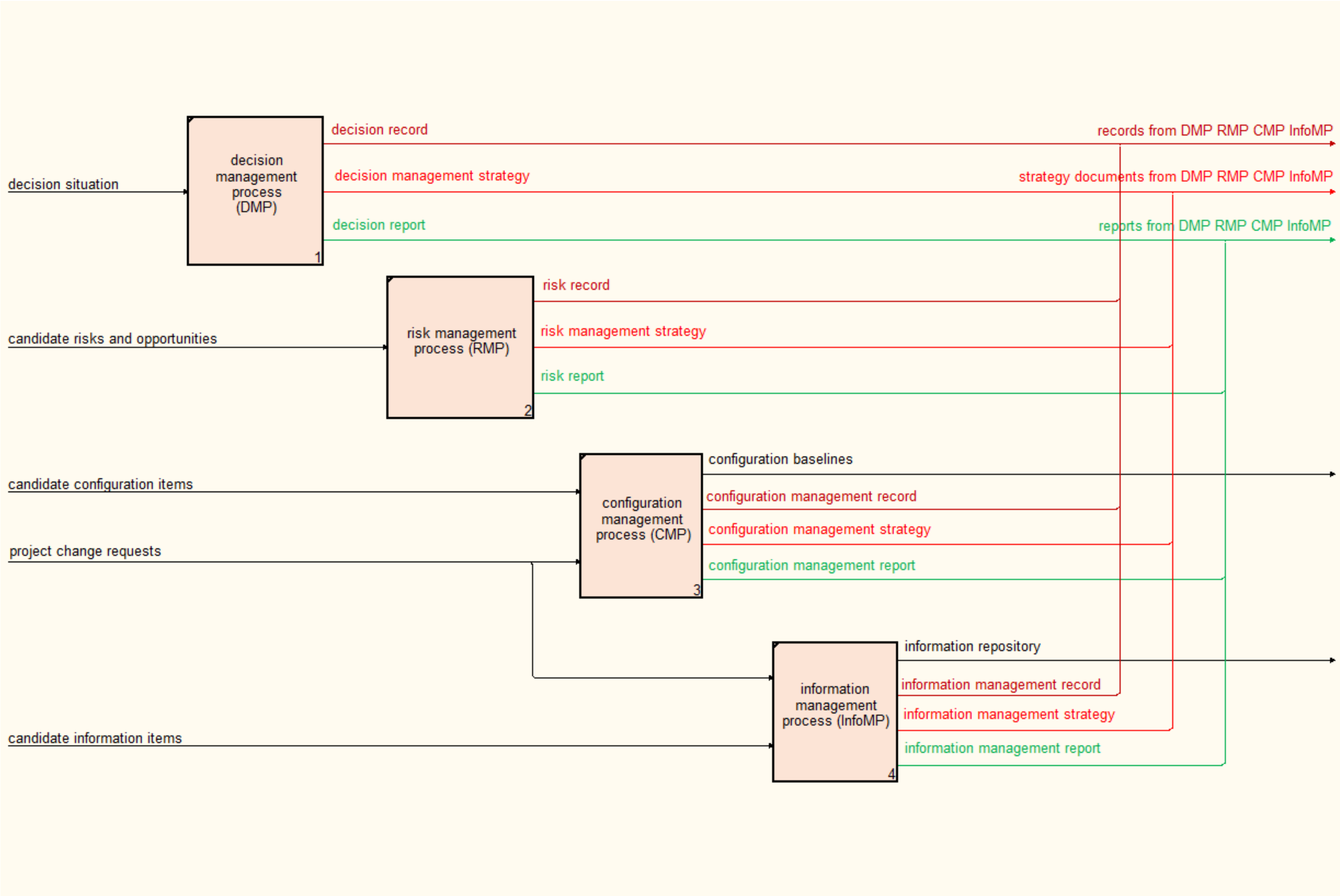
Appendix Figure 6 (A4) Organizational project-enabling Processes (OPPs)



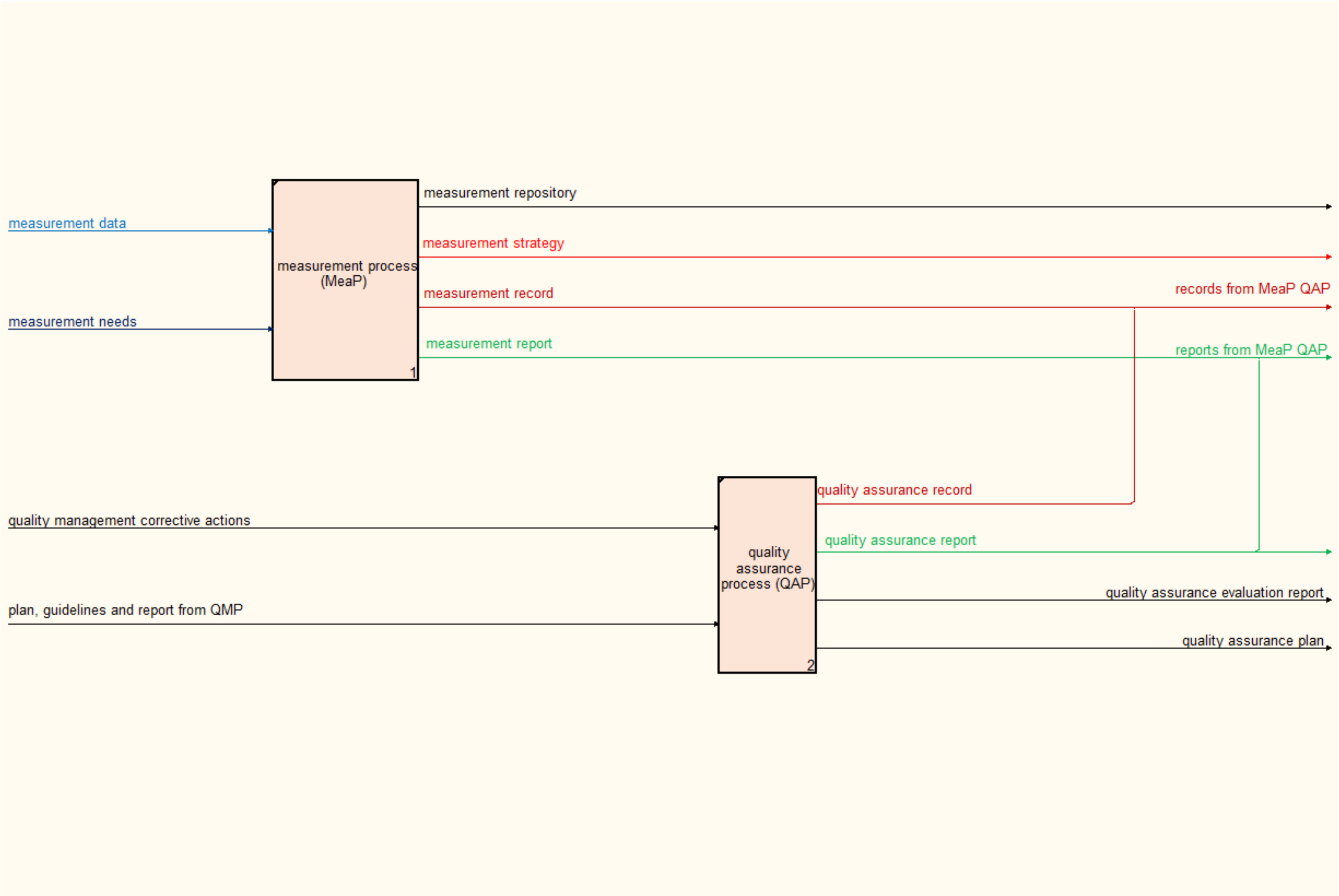
Appendix Figure 7 (A11) BMAP, SNRDP, SRDP, ADP and DDP



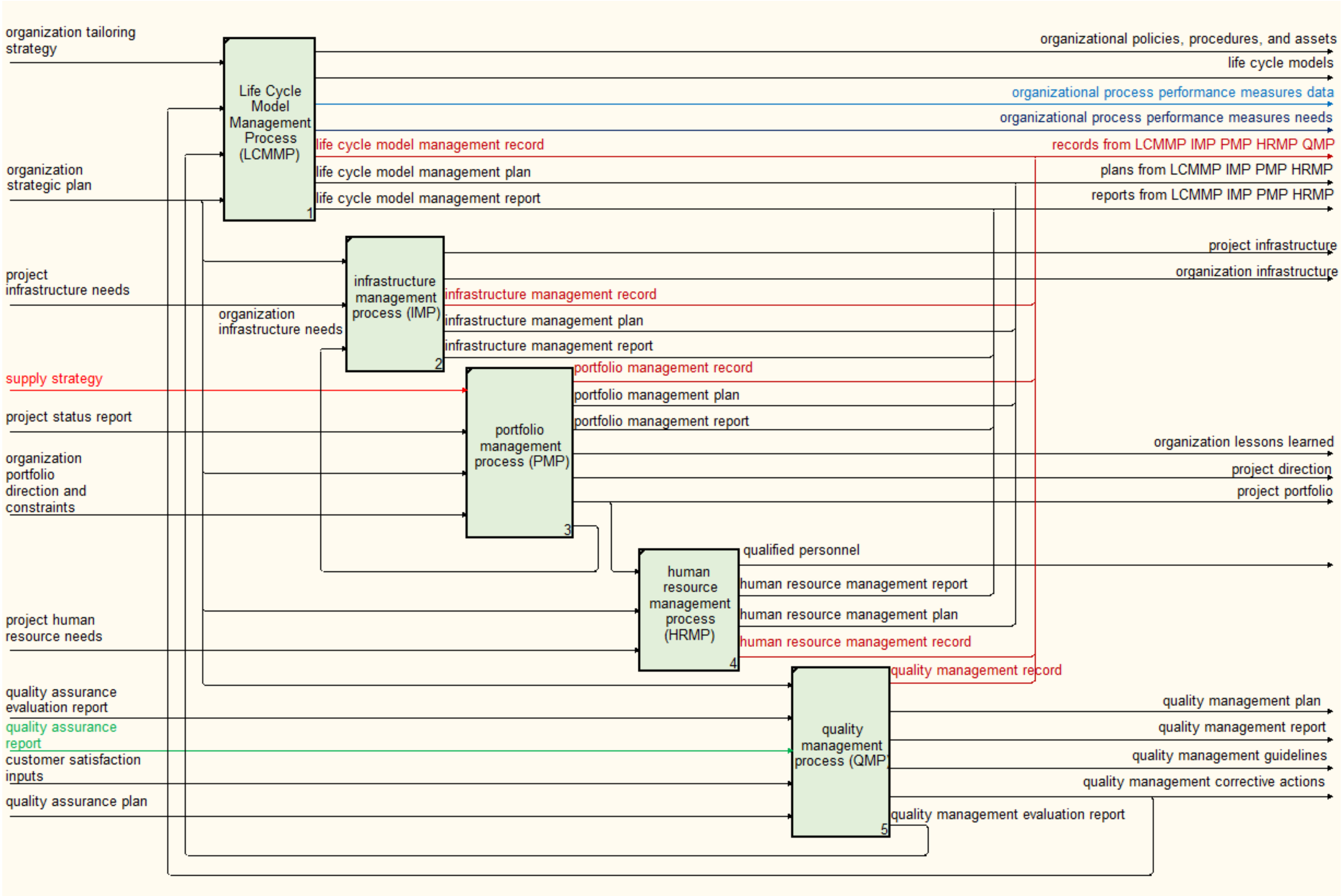
Appendix Figure 8 (A12) IP, InteP, VP, TP, VaP, OP, MP, and DP



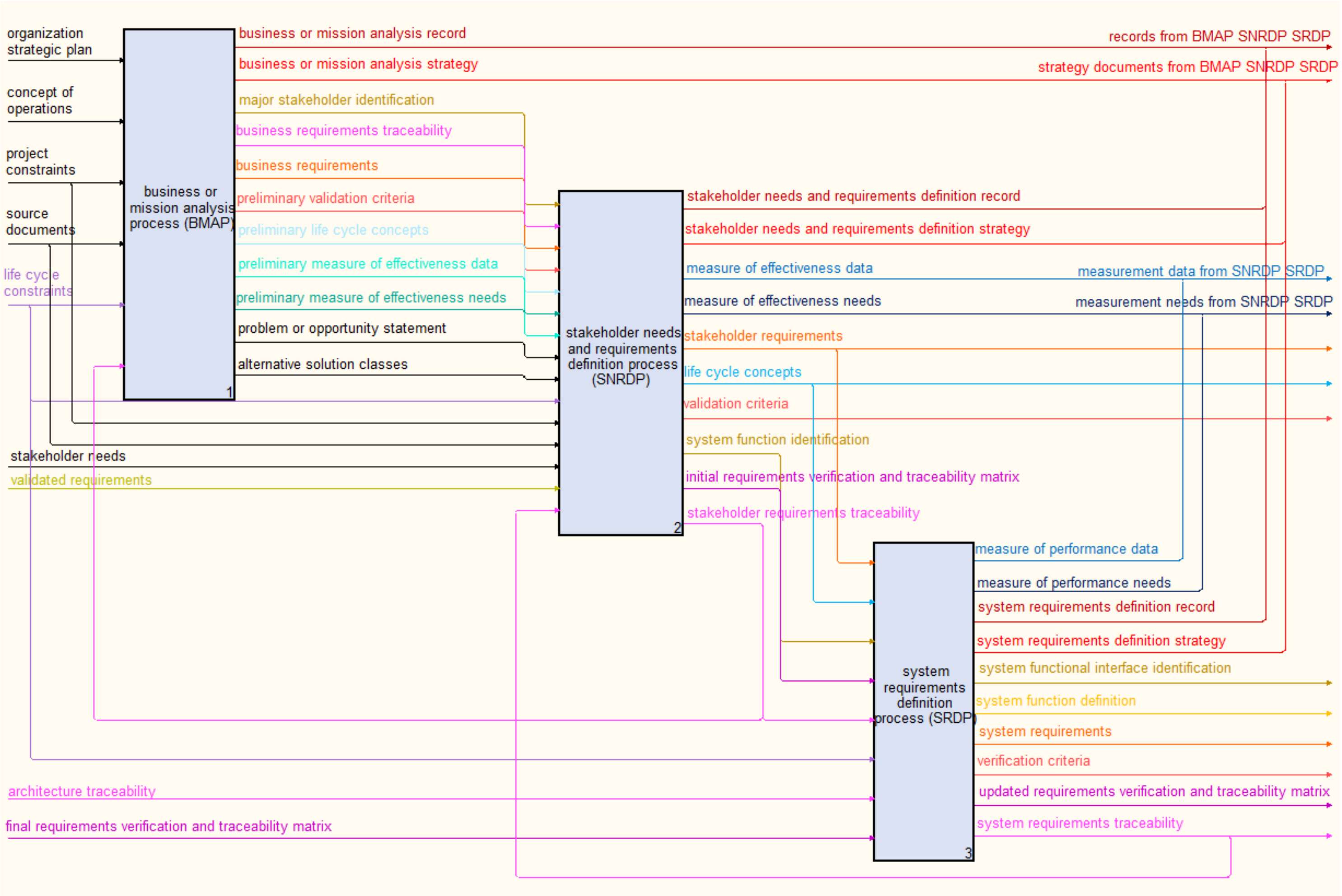
Appendix Figure 9 (A21) DMP, RMP, CMP, and InfoMP



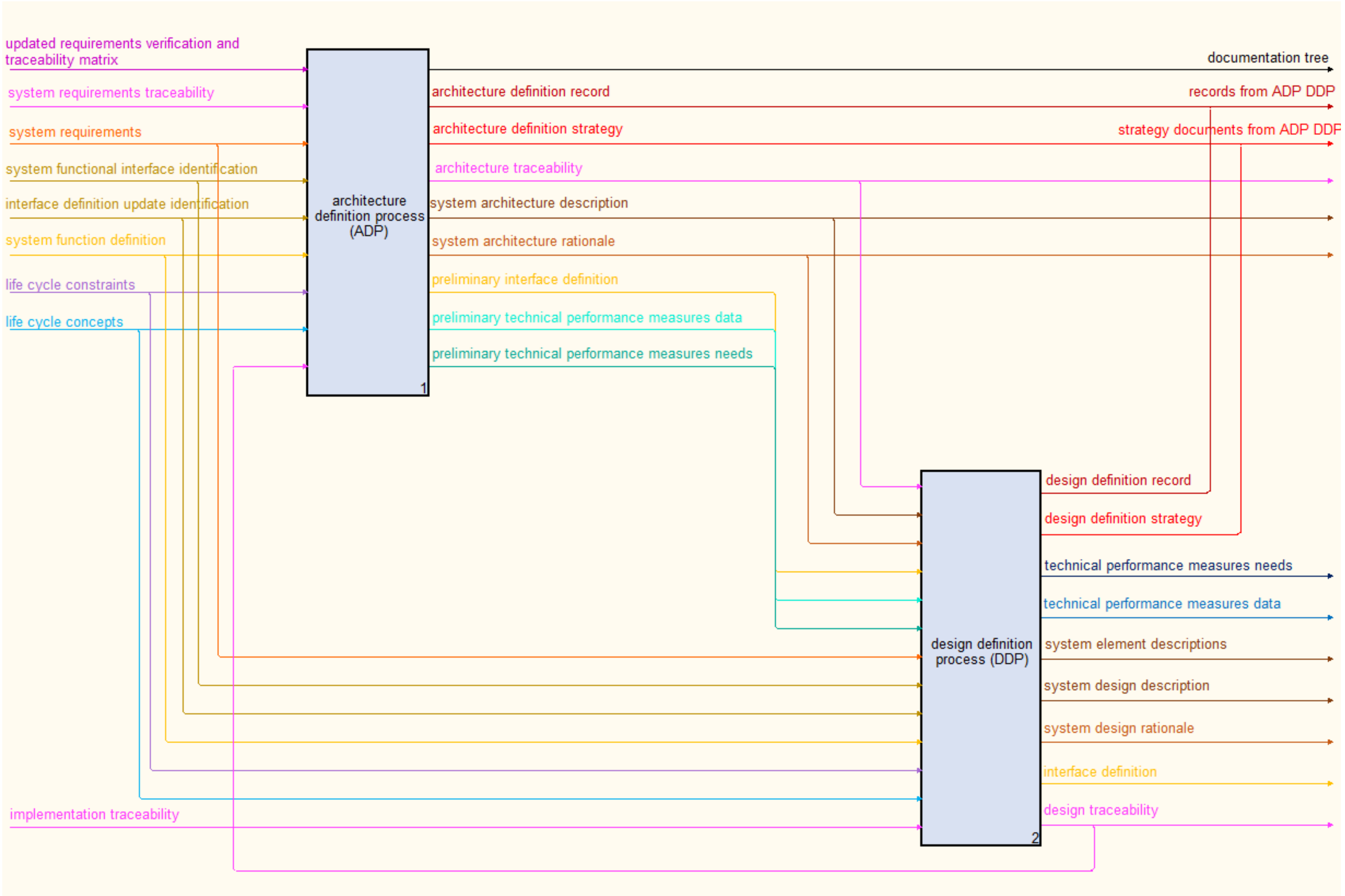
Appendix Figure 10 (A22) MeaP and QAP



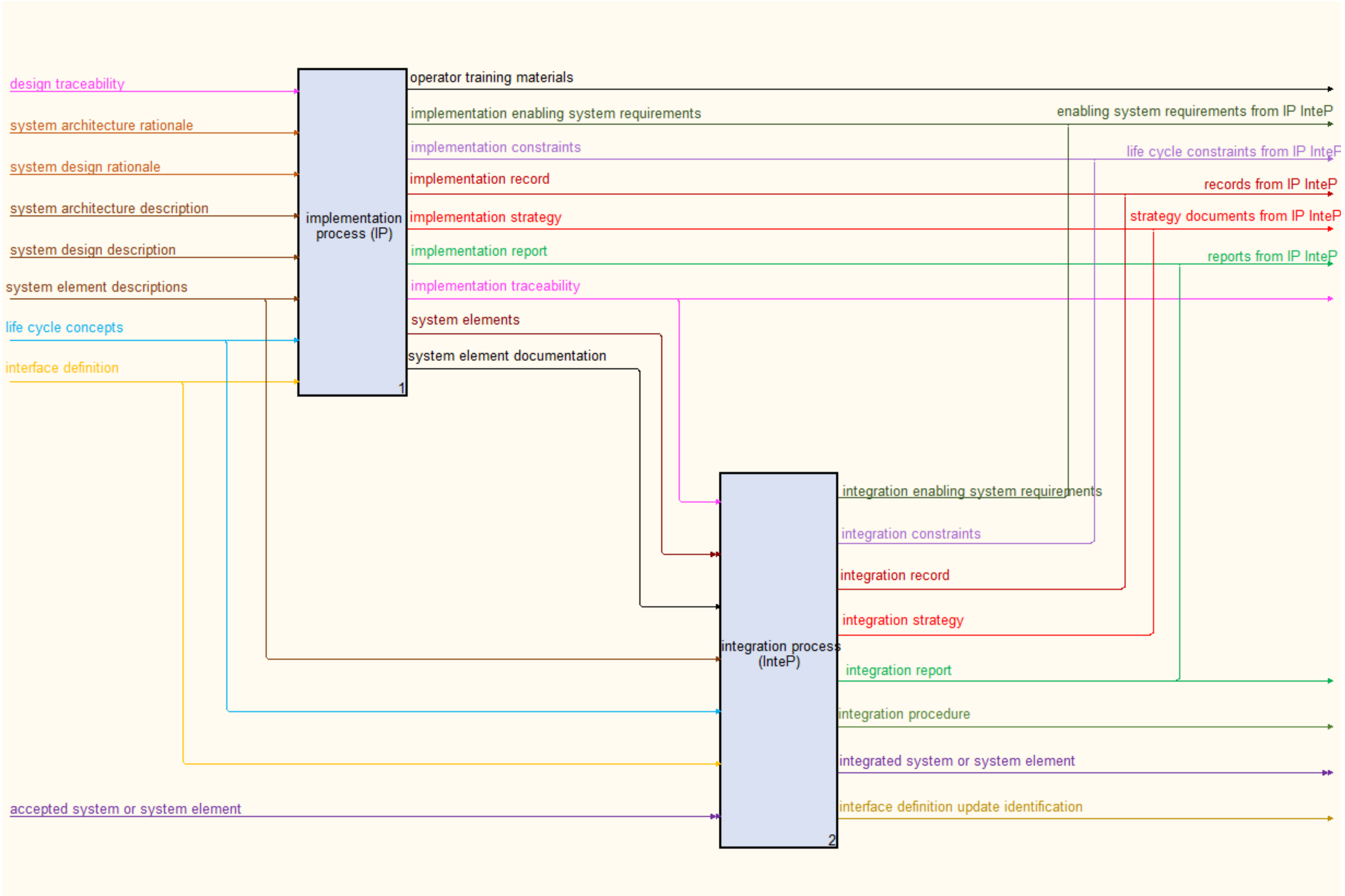
Appendix Figure 11 (A41) LCMMP, IMP, PMP, HRMP, and QMP



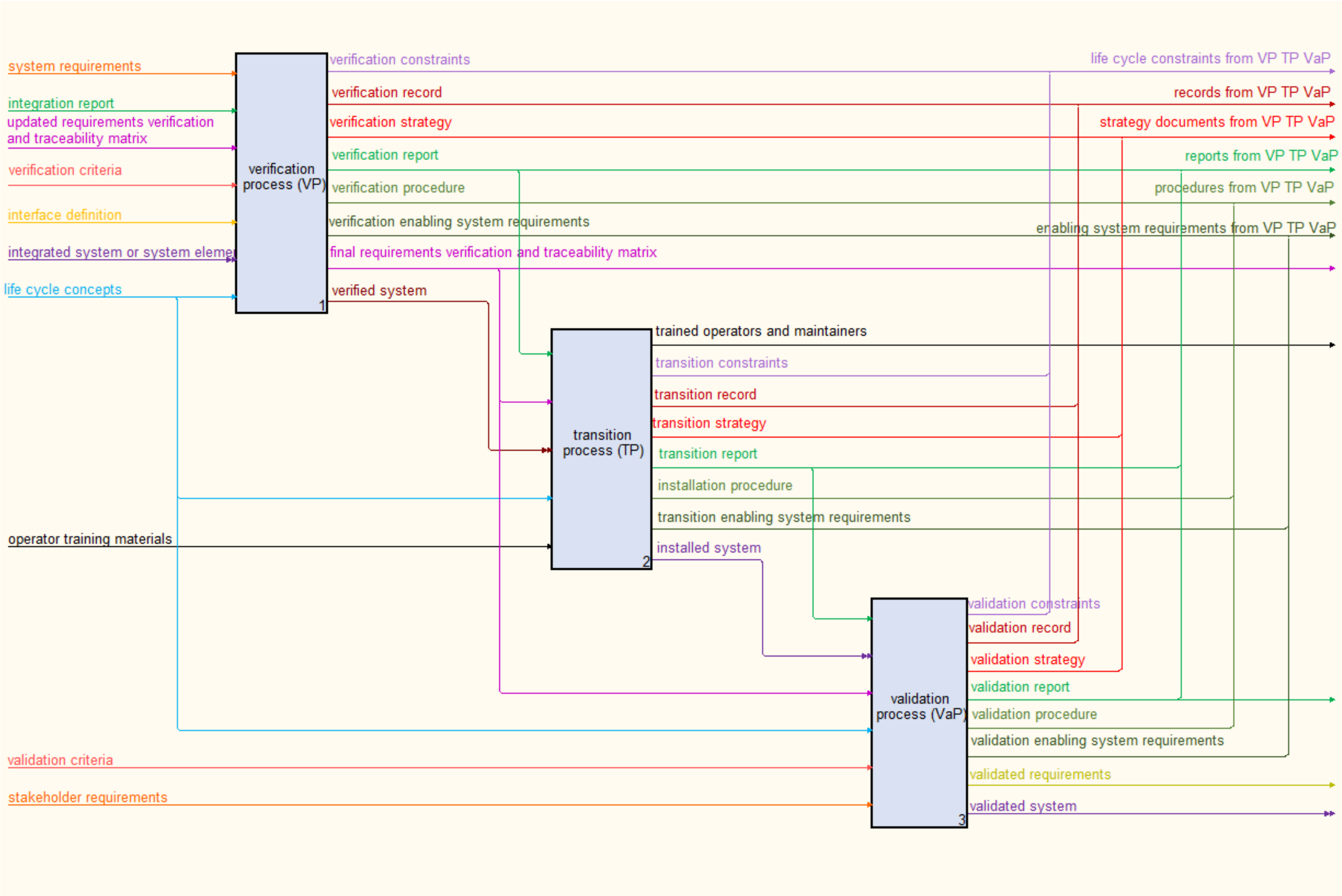
Appendix Figure 12 (A111) BMAP, SNRDP and SRDP



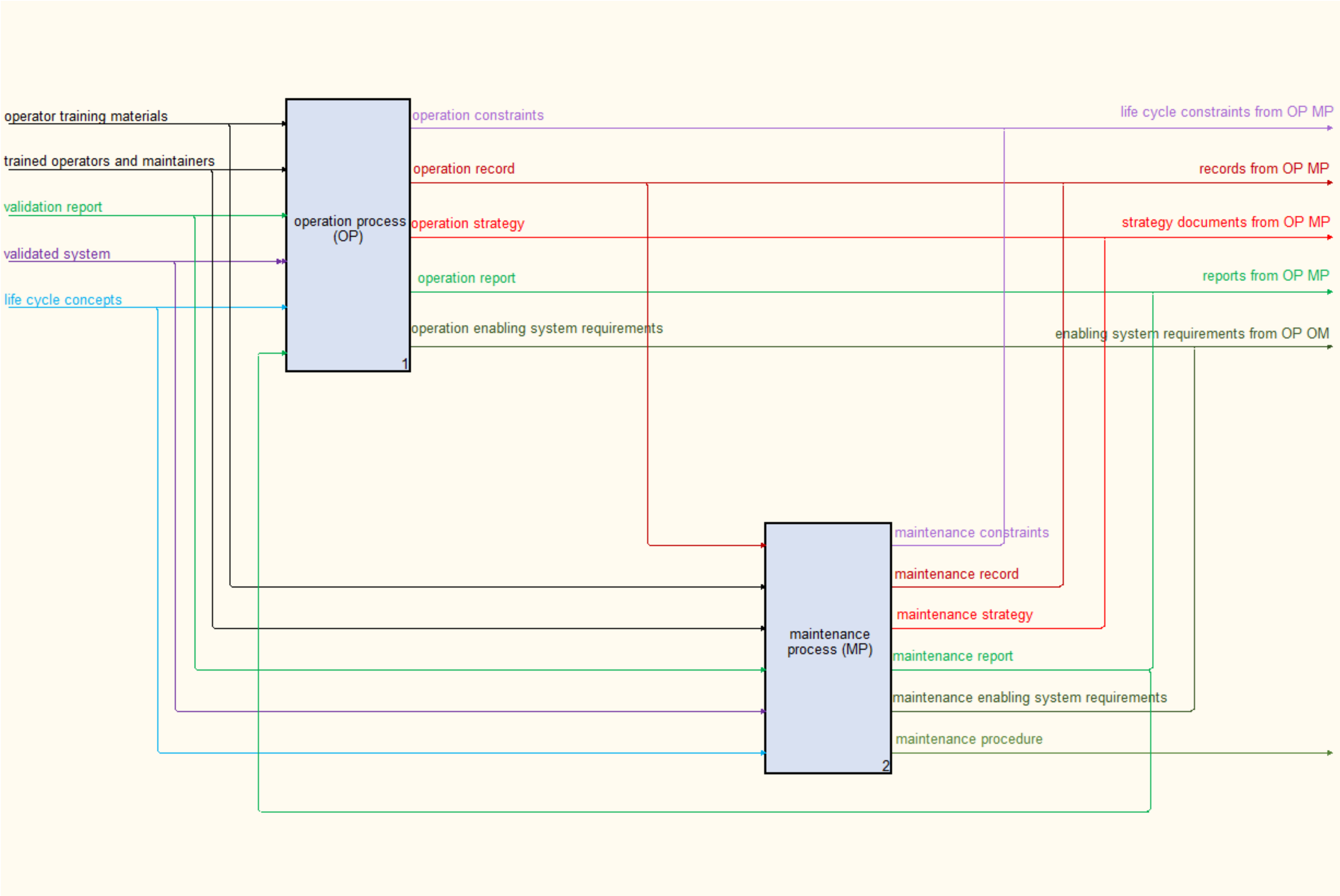
Appendix Figure 13 (A112) ADP and DDP



Appendix Figure 14 (A131) IP and InteP



Appendix Figure 15 (A132) VP, TP and VaP



Appendix Figure 16 (A133) OP and MP

Appendix 2

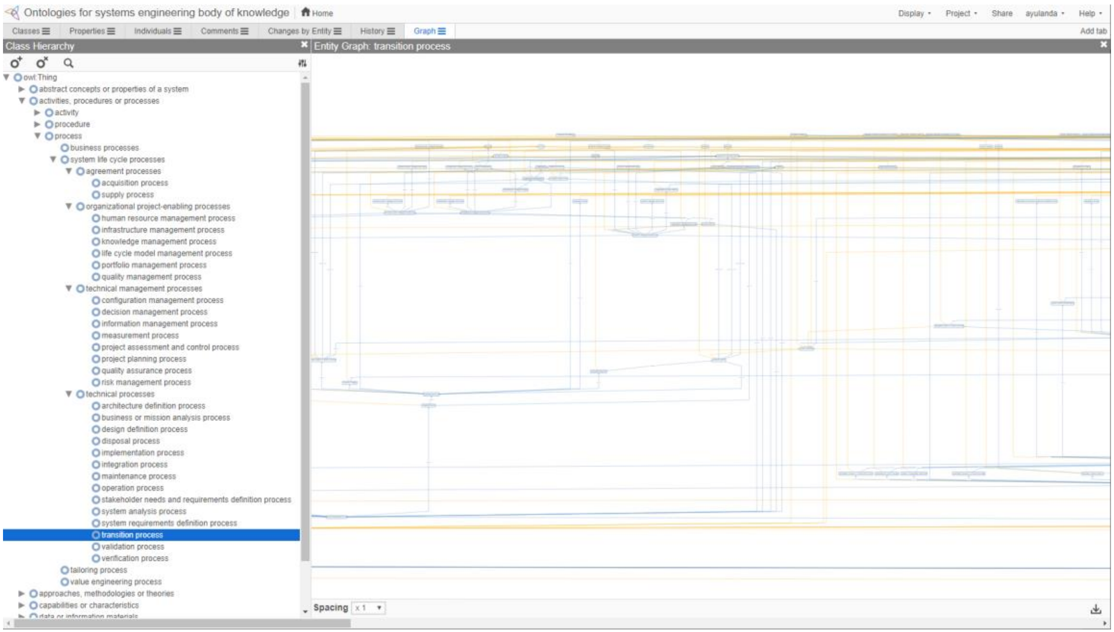
This appendix attaches the developed SE ontology visualized by three tools.

First, WebProtégé is used for editing the SE ontology. The link to the ontology is as follows.

https://webprotege.stanford.edu/#projects/903d1233-156b-49a8-9e32-13f05f4d0031/edit/Classes

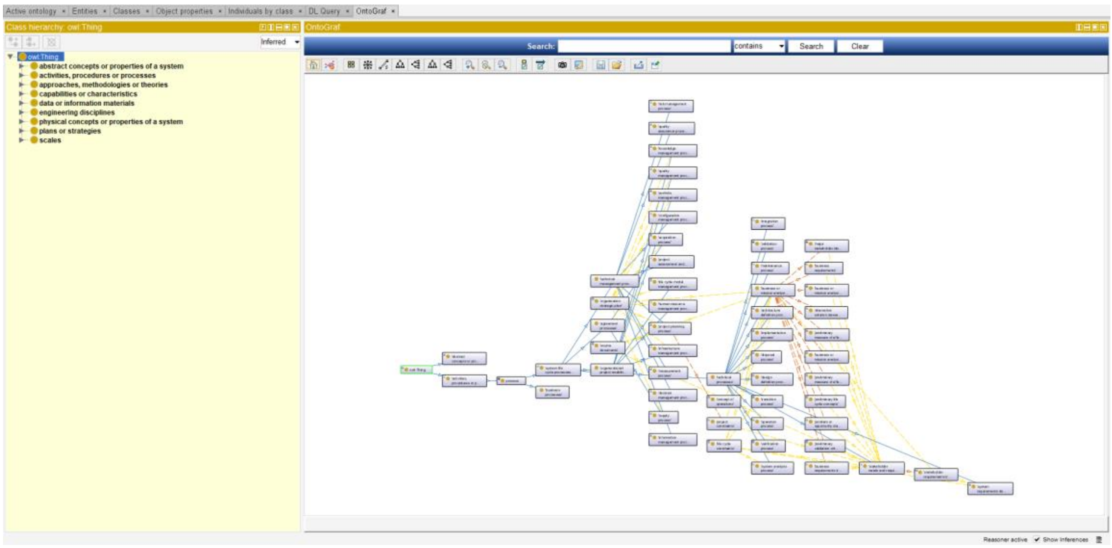
It is open for viewing and commenting to the public (register and login required).

The interface is shown in Appendix Figure 17.



Appendix Figure 17 The SE ontology on WebProtégé

Second, the source file of the SE ontology is in OWL, and all the reasoning is run by Protégé. Plenty of plugins can be used to view the ontology from different perspectives. For example, Appendix Figure 18 shows the concept hierarchy and the non-taxonomic relations between the system life cycle processes by OntoGraf.



Appendix Figure 18 The SE ontology on Protégé

Third, WebVOWL is used to analyze the clustering of the SE concepts. Appendix Figure 19 presents the SE ontology collapsed at the degree of 6.

Appendix Figure 19 The SE ontology on WEBVOWL

BIBLIOGRAPHY

Ackoff, R.L., 1971. Towards a system of systems concepts. Manage. Sci. 17, 661–671.

Adcock, R., Hutchison, N., Nielsen, C., 2016. Defining an architecture for the Systems Engineering Body of Knowledge. 10th Annu. Int. Syst. Conf. SysCon 2016 - Proc. 1–7. https://doi.org/10.1109/SYSCON.2016.7490640

Agrawal, V., 2016. Towards the ontology of ISO/IEC 27005:2011 risk management standard, in: Proceedings of the Tenth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2016). University of Plymouth, pp. 101–111.

Ahmad, M.N., Zakaria, N.H., Sedera, D., 2011. Ontology-based knowledge management for enterprise systems. Int. J. Enterp. Inf. Syst. 7, 64–90. https://doi.org/10.4018/jeis.2011100104

Al-Fedaghi, S., 2015. Heraclitean ontology for specifying systems. Int. Rev. Comput. Softw. 10, 602. https://doi.org/10.15866/irecos.v10i6.6493

Ali, N., Hong, J.-E., 2018. Failure detection and prevention for cyber-physical systems using ontology-based knowledge base. Computers 7, 68. https://doi.org/10.3390/computers7040068

Allen, M., Badecker, W., Osterhout, L., 2003. Morphological analysis in sentence processing: An ERP study. Lang. Cogn. Process. 18, 405–430. https://doi.org/10.1080/01690960244000054

Alobaidi, M., Malik, K.M., Hussain, M., 2018. Automated ontology generation framework powered by linked biomedical ontologies for disease-drug domain. Comput. Methods Programs Biomed. 165, 117–128. https://doi.org/10.1016/j.cmpb.2018.08.010

Alves, A., Antunes, B., Pereira, F.C., Bento, C., 2009. Semantic enrichment of places: Ontology learning from web. Int. J. Knowledge-based Intell. Eng. Syst. 13, 19–30. https://doi.org/10.3233/KES-2009-0170

Annamalai, G., Hussain, R., Cakkol, M., Roy, R., Evans, S., Tiwari, A., 2011. An ontology for product-service systems, in: Functional Thinking for Value Creation. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 231–236. https://doi.org/10.1007/978-3-642-19689-8_41

Ansaldi, S., Monti, M., Agnello, P., Giannini, F., 2012. An ontology for the identification of the most appropriate risk management methodology, in: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin, Heidelberg, pp. 444–453. https://doi.org/10.1007/978-3-642-33618-8_60

Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M., 2018. A survey of ontology learning techniques and applications. Database 2018. https://doi.org/10.1093/database/bay101

Aslaksen, E.W., Street, C., Leonards, S., 2011. Elements of a systems engineering ontology, in: 2011 Systems Engineering and Test and Evaluation in the Next Decade Symposium. Canberra, Australia, pp. 1–18.

Balduccini, M., Griffor, E., Huth, M., Vishik, C., Burns, M., Wollman, D., 2018. Ontology-based reasoning about the trustworthiness of cyber-physical systems, in: Living in the Internet of Things: Cybersecurity of the IoT - 2018. Institution of Engineering and Technology, pp. 12 (10 pp.)-12 (10 pp.). https://doi.org/10.1049/cp.2018.0012

Ball, G.L., Runge, C., 2014. Producing reusable engineered systems through ontology: implementing an information sciences approach to architecture-driven, model-based, concurrent engineering. J. Def. Model. Simul. Appl. Methodol. Technol. 11, 219–226. https://doi.org/10.1177/1548512913502259

Barforush, A.A., Rahnama, A., 2012. Ontology learning: Revisted. J. Web Eng. 11, 269–289.

Benali, H., Ben Saoud, N.B., Ben Ahmed, M., 2014. Context-based ontology to describe system-of-systems interoperability, in: 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA). IEEE, pp. 64–71. https://doi.org/10.1109/AICCSA.2014.7073180

Benslimane, S., Malki, M., Rahmouni, M., Rahmoun, A., 2008. Towards ontology extraction from data-intensive web sites: An HTML forms-based reverse engineering approach. Int. Arab J. Inf. Technol. 5, 34–44.

Bergholtz, M., Eriksson, O., 2015. Towards a socio-institutional ontology for conceptual modelling of information systems, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 225–235. https://doi.org/10.1007/978-3-319-25747-1_23

Bermejo-Alonso, J., Hernandez, C., Sanz, R., 2016. Model-based engineering of autonomous systems using ontologies and metamodels, in: 2016 IEEE International Symposium on Systems Engineering (ISSE). IEEE, pp. 1–8. https://doi.org/10.1109/SysEng.2016.7753185

Bertoa, M.F., Vallecillo, A., García, F., 2006. An ontology for software measurement, in: Calero, C., Ruiz, F., Piattini, M. (Eds.), Ontologies for Software Engineering and Software Technology. Springer Berlin Heidelberg, pp. 175–196. https://doi.org/10.1007/3-540-34518-3_6

Bhatt, M., Wouters, C., Flahive, A., Rahayu, W., Taniar, D., 2004. Semantic Completeness in Sub-ontology Extraction Using Distributed Methods, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 508–517. https://doi.org/10.1007/978-3-540-24767-8_53

Bird, S., Loper, E., 2020. Natural Language Toolkit [WWW Document]. URL http://www.nltk.org/ (accessed 3.8.20).

Birkholz, H., Sieverdingbeck, I., Sohr, K., Bormann, C., 2012. IO: an interconnected asset ontology in support of risk management processes, in: 2012 Seventh International Conference on Availability, Reliability and Security. IEEE, pp. 534–541. https://doi.org/10.1109/ARES.2012.73

BKCASE Editorial Board, 2017. The Guide to the Systems Engineering Body of Knowledge (SEBoK) [WWW Document]. Guid. to Syst. Eng. Body Knowl. (SEBoK), v. 1.9. R.D. Adcock (EIC). Hoboken, NJ Trust. Stevens Inst. Technol. URL www.sebokwiki.org (accessed 2.18.20).

Blanco, C., Lasheras, J., Valencia-Garc, R., Fern, E., Toval, A., Piattini, M., 2008. A systematic review and comparison of security ontologies, in: 2008 Third International Conference on Availability, Reliability and Security. IEEE, pp. 813–820. https://doi.org/10.1109/ARES.2008.33

Blokland, P.J., Reniers, G.L.L., 2018. An ontological and semantic foundation for safety science, in: Safety and Reliability - Safe Societies in a Changing World - Proceedings of the 28th International European Safety and Reliability Conference, ESREL 2018. pp. 3157–3166.

Borgo, S., Carrara, M., Garbacz, P., Vermaas, P.E., 2009. A formal ontological perspective on the behaviors and functions of technical artifacts. Artif. Intell. Eng. Des. Anal. Manuf. 23, 3. https://doi.org/10.1017/S0890060409000079

Borgo, S., Leitão, P., 2007. Foundations for a Core Ontology of Manufacturing, in: Ontologies. Springer US, Boston, MA, pp. 751–775. https://doi.org/10.1007/978-0-387-37022-4_27

Borst, P., Akkermans, H., Top, J., 1997. Engineering ontologies. Int. J. Hum. Comput. Stud. 46, 365–406. https://doi.org/10.1006/ijhc.1996.0096

Bouras, A., Aqle, A., Alsardy, A., Nizam, R., 2016. Ontologies for better project planning. J. Mod. Proj. Manag. 3, 74–79.

Brachman, Fikes, Levesque, 1983. Krypton: a functional approach to knowledge representation. Computer (Long. Beach. Calif). 16, 67–73. https://doi.org/10.1109/MC.1983.1654200

Brings, J., Daun, M., Hildebrandt, C., Törsleff, S., 2018. An ontological context modeling framework for coping with the dynamic contexts of cyber-physical systems, in: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development. SCITEPRESS - Science and Technology Publications, pp. 396–403. https://doi.org/10.5220/0006603403960403

Buede, D., Miller, W., 2016. Introduction to Systems Engineering, 3rd ed, The Engineering Design of Systems: Models and Methods. Hoboken, New Jersey, USA.

Cai, Y., Chen, W.-H., Leung, H.-F., Li, Q., Xie, H., Lau, R.Y.K., Min, H., Wang, F.L., 2016. Context-aware ontologies generation with basic level concepts from collaborative tags. Neurocomputing 208, 25–38. https://doi.org/10.1016/j.neucom.2016.02.070

Chandrasekaran, B., Josephson, J.R., Benjamins, V.R., 1999. What are ontologies, and why do we need them? IEEE Intell. Syst. 14, 20–26. https://doi.org/10.1109/5254.747902

Chen, X., Chen, C.-H., Leong, K.F., Jiang, X., 2013. An ontology learning system for customer needs representation in product development. Int. J. Adv. Manuf. Technol. 67, 441–453. https://doi.org/10.1007/s00170-012-4496-2

Chourabi, O., Pollet, Y., Ben, M., 2010. An ontological framework for knowledge management in systems engineering processes, in: Knowledge Management. InTech, pp. 149–169. https://doi.org/10.5772/9554

Christophe, F., Sell, R., Bernard, A., Coatanéa, E., 2009. OPAS: ontology processing for assisted synthesis of conceptual design solutions, in: Volume 5: 35th Design Automation Conference, Parts A and B. ASME, pp. 249–260. https://doi.org/10.1115/DETC2009-87776

Cimiano, P., 2006. Ontology Learning and Population from Text, Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer US. https://doi.org/10.1007/978-0-387-39252-3

Cimiano, P., Völker, J., 2005. Towards large-scale, open-domain and ontology-based named entity classification. Int. Conf. Recent Adv. Nat. Lang. Process. RANLP 2005-Janua, 166–172.

Cimiano, P., Völker, J., Studer, R., 2006. Ontologies on demand? A description of the state-of-the-art, applications, challenges and trends for ontology learning from text. Information-wiss. und Prax. 57, 315–320.

Colace, F., De Santo, M., Greco, L., Amato, F., Moscato, V., Picariello, A., 2014. Terminological ontology learning and population using latent Dirichlet allocation. J. Vis. Lang. Comput. 25, 818–826. https://doi.org/10.1016/j.jvlc.2014.11.001

Cronin, P., Ryan, F., Coughlan, M., 2008. Undertaking a literature review: a step-by-step approach. Br. J. Nurs. 17, 38–43. https://doi.org/10.12968/bjon.2008.17.1.28059

Cruz, S.Y., Silega, M.N., Parra, F.A., Gómez, B.O., 2018. Description and Analysis of Design Decisions: An Ontological Approach, in: Valencia-García, R., Alcaraz-Mármol, G., Cioppo-Morstadt, J. Del, Vera-Lucio, N., Bucaram-Leverone, M. (Eds.), Communications in Computer and Information Science. Springer, Cham, pp. 174–185. https://doi.org/10.1007/978-3-030-00940-3_13

Dahlberg, R., 2016. Bridging the Gap, in: Kotsireas, I., Nagurney, A., Pardalos, P. (Eds.), Dynamics of Disasters—Key Concepts, Models, Algorithms, and Insights. Springer, Cham, pp. 37–56. https://doi.org/10.1007/978-3-319-43709-5_3

Daun, M., Brings, J., Weyer, T., Tenbergen, B., 2016. Fostering concurrent engineering of cyber-physical systems a proposal for an ontological context framework, in: 2016 3rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC). IEEE, pp. 5–10. https://doi.org/10.1109/EITEC.2016.7503689

David, J., Guillet, F., Briand, H., 2006. Matching directories and OWL ontologies with AROMA. Int. Conf. Inf. Knowl. Manag. Proc. 830–831. https://doi.org/10.1145/1183614.1183752

Derler, P., Lee, E.A., Vincentelli, A.S., 2012. Modeling Cyber–Physical Systems. Proc. IEEE 100, 13–28. https://doi.org/10.1109/JPROC.2011.2160929

Di Maio, P., 2011. Toward a semantic vocabulary for systems engineering, in: Proceedings of the International Conference on Web Intelligence, Mining and Semantics - WIMS '11. ACM Press, New York, New York, USA, p. 1. https://doi.org/10.1145/1988688.1988712

Di Maio, P., 2010. Digital ecosystems for knowledge management in systems engineering, in: 4th IEEE International Conference on Digital Ecosystems and Technologies. IEEE, pp. 482–487. https://doi.org/10.1109/DEST.2010.5610603

Dibowski, H., Ploennigs, J., Wollschlaeger, M., 2018. Semantic Device and System Modeling for Automation Systems and Sensor Networks. IEEE Trans. Ind. Informatics 14, 1298–1311. https://doi.org/10.1109/TII.2018.2796861

Ding, Y., Foo, S., 2002. Ontology research and development. Part 1 - a review of ontology generation. J. Inf. Sci. 28, 123–136. https://doi.org/10.1177/016555150202800204

Dogan, H., Barot, V., Henshaw, M., Siemieniuch, C., Sinclair, M., 2014. Systems of systems engineering thesaurus approach: from concept to realisation. Int. J. Syst. Syst. Eng. 5, 228. https://doi.org/10.1504/IJSSE.2014.065751

Dogan, H., Henshaw, M.J. de C., Johnson, J., 2012. An incremental hybridisation of heterogeneous case studies to develop an ontology for capability engineering, in: INCOSE International Symposium. pp. 956–971. https://doi.org/10.1002/j.2334-5837.2012.tb01382.x

Dombayci, C., Farreres, J., Rodríguez, H., Muñoz, E., Capón-García, E., Espuña, A., Graells, M., 2015. On the process of building a process systems engineering ontology using a semi-automatic construction approach, in: Computer Aided Chemical Engineering. pp. 941–946. https://doi.org/10.1016/B978-0-444-63577-8.50002-4

Dong, H., Hussain, F.K., 2014. Self-Adaptive Semantic Focused Crawler for Mining Services Information Discovery. IEEE Trans. Ind. Informatics 10, 1616–1626. https://doi.org/10.1109/TII.2012.2234472

Dong, H., Hussain, F.K., 2013. SOF: a semi-supervised ontology-learning-based focused crawler. Concurr. Comput. Pract. Exp. 25, 1755–1770. https://doi.org/10.1002/cpe.2980

Dong, M., Yang, D., Su, L., 2011. Ontology-based service product configuration system modeling and development. Expert Syst. Appl. 38, 11770–11786. https://doi.org/10.1016/j.eswa.2011.03.064

Dori, D., 2016. Model-based systems engineering with OPM and SysML, 1st ed. Springer-Verlag New York, New York.

Dori, D., 2002. Object-process methodology. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-56209-9

Dori, D., Jbara, A., Levi, N., Wengrowicz, N., 2018. Object-process methodology, OPM ISO 19450 - OPCloud and the evolution of OPM modeling tools [WWW Document]. Syst. Eng. Lett. Proj. Perform. Int. SyEN 61. URL https://www.ppi-int.com/syen61-a1/ (accessed 4.30.19).

Dori, D., Reinhartz-Berger, I., 2003. An OPM-based metamodel of system development process, in: Lecture Notes in Computer Science. pp. 105–117.

Dori, D., Sillitto, H., 2017. What is a system? An ontological framework. Syst. Eng. 20, 207–219. https://doi.org/10.1002/sys.21383

Dwivedi, Y.K., Ravichandran, K., Williams, M.D., Miller, S., Lal, B., Antony, G. V., Kartik, M., 2013. IS/IT Project Failures: A Review of the Extant Literature for Deriving a Taxonomy of Failure Factors, in: Dwivedi, Y.K., Henriksen, H.Z., Wastell, D., Dé, R. (Eds.), Grand Successes and Failures in IT. Public and Private Sectors. TDIT 2013. IFIP Advances in Information and Communication Technology. Springer, Berlin, Heidelberg, pp. 73–88. https://doi.org/10.1007/978-3-642-38862-0_5

Easterbrook, S., 2014. From computational thinking to dystems thinking: a conceptual toolkit for sustainability computing, in: Proceedings of the 2014 Conference ICT for Sustainability. Atlantis Press, Paris, France, pp. 235–244. https://doi.org/10.2991/ict4s-14.2014.28

Eisner, H., 2008. Definitions of systems engineering, in: Eisner, H. (Ed.), Essentials of Project and Systems Engineering Management. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, pp. 1–40.

Eito-Brun, R., 2018. Specification and deployment of a semantic database for system configuration management, in: Stamelos, I., O'Connor, R. V., Rout, T., Dorling, A. (Eds.), Software Process Improvement and Capability Determination. SPICE 2018. Communications in Computer and Information Science. Springer, Cham, pp. 263–271. https://doi.org/10.1007/978-3-030-00623-5_18

Eito-Brun, R., 2016. Design of an ontologies for the exchange of software engineering data in the aerospace industry, in: Communications in Computer and Information Science. pp. 71–78. https://doi.org/10.1007/978-3-319-45880-9_6

El Kadiri, S., Kiritsis, D., 2015. Ontologies in the context of product lifecycle management: state of the art literature review. Int. J. Prod. Res. 53, 5657–5668. https://doi.org/10.1080/00207543.2015.1052155

Engel, G., Greiner, T., Seifert, S., 2018. Ontology-Assisted Engineering of Cyber–Physical Production Systems in the Field of Process Technology. IEEE Trans. Ind. Informatics 14, 2792–2802. https://doi.org/10.1109/TII.2018.2805320

Ernadote, D., 2017. Ontology-based pattern for system engineering, in: 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS). IEEE, pp. 248–258. https://doi.org/10.1109/MODELS.2017.4

Ernadote, D., 2015. An ontology mindset for system engineering, in: 2015 IEEE International Symposium on Systems Engineering (ISSE). IEEE, pp. 454–460. https://doi.org/10.1109/SysEng.2015.7302797

Eskins, D., Sanders, W.H., 2011. The multiple-asymmetric-utility system model: a framework for modeling cyber-human systems, in: 2011 Eighth International Conference on Quantitative Evaluation of SysTems. IEEE, pp. 233–242. https://doi.org/10.1109/QEST.2011.38

Fensel, D., Facca, F.M., Simperl, E., Toma, I., 2011. Web service modeling ontology, in: Semantic Web Services. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 107–129. https://doi.org/10.1007/978-3-642-19193-0_7

Ferchichi, A., Bigand, M., Lefèbvre, H., 2008. An ontology for quality standards integration in software collaborative projects, in: CEUR Workshop Proceedings. pp. 17–30.

Ferreira, S., Tejeda, J., 2011. An ontology for unmanned and autonomous systems of systems test and evaluation, in: INCOSE International Symposium. pp. 1082–1091. https://doi.org/10.1002/j.2334-5837.2011.tb01267.x

Figueiredo, G., Duchardt, A., Hedblom, M.M., Guizzardi, G., 2018. Breaking into pieces: An ontological approach to conceptual model complexity management, in: 2018 12th International Conference on Research Challenges in Information Science (RCIS). IEEE, pp. 1–10. https://doi.org/10.1109/RCIS.2018.8406642

Forsberg, K., Roedler, G., 2011. Knowledge Management and the INCOSE Knowledge Management Working Group. INSIGHT 14, 13–15. https://doi.org/10.1002/inst.201114113

Forsberg, K., Roedler, G., Ring, J., Hodgson, R., Sillitto, H., Jorgensen, R., 2010. 5.1.0 The Future of Knowledge Management in INCOSE. INCOSE Int. Symp. 20, 605–628. https://doi.org/10.1002/j.2334-5837.2010.tb01092.x

Fraga, A., Llorens, J., 2015. How to build ontologies for requirements systems engineering projects aiding the quality management process, in: Fred, A., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (Eds.), Knowledge Discovery, Knowledge Engineering and Knowledge Management. Springer Verlag, pp. 457–476. https://doi.org/10.1007/978-3-319-25840-9_28

Fraga, A., Llorens, J., Alonso, L., Fuentes, J.M., 2015. Ontology-Assisted Systems Engineering Process with Focus in the Requirements Engineering Process, in: Boulanger, F., Krob, D., Morel, G., Roussel, J.-C. (Eds.), Complex Systems Design & Management. Springer International Publishing, Cham, pp. 149–161.

Friedman, R.S., Prusak, L., 2008. On heuristics, narrative and knowledge management. Technovation 28, 812–817. https://doi.org/10.1016/j.technovation.2008.07.002

Gacitua, R., Sawyer, P., Rayson, P., 2008. A flexible framework to experiment with ontology learning techniques. Knowledge-Based Syst. 21, 192–199. https://doi.org/10.1016/j.knosys.2007.11.009

Gaeta, M., Orciuoli, F., Paolozzi, S., Salerno, S., 2011. Ontology Extraction for Knowledge Reuse: The e-Learning Perspective. IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans 41, 798–809. https://doi.org/10.1109/TSMCA.2011.2132713

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L., 2002. Sweetening Ontologies with DOLCE. pp. 166–181. https://doi.org/10.1007/3-540-45810-7_18

Gaševic, D., Djuric, D., Devedžic, V., 2009. Ontologies, in: Model Driven Engineering and Ontology Development. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 45–80. https://doi.org/10.1007/978-3-642-00282-3_2

Geisler, S., Quix, C., Weber, S., Jarke, M., 2016. Ontology-based data quality management for data streams. J. Data Inf. Qual. 7, 1–34. https://doi.org/10.1145/2968332

Gero, J.S., Kannengiesser, U., 2007. A function–behavior–structure ontology of processes. AI EDAM 21. https://doi.org/10.1017/S0890060407000340

Gharajedaghi, J., 2011. Interdependency and choice, in: Chester, P., Roumeliotis, R., Day, R. (Eds.), Systems Thinking Managing Chaos and Complexity: A Platform for Designing Business Architecture. Elsevier, Inc., Burlington, MA, p. 9.

Giachetti, R.E., 2015. Evaluation of the DoDAF meta-model's support of systems engineering, in: Procedia Computer Science. pp. 254–260. https://doi.org/10.1016/j.procs.2015.09.208

Gil, R., Martin-Bautista, M.J., 2014. SMOL: a systemic methodology for ontology learning from heterogeneous sources. J. Intell. Inf. Syst. 42, 415–455. https://doi.org/10.1007/s10844-013-0296-x

Givehchi, O., Landsdorf, K., Simoens, P., Colombo, A.W., 2017. Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems. IEEE Trans. Ind. Informatics 13, 3370–3378. https://doi.org/10.1109/TII.2017.2740434

Gómez-Pérez, A., Manzano-Macho, D., 2005. An overview of methods and tools for ontology learning from texts. Knowl. Eng. Rev. 19, 187–212. https://doi.org/10.1017/S0269888905000251

Gonsalves, T., Itoh, K., 2008. Service systems performance ontology. J. Integr. Des. Process Sci. 12, 1–6.

Gonzalez-Perez, C., Henderson-Sellers, B., McBride, T., Low, G.C., Larrucea, X., 2016. An Ontology for ISO software engineering standards: 2) Proof of concept and application. Comput. Stand. Interfaces 48, 112–123. https://doi.org/10.1016/j.csi.2016.04.007

Gorshkov, S., Kralin, S., Miroshnichenko, M., 2016. Multi-viewpoint ontologies for decision-making Support, in: Ngomo, A.-C.N., Křemen, P. (Eds.), Communications in Computer and Information Science. Springer, Cham, pp. 3–17. https://doi.org/10.1007/978-3-319-45880-9_1

Green, P.F., Rosemann, M., Indulska, M., 2005. Ontological evaluation of enterprise systems interoperability using ebXML. IEEE Trans. Knowl. Data Eng. 17, 713–725. https://doi.org/10.1109/TKDE.2005.79

Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum. Comput. Stud. 43, 907–928. https://doi.org/10.1006/ijhc.1995.1081

Grubic, T., Fan, I.-S., 2010. Supply chain ontology: review, analysis and synthesis. Comput. Ind. 61, 776–786. https://doi.org/10.1016/j.compind.2010.05.006

Guarino, N., 1998. Formal Ontology and Information Systems. Form. Ontol. Inf. Syst. Proc. first Int. Conf. 46, 81–97. https://doi.org/10.1.1.29.1776

Guarino, N., 1992. Concepts, attributes and arbitrary relations. Data Knowl. Eng. 8, 249–261. https://doi.org/10.1016/0169-023X(92)90025-7

Guarino, N., Giaretta, P., 1995. Ontologies and knowledge bases: towards a terminological clarification, in: Mars, N.J.I. (Ed.), Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing. IOS Press, Amsterdam, The Netherlands, pp. 25--32.

Guessi, M., Moreira, D.A., Abdalla, G., Oquendo, F., Nakagawa, E.Y., 2015. OntolAD: a formal ontology for architectural descriptions, in: Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC '15. ACM Press, New York, New York, USA, pp. 1417–1424. https://doi.org/10.1145/2695664.2695739

Guizzardi, G., Wagner, G., Almeida, J.P.A., Guizzardi, R.S.S., 2015. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. Appl. Ontol. 10, 259–271. https://doi.org/10.3233/AO-150157

Guo, C.P., Nunes, B.M., 2009. Surfacing ERP exploitation risks through a risk ontology. Ind. Manag. Data Syst. 109, 926–942. https://doi.org/10.1108/02635570910982283

Guo, J., Xu, L.D., Xiao, G., Gong, Z., 2012. Improving Multilingual Semantic Interoperation in Cross-Organizational Enterprise Systems Through Concept Disambiguation. IEEE Trans. Ind. Informatics 8, 647–658. https://doi.org/10.1109/TII.2012.2188899

Gutierrez, R., 2018. A Requirement Ontology To Guide The Analysis Of System Life Cycle Processes. Concordia University.

Hahn, A., große Austing, S., Häusler, S., Reinelt, M., 2008. Requirements for Ontology Based Design Project Assessment, in: Margaria, T., Steffen, B. (Eds.), Communications in Computer and Information Science. Springer, Berlin, Heidelberg, pp. 684–692. https://doi.org/10.1007/978-3-540-88479-8_49

Hallberg, N., Jungert, E., Pilemalm, S., 2014. Ontology for systems development. Int. J. Softw. Eng. Knowl. Eng. 24, 329–345. https://doi.org/10.1142/S0218194014500132

Hallberg, N., Pilemalm, S., Timpka, T., 2012. Quality driven requirements engineering for development of crisis management systems. Int. J. Inf. Syst. Cris. Response Manag. 4, 35–52. https://doi.org/10.4018/jiscrm.2012040103

Haskins, C., 2014. A Journey Through The Systems Landscape. INSIGHT 17, 63–64. https://doi.org/10.1002/inst.201417263a

Haskins, C., 2011. 4.6.1 A historical perspective of MBSE with a view to the future. INCOSE Int. Symp. 21, 493–509. https://doi.org/10.1002/j.2334-5837.2011.tb01220.x

Haskins, C., Dahl, H.J., 2013. Getting students hooked on systems engineering! INCOSE Int. Symp. 23, 852–860. https://doi.org/10.1002/j.2334-5837.2013.tb03058.x

Haskins, C., Ruud, K.S., 2018. Systems Engineering: Making People Talk!, in: Disciplinary Convergence in Systems Engineering Research. Springer International Publishing, Cham, pp. 1081–1093. https://doi.org/10.1007/978-3-319-62217-0_75

Hatchuel, A., Weil, B., Le Masson, P., 2013. Towards an ontology of design: lessons from C–K design theory and Forcing. Res. Eng. Des. 24, 147–163. https://doi.org/10.1007/s00163-012-0144-y

Hazman, M., Beltagy, S.R. El, Rafea, A., 2009. Ontology learning from domain specific web documents. Int. J. Metadata, Semant. Ontol. 4, 24. https://doi.org/10.1504/IJMSO.2009.026251

He, Y., Zhang, J., Yue, L., Li, Z., Tang, L., 2014. Based on ontology methodology to model and evaluate System of Systems (SoS), in: 2014 9th International Conference on System of Systems Engineering (SOSE). IEEE, pp. 101–106. https://doi.org/10.1109/SYSOSE.2014.6892471

Henderson-Sellers, B., Gonzalez-Perez, C., McBride, T., Low, G., 2014. An ontology for ISO software engineering standards: 1) Creating the infrastructure. Comput. Stand. Interfaces 36, 563–576. https://doi.org/10.1016/j.csi.2013.11.001

Hennig, C., Viehl, A., Kämpgen, B., Eisenmann, H., 2016. Ontology-Based Design of Space Systems, in: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F. (Eds.), The Semantic Web – ISWC 2016. ISWC 2016. Lecture Notes in Computer Science. Springer, Cham, pp. 308–324. https://doi.org/10.1007/978-3-319-46547-0_29

Hepp, M., 2006. Products and Services Ontologies. Int. J. Semant. Web Inf. Syst. 2, 72–99. https://doi.org/10.4018/jswis.2006010103

Herrera, F., Niaki, S.H.A., Sander, I., 2013. Towards a modelling and design framework for mixed-criticality SoCs and systems-of-systems, in: 2013 Euromicro Conference on Digital System Design. IEEE, pp. 989–996. https://doi.org/10.1109/DSD.2013.112

Herzig, S.J.I., Qamar, A., Reichwein, A., Paredis, C.J.J., 2011. A conceptual framework for consistency management in model-based systems engineering, in: Volume 2: 31st Computers and Information in Engineering Conference, Parts A and B. ASME, pp. 1329–1339. https://doi.org/10.1115/DETC2011-47924

Hildebrandt, C., Torsleff, S., Caesar, B., Fay, A., 2018. Ontology building for cyber-physical systems: a domain expert-centric approach, in: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). IEEE, pp. 1079–1086. https://doi.org/10.1109/COASE.2018.8560465

Honour, E.C., 2004. Understanding the Value of Systems Engineering. INCOSE Int. Symp. 14, 1207–1222. https://doi.org/10.1002/j.2334-5837.2004.tb00567.x

Honour, E.C., Valerdi, R., 2014. Advancing an ontology for systems engineering to allow consistent measurement [WWW Document]. URL https://dspace.mit.edu/handle/1721.1/84432 (accessed 4.30.19).

Hoppe, T., Eisenmann, H., Viehl, A., Bringmann, O., 2017a. Guided systems engineering by profiled ontologies, in: 2017 IEEE International Systems Engineering Symposium (ISSE). IEEE, pp. 1–6. https://doi.org/10.1109/SysEng.2017.8088255

Hoppe, T., Eisenmann, H., Viehl, A., Bringmann, O., 2017b. Shifting from data handling to knowledge engineering in aerospace industry, in: 2017 IEEE International Systems Engineering Symposium (ISSE). IEEE, pp. 1–6. https://doi.org/10.1109/SysEng.2017.8088312

Hourali, M., Montazer, G.A., 2011. A New Approach for Automating the Ontology Learning Process Using Fuzzy Theory and ART Neural Network. J. Converg. Inf. Technol. 6, 24–32. https://doi.org/10.4156/jcit.vol6.issue10.4

Hu, F., Shao, Z., Ruan, T., 2014. Self-Supervised Chinese Ontology Learning from Online Encyclopedias. Sci. World J. 2014, 1–13. https://doi.org/10.1155/2014/848631

Hutchison, N., Wade, J., Luna, S., 2017. The Roles of Systems Engineers Revisited. INCOSE Int. Symp. 27, 200–213. https://doi.org/10.1002/j.2334-5837.2017.00354.x

Idrissi, B.E., Baina, S., Baina, K., 2014. Upgrading the semantics of the relational model for rich OWL 2 ontology learning. J. Theor. Appl. Inf. Technol. 68, 138–148.

INCOSE, 2015a. International Council on Systems Engineering. Systems Engineering Handbook: a Guide for System Life Cycle Processes and Activities, 4th Editio. ed. John Wiley & Sons, Inc., Hoboken, New Jersey, USA.

INCOSE, 2015b. Systems Engineering Handbook Scope, in: Walden, D.D., Roedler, G.J., Forsberg, K.J., Hamelin, R.D., Shortell, T.M. (Eds.), International Council on Systems Engineering. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, pp. 3–4.

INCOSE, 2015c. Systems engineering overview, in: Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D., Shortell, T.M. (Eds.), Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, pp. 5–24.

INCOSE, 2015d. Systems science and systems thinking, in: Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D., Shortell, T.M. (Eds.), Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, p. 17.

INCOSE, 2015e. Generic Life Cycle Stages, in: Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D., Shortell, T.M. (Eds.), Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, p. 25.

INCOSE, 2014. Systems engineering vision 2025 [WWW Document]. URL https://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf (accessed 2.18.20).

ISO/IEC/IEEE International Standard, 2015. Systems and software engineering - System life cycle processes, in: ISO/IEC/IEEE 15288. International Organisation for Standardisation / International Electrotechnical Commissions / Institute of Electrical and Electronics Engineers, Geneva, Switzerland, pp. 1–118.

ISO/PAS International Standard, 2015. Automation systems and integration - object-process methodology, in: ISO/PAS 19450. International Organisation for Standardisation, Geneva, Switzerland, pp. 1–162.

Jacobson, M.J., Kapur, M., So, H.-J., Lee, J., 2011. The ontologies of complexity and learning about complex systems. Instr. Sci. 39, 763–783. https://doi.org/10.1007/s11251-010-9147-0

Jeong, S., Kong, J.-S., Sohn, M., Lee, H.J., 2018. Ontology development for dynamic service composition in cyber-physical systems (CPSs), in: Advances in Intelligent Systems and Computing. pp. 601–610. https://doi.org/10.1007/978-3-319-61542-4_59

Jiang, S., Zhang, J., 2013. Development of an ontology-based semantic retrieval method for construction project risk management, in: ICCREM 2013. American Society of Civil Engineers, Reston, VA, pp. 750–760. https://doi.org/10.1061/9780784413135.070

Jiang, X., Tan, A.-H., 2010. CRCTOL: A semantic-based domain ontology learning system. J. Am. Soc. Inf. Sci. Technol. 61, 150–168. https://doi.org/10.1002/asi.21231

Jirkovsky, V., Obitko, M., Marik, V., 2017. Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration. IEEE Trans. Ind. Informatics 13, 660–667. https://doi.org/10.1109/TII.2016.2596101

Jung, J.J., 2004. An application of collaborative web browsing based on ontology learning from user activities on the web. Comput. Informatics 23, 337–353.

Kaderka, J., Rozek, M., Arballo, J., Wagner, D., Ingham, M., 2018. The behavior, constraint, and scenario (BeCoS) tool: a web-based software application for modeling behaviors and scenarios, in: 2018 AIAA Aerospace Sciences Meeting. American Institute of Aeronautics and Astronautics, Reston, Virginia. https://doi.org/10.2514/6.2018-1216

Keating, C.B., 2005. Research Foundations for System of Systems Engineering, in: 2005 IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp. 2720–2725. https://doi.org/10.1109/ICSMC.2005.1571561

Khan, K.S., Kunz, R., Kleijnen, J., Antes, G., 2003. Five steps to conducting a systematic review. J. R. Soc. Med. 96, 118–21.

Kim, H.M., Fox, M.S., 2002. Towards a Data Model for Quality Management Web Services: An Ontology of Measurement for Enterprise Modeling. Springer Verlag, pp. 230–244. https://doi.org/10.1007/3-540-47961-9_18

Kitchenham, B., Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering, EBSE 2007-001.

Kitchenham, B.A., Budgen, D., Brereton, P., 2015. Evidence-based software engineering and systematic literature reviews, 1st ed. CRC Press.

Kong, J., 2007. Ontology Learning for Chinese Information Organization and Knowledge Discovery in Ethnology and Anthropology. Data Sci. J. 6, S500–S510. https://doi.org/10.2481/dsj.6.S500

Kossiakoff, A., Sweet, W., Seymour, S., Biemer, S., 2011. Systems Engineering and the World of Modern Systems, in: Systems Engineering Principles and Practice. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, p. 560.

Langford, G., Langford, T., 2017. The making of a system of systems: ontology reveals the true nature of emergence, in: 2017 12th System of Systems Engineering Conference (SoSE). IEEE, pp. 1–5. https://doi.org/10.1109/SYSOSE.2017.7994936

Lau, R.Y.K., Dawei Song, Yuefeng Li, Cheung, T.C.H., Jin-Xing Hao, 2009. Toward a Fuzzy Domain Ontology Extraction Method for Adaptive e-Learning. IEEE Trans. Knowl. Data Eng. 21, 800–813. https://doi.org/10.1109/TKDE.2008.137

Lee, C.-H., Na, J.-C., Khoo, C., 2003. Ontology Learning for Medical Digital Libraries, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 302–305. https://doi.org/10.1007/978-3-540-24594-0_29

Lee, C.-S., Wang, M.-H., Yan, Z.-R., Lo, C.-F., Chuang, H.-H., Lin, Y.-C., 2008. Intelligent estimation agent based on CMMI ontology for project planning, in: 2008 IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp. 228–233. https://doi.org/10.1109/ICSMC.2008.4811279

Lee, E.A., 2008. Cyber physical systems: design challenges, in: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). IEEE, pp. 363–369. https://doi.org/10.1109/ISORC.2008.25

Lee, J., Goodwin, R., 2006. Ontology management for large-scale enterprise systems. Electron. Commer. Res. Appl. 5, 2–15. https://doi.org/10.1016/j.elerap.2005.08.003

Lemey, E., Poels, G., 2011. Towards a service system ontology for service science, in: CEUR Workshop Proceedings. pp. 250–264. https://doi.org/10.1007/978-3-642-25535-9_17

Li, Z., Yang, M.C., Ramani, K., 2009. A methodology for engineering ontology acquisition and validation. Artif. Intell. Eng. Des. Anal. Manuf. 23, 37. https://doi.org/10.1017/S0890060409000092

Lin, H.K., Harding, J.A., 2007. A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. Comput. Ind. 58, 428–437. https://doi.org/10.1016/j.compind.2006.09.015

Lisi, F.A., Esposito, F., 2009. On Ontologies as Prior Conceptual Knowledge in Inductive Logic Programming. pp. 3–17. https://doi.org/10.1007/978-3-642-01891-6_1

Líska, M., Návrat, P., 2010. An approach to project planning employing software and systems engineering meta-model represented by an ontology. Comput. Sci. Inf. Syst. 7, 721–736. https://doi.org/10.2298/CSIS091110030L

Liu, H., Liu, Y., Pauwels, P., Guo, H., Gu, M., 2017. Enhanced Explicit Semantic Analysis for Product Model Retrieval in Construction Industry. IEEE Trans. Ind. Informatics 13, 3361–3369. https://doi.org/10.1109/TII.2017.2708727

Liu, K., Chapman, W.W., Savova, G., Chute, C.G., Sioutos, N., Crowley, R.S., 2011. Effectiveness of Lexico-syntactic Pattern Matching for Ontology Enrichment with Clinical Documents. Methods Inf. Med. 50, 397–407. https://doi.org/10.3414/ME10-01-0020

Liu, Kaihong, Hogan, W.R., Crowley, R.S., 2011. Natural Language Processing methods and systems for biomedical ontology learning. J. Biomed. Inform. 44, 163–179. https://doi.org/10.1016/J.JBI.2010.07.006

Liu, W., Weichselbraun, A., Scharl, A., Chang, E., 2005. Semi-Automatic Ontology Extension Using Spreading Activation. J. Univers. Knowl. Manag. 0, 50–58.

Lopez-Lorca, A.A., Beydoun, G., Martínez-Béjar, R., Tootell, H., 2011. Supporting developers in complex systems modelling, in: Proceedings of 2011 Australasian Conference on Information Systems. Sydney, Australia.

Lykourentzou, I., Kalliakmanis, A., Latour, T., Kapetanios, E., Papadaki, K., Djaghloul, Y., Charalabis, I., 2011. Ontology-based operational risk management, in: 2011 IEEE 13th Conference on Commerce and Enterprise Computing. IEEE, pp. 153–160. https://doi.org/10.1109/CEC.2011.18

Lynch, K., Ramsey, R., Ball, G., Schmit, M., Collins, K., 2017. Conceptual design acceleration for cyber-physical systems, in: 2017 Annual IEEE International Systems Conference (SysCon). IEEE, pp. 1–6. https://doi.org/10.1109/SYSCON.2017.7934771

Lynch, K., Ramsey, R., Ball, G., Schmit, M., Collins, K., 2016. Ontology-driven metamodel validation in cyber-physical systems, in: Advances in Intelligent Systems and Computing. pp. 1255–1258. https://doi.org/10.1007/978-3-319-32467-8_109

Ma, M., Liu, L., Lin, Y., Pan, D., Wang, P., 2017. Event description and detection in cyber-physical systems: an ontology-based language and approach, in: 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS). IEEE, pp. 1–8. https://doi.org/10.1109/ICPADS.2017.00012

Madni, A.M., Lin, W., Madni, C.C., 2001. IDEON: an extensible ontology for designing, integrating, and managing collaborative distributed enterprises. Syst. Eng. 4, 35–48. https://doi.org/10.1002/1520-6858(2001)4:1<35::AID-SYS4>3.0.CO;2-F

Madni, A.M., Madni, C.C., Weiwen Lin, 1998. IDEON/IPPD: an ontology for systems engineering process design and management, in: SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218). IEEE, San Diego, CA, USA, pp. 2597–2602. https://doi.org/10.1109/ICSMC.1998.725050

Madni, M., Sievers, M., 2018a. Model-based systems engineering: Motivation, current status, and research opportunities. Syst. Eng. 21, 172–190. https://doi.org/10.1002/sys.21438

Madni, M., Sievers, M., 2018b. Model-based systems engineering: motivation, current status, and needed advances, in: Madni, A.M., Boehm, B., Ghanem, R.G., Erwin, D., Wheaton, M.J. (Eds.), Disciplinary Convergence in Systems Engineering Research. Springer, Cham, Cham, pp. 311–325. https://doi.org/10.1007/978-3-319-62217-0_22

Madni, M., Sievers, M., 2014a. System of systems integration: key considerations and challenges. Syst. Eng. 17, 330–347. https://doi.org/10.1002/sys.21272

Madni, M., Sievers, M., 2014b. Systems Integration: Key Perspectives, Experiences, and Challenges. Syst. Eng. 17, 37–51. https://doi.org/10.1002/sys.21249

Maedche, A., Staab, S., 2001. Ontology Learning for the Semantic Web. IEEE Intell. Syst. 16, 72–79. https://doi.org/10.1109/5254.920602

Mandutianu, S., Moshir, M., Donahue, K., Laboratory, J.P., 2009. Conceptual model for space mission systems design, in: INCOSE International Symposium. pp. 110–131. https://doi.org/10.1002/j.2334-5837.2009.tb00941.x

Manenti, G., Ebrahimiarjestan, M., Yang, L., Yu, M., 2019. Functional Modelling and IDEF0 to Enhance and Support Process Tailoring in Systems Engineering, in: 2019 International Symposium on Systems Engineering (ISSE). IEEE, pp. 1–8. https://doi.org/10.1109/ISSE46696.2019.8984539

Mann, C.J.H., 2003. The description logic handbook – theory, implementation and applications. Kybernetes 32, k.2003.06732iae.006. https://doi.org/10.1108/k.2003.06732iae.006

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D., 2015. The Stanford CoreNLP Natural Language Processing Toolkit 55–60. https://doi.org/10.3115/v1/p14-5010

Martin, A., Emmenegger, S., Hinkelmann, K., Thönssen, B., 2017. A viewpoint-based case-based reasoning approach utilising an enterprise architecture ontology for experience management. Enterp. Inf. Syst. 11, 551–575. https://doi.org/10.1080/17517575.2016.1161239

Martin, J., 2012. Toward a Common Language for Systems Praxis. INSIGHT 15, 48–49. https://doi.org/10.1002/inst.201215248

Martin, J., Chroust, G., Lawson, H.B., Sillitto, H., Singer, M., Bendz, J., Hybertson, D., Martin, R., Singer, J., Takaku, T., 2013. Towards a Common Language for Systems Praxis, in: INCOSE International Symposium. Wiley, Philadelphia, PA, pp. 739–754. https://doi.org/10.1002/j.2334-5837.2013.tb03051.x

Mason, G.L., 2005. Connectivity as a basis for a systems modelling ontology. Syst. Res. Behav. Sci. 22, 69–80. https://doi.org/10.1002/sres.613

Matsuoka, T., 2014. Overview of system reliability analyses for PSA, in: Progress of Nuclear Safety for Symbiosis and Sustainability. Springer Japan, Tokyo, pp. 83–95. https://doi.org/10.1007/978-4-431-54610-8_10

Mezhuyev, V., 2014. Ontology based development of domain specific languages for systems engineering, in: 2014 International Conference on Computer and Information Sciences (ICCOINS). IEEE, pp. 1–6. https://doi.org/10.1109/ICCOINS.2014.6868825

Miao, H., Sun, J., 2006. The role of ontology in service oriented system development, in: 2006 Second IEEE International Symposium on Service-Oriented System Engineering (SOSE'06). IEEE, pp. 179–182. https://doi.org/10.1109/SOSE.2006.37

Mikroyannidis, A., Theodoulidis, B., 2010. Ontology management and evolution for business intelligence. Int. J. Inf. Manage. 30, 559–566. https://doi.org/10.1016/j.ijinfomgt.2009.10.002

Miller, D., 2017. An ontology for future airspace system architectures, in: 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC). IEEE, pp. 1–8. https://doi.org/10.1109/DASC.2017.8102091

Missikoff, M., Navigli, R., Velardi, P., 2002. Integrated approach to Web ontology learning and engineering. Computer (Long. Beach. Calif). 35, 60–63. https://doi.org/10.1109/MC.2002.1046976

Musen, M.A., 2015. The protégé project. AI Matters 1, 4–12. https://doi.org/10.1145/2757001.2757003

Nardi, J.C., Falbo, R. de A., Almeida, J.P.A., Guizzardi, G., Pires, L.F., van Sinderen, M.J., Guarino, N., Fonseca, C.M., 2015. A commitment-based reference ontology for services. Inf. Syst. 54, 263–288. https://doi.org/10.1016/j.is.2015.01.012

NASA, 2016. Fundamentals of Systems Engineering, in: Hirshorn, S.R. (Ed.), NASA Systems Engineering Handbook. NASA, Washington, DC, p. 3.

Navigli, R., Velardi, P., Gangemi, A., 2003. Ontology learning and its application to automated terminology translation. IEEE Intell. Syst. 18, 22–31. https://doi.org/10.1109/MIS.2003.1179190

Nebot, V., Berlanga, R., 2012. Finding association rules in semantic web data. Knowledge-Based Syst. 25, 51–62. https://doi.org/10.1016/j.knosys.2011.05.009

Nota, G., Aiello, R., Di Gregorio, M.P., 2010. Ontology Based Risk Management, in: Faggini, M., Vinci, C.P. (Eds.), New Economic Windows. Springer, Milano, pp. 235–251. https://doi.org/10.1007/978-88-470-1778-8_14

OBO Technical WG, 2019. Software Ontology (SWO) [WWW Document]. URL http://www.obofoundry.org/ontology/swo.html (accessed 10.22.19).

Oliga, J.C., 1988. Methodological foundations of systems methodologies. Syst. Pract. 1, 87–112. https://doi.org/10.1007/BF01059890

Orellana, D.W., Madni, M., 2014. Human system integration ontology: enhancing model based systems engineering to evaluate human-system performance, in: Procedia Computer Science. Elsevier B.V., pp. 19–25. https://doi.org/10.1016/j.procs.2014.03.003

Ormrod, D., Turnbull, B., O'Sullivan, K., 2015. System of systems cyber effects simulation ontology, in: 2015 Winter Simulation Conference (WSC). IEEE, pp. 2475–2486. https://doi.org/10.1109/WSC.2015.7408358

Page, D., 2008. Systematic literature searching and the bibliographic database haystack. Electron. J. Bus. Res. Methods 6, 171–180.

Pardo-Calvache, C.J., García-Rubio, F.O., Piattini-Velthuis, M., Pino-Correa, F.J., Baldassarre, M.T., 2014. A reference ontology for harmonizing process reference models. Rev. Fac. Ing. 1, 29–42.

Park, J., Cho, W., Rho, S., 2010. Evaluating ontology extraction tools using a comprehensive evaluation framework. Data Knowl. Eng. 69, 1043–1061. https://doi.org/10.1016/j.datak.2010.07.002

Petnga, L., Austin, M., 2016. An ontological framework for knowledge modeling and decision support in cyber-physical systems. Adv. Eng. Informatics 30, 77–94. https://doi.org/10.1016/j.aei.2015.12.003

Petnga, L., Austin, M., 2013. Ontologies of time and time-based reasoning for MBSE of cyber-physical systems, in: Procedia Computer Science. pp. 403–412. https://doi.org/10.1016/j.procs.2013.01.042

Petnga, L., Austin, M.A., 2015. Spatial ontologies and models for safety-critical cyber-physical systems, in: 2015 International Conference on Complex Systems Engineering (ICCSE). IEEE, pp. 1–6. https://doi.org/10.1109/ComplexSys.2015.7385992

Pfaff, M., Krcmar, H., 2018. A web-based system architecture for ontology-based data integration in the domain of IT benchmarking. Enterp. Inf. Syst. 12, 236–258. https://doi.org/10.1080/17517575.2017.1329552

Prasad, N., Jain, S., Gupta, S., 2019. Electrical Components of Maglev Systems: Emerging Trends. Urban Rail Transit 5, 67–79. https://doi.org/10.1007/s40864-019-0104-1

Quan, T.T., Hui, S.C., Fong, A.C.M., 2006. Automatic Fuzzy Ontology Generation for Semantic Help-Desk Support. IEEE Trans. Ind. Informatics 2, 155–164. https://doi.org/10.1109/TII.2006.873363

Rani, Monika, Dhar, A.K., Vyas, O.P., 2017. Semi-automatic terminology ontology learning based on topic modeling. Eng. Appl. Artif. Intell. 63, 108–125. https://doi.org/10.1016/j.engappai.2017.05.006

Rani, M., Dhar, A.K., Vyas, O.P., 2017. Semi-automatic terminology ontology learning based on topic modeling. Eng. Appl. Artif. Intell. 63, 108–125. https://doi.org/10.1016/j.engappai.2017.05.006

Raskin, R., 2006. Development of ontologies for earth system science. Spec. Pap. 397 Geoinformatics Data to Knowl. 397, 195–199. https://doi.org/10.1130/2006.2397(14)

Rauzy, A.B., Haskins, C., 2019. Foundations for model-based systems engineering and model-based safety assessment. Syst. Eng. 22, 146–155. https://doi.org/10.1002/sys.21469

Remolona, M.F.M., Conway, M.F., Balasubramanian, S., Fan, L., Feng, Z., Gu, T., Kim, H., Nirantar, P.M., Panda, S., Ranabothu, N.R., Rastogi, N., Venkatasubramanian, V., 2017. Hybrid ontology-learning materials engineering system for pharmaceutical products: Multi-label entity recognition and concept detection. Comput. Chem. Eng. 107, 49–60. https://doi.org/10.1016/j.compchemeng.2017.03.012

Rese, M., Meier, H., Gesing, J., Boßlau, M., 2013. An ontology of business models for industrial product-service systems, in: The Philosopher's Stone for Sustainability. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 191–196. https://doi.org/10.1007/978-3-642-32847-3_32

Ring, J., 2002. Toward an ontology of SE. INSIGHT 5, 19–22. https://doi.org/10.1002/inst.20025119

Ring, J., Troncale, L., 2014. An Unambiguous Language for Systems Process Design and Engineering. Procedia Comput. Sci. 28, 635–642. https://doi.org/10.1016/j.procs.2014.03.077

Roche, C., 2000. Corporate ontologies and concurrent engineering. J. Mater. Process. Technol. 107, 187–193. https://doi.org/10.1016/S0924-0136(00)00713-5

Roldán, M.L., Gonnet, S., Leone, H., 2018. An ontology-based approach for sharing, integrating, and eetrieving architectural knowledge. Electron. Notes Theor. Comput. Sci. 339, 43–62. https://doi.org/10.1016/j.entcs.2018.06.004

Rosa, R.L., Schwartz, G.M., Ruggiero, W.V., Rodriguez, D.Z., 2019. A Knowledge-Based Recommendation System That Includes Sentiment Analysis and Deep Learning. IEEE Trans. Ind. Informatics 15, 2124–2135. https://doi.org/10.1109/TII.2018.2867174

Rousseau, D., Wilby, J., Billingham, J., Blachfellner, S., 2016. A typology for the systems field. Systema 4, 15–47.

Ruiz-Martínez, J.M., Valencia-García, R., Fernández-Breis, J.T., García-Sánchez, F., Martínez-Béjar, R., 2011. Ontology learning from biomedical natural language documents using UMLS. Expert Syst. Appl. 38, 12365–12378. https://doi.org/10.1016/j.eswa.2011.04.016

Ryan, E.T., Jacques, D.R., Colombi, J.M., 2013. An ontological framework for clarifying flexibility-related terminology via literature survey. Syst. Eng. 16, 99–110. https://doi.org/10.1002/sys.21222

Ryan, M.J., Wheatcraft, L.S., 2017. On a cohesive set of requirements engineering terms. Syst. Eng. 20, 118–130. https://doi.org/10.1002/sys.21382

Sales, T.P., Baião, F., Guizzardi, G., Almeida, J.P.A., Guarino, N., Mylopoulos, J., 2018. The common ontology of value and risk, in: Trujillo, J.C., Davis, K.C., Du, X., Li, Z., Ling, T.W., Li, G., Lee, M.L. (Eds.), Conceptual Modeling, 37th International Conference, ER 2018. Springer Nature Switzerland, Xi'an, China, pp. 121–135. https://doi.org/10.1007/978-3-030-00847-5_11

Samhan, A., Odeh, M., Sa, J., Kossmann, M., 2016. OntoSoS.CM: A business process architecture driven and semantically enriched change management framework for systems of systems engineering, in: 2016 IEEE International Symposium on Systems Engineering (ISSE). IEEE, pp. 1–7. https://doi.org/10.1109/SysEng.2016.7753178

Sánchez, D.M., Cavero, J.M., Marcos, E., 2009. The concepts of model in information systems engineering: a proposal for an ontology of models. Knowl. Eng. Rev. 24, 5. https://doi.org/10.1017/S0269888909000150

Santoso, H.A., Haw, S., Abdul-Mehdi, Z.T., 2011. Ontology extraction from relational database: Concept hierarchy as background knowledge. Knowledge-Based Syst. 24, 457–464. https://doi.org/10.1016/j.knosys.2010.11.003

Sarder, M.B., Ferreira, S., 2007. Developing systems engineering ontologies, in: 2007 IEEE International Conference on System of Systems Engineering. IEEE, San Antonio, TX, USA, pp. 1–6. https://doi.org/10.1109/SYSOSE.2007.4304237

Sarder, M.B., Ferreira, S., Rogers, J., Liles, D.H., 2007. A methodology for design ontology modeling, in: F.Kocaoglu, D., R.Anderson, T., U.Daim, T. (Eds.), PICMET '07 - 2007 Portland International Conference on Management of Engineering & Technology. IEEE, pp. 1011–1018. https://doi.org/10.1109/PICMET.2007.4349422

Sartakov, V.A., 2015. Ontological representation of networks for IDS in cyber-physical systems, in: Communications in Computer and Information Science. pp. 421–430. https://doi.org/10.1007/978-3-319-26123-2_40

Scheuermann, A., Leukel, J., 2014. Supply chain management ontology from an ontology engineering perspective. Comput. Ind. 65, 913–923. https://doi.org/10.1016/j.compind.2014.02.009

Schindel, W.D., 1997. The tower of babel:language and meaning in system engineering, in: SAE Technical Papers 973217. SAE International. https://doi.org/10.4271/973217

Schmid, A., 2009. What does emergence in computer simulations? simulation between epistemological and ontological emergence, in: Squazzoni, F. (Ed.), Epistemological Aspects of Computer Simulation in the Social Sciences. EPOS 2006. Lecture Notes in Computer Science. pp. 60–68. https://doi.org/10.1007/978-3-642-01109-2_5

Schmit, M., Briceno, S., Collins, K., Mavris, D., Lynch, K., Ball, G., 2016. Semantic design space refinement for model-based systems engineering, in: 2016 Annual IEEE Systems Conference (SysCon). IEEE, pp. 1–8. https://doi.org/10.1109/SYSCON.2016.7490579

Schneider, T., Hashemi, A., Bennett, M., Brady, M., Casanave, C., Graves, H., Gruninger, M., Guarino, N., Levenchuk, A., Lucier, E., Obrst, L., Ray, S., Sriram, R.D., Vizedom, A., West, M., Whetzel, T., Yim, P., 2012. Ontology for Big Systems: The Ontology Summit 2012 Communiqué. Appl. Ontol. 7, 357–371.

Shani, U., 2017. Can ontologies prevent MBSE models from becoming obsolete?, in: 2017 Annual IEEE International Systems Conference (SysCon). IEEE, pp. 1–8. https://doi.org/10.1109/SYSCON.2017.7934726

Silega, N., Noguera, M., Macias, D., 2016. Ontology-based transformation from CIM to PIM. IEEE Lat. Am. Trans. 14, 4156–4165. https://doi.org/10.1109/TLA.2016.7785947

Sillitto, H., 2011. Sharing Systems Engineering Knowledge through INCOSE: INCOSE as an Ultra-Large-Scale System? INSIGHT 14, 20–22. https://doi.org/10.1002/inst.201114120

Sim, S.K., Duffy, A.H.B., 2003. Towards an ontology of generic engineering design activities. Res. Eng. Des. 14, 200–223. https://doi.org/10.1007/s00163-003-0037-1

Simperl, E., Tempich, C., Vrandečić, D., 2008. A methodology for ontology learning, Frontiers in Artificial Intelligence and Applications.

Singer, J., Sillitto, H., Bendz, J., Chroust, G., Hybertson, D., Lawson, H., Martin, J., Martin, R., Singer, M., Takaku, T., 2012. The Systems Praxis Framework, in: Systems and Science at Crossroads – Sixteenth IFSR Conversation. Linz, Austria.

Studer, R., Benjamins, V.R., Fensel, D., 1998. Knowledge engineering: principles and methods. Data Knowl. Eng. 25, 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

Teslya, N., Ryabchikov, I., 2018. Ontology-driven approach for describing industrial socio-cyberphysical systems' components, in: Ronzhin, A., Shishlakov, V. (Eds.), MATEC Web of Conferences. p. 03027. https://doi.org/10.1051/matecconf/201816103027

Torsleff, S., Hildebrandt, C., Daun, M., Brings, J., Fay, A., 2018. Developing ontologies for the collaboration of cyber-physical systems: requirements and solution approach, in: 2018 4th International Workshop on Emerging Ideas and Trends in the Engineering of Cyber-Physical Systems (EITEC). IEEE, pp. 25–32. https://doi.org/10.1109/EITEC.2018.00009

Tranfield, D., Denyer, D., Smart, P., 2003. Towards a methodology for developing evidence-informed management knowledge by means of systematic review. Br. J. Manag. 14, 207–222. https://doi.org/10.1111/1467-8551.00375

Trappey, A.J.C., Trappey, C. V, Govindarajan, U.H., Jhuang, A.C.C., 2018. Construction and validation of an ontology-based technology function matrix: technology mining of cyber physical system patent portfolios. World Pat. Inf. 55, 19–24. https://doi.org/10.1016/j.wpi.2018.08.001

Triantis, K.P., Collopy, P.D., 2014. A comprehensive basis for systems engineering theory, in: 2014 IEEE International Systems Conference Proceedings. IEEE, pp. 97–102. https://doi.org/10.1109/SysCon.2014.6819242

Trokanas, N., Cecelja, F., 2016. Ontology evaluation for reuse in the domain of process systems engineering. Comput. Chem. Eng. 85, 177–187. https://doi.org/10.1016/j.compchemeng.2015.12.003

Tserng, H.P., Yin, S.Y.L., Dzeng, R.J., Wou, B., Tsai, M.D., Chen, W.Y., 2009. A study of ontology-based risk management framework of construction projects through project life cycle. Autom. Constr. 18, 994–1008. https://doi.org/10.1016/j.autcon.2009.05.005

Tudorache, T., 2019. Protégé wiki [WWW Document]. URL https://protege.stanford.edu/products.php (accessed 3.18.19).

Tudorache, T., Nyulas, C., Noy, N.F., Musen, M.A., 2013. WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the Web. Semant. Web 4, 89–99. https://doi.org/10.3233/SW-2012-0057

Uschold, M., King, M., 1995. Towards a methodology for building ontologies, in: Workshop on Basic Ontological Issues in Knowledge Sharing. Citeseer, pp. 1–13.

van Ruijven, L.C., 2015. Ontology for systems engineering as a base for MBSE, in: INCOSE International Symposium. pp. 250–265. https://doi.org/10.1002/j.2334-5837.2015.00061.x

van Ruijven, L.C., 2013. Ontology for systems engineering, in: Paredis, C.J.J., Bishop, C., Bodner, D. (Eds.), Procedia Computer Science. Elsevier, Atlanta, GA, pp. 383–392. https://doi.org/10.1016/j.procs.2013.01.040

Vanherpen, K., Denil, J., David, I., De Meulenaere, P., Mosterman, P.J., Torngren, M., Qamar, A., Vangheluwe, H., 2016. Ontological reasoning for consistency in the design of cyber-physical systems, in: 2016 1st International Workshop on Cyber-Physical Production Systems (CPPS). IEEE, pp. 1–8. https://doi.org/10.1109/CPPS.2016.7483922

Vargas, I.G., Gottardi, T., Braga, R.T.V., 2016. Approaches for integration in system of systems, in: Proceedings of the 4th International Workshop on Software Engineering for Systems-of-Systems - SESoS '16. ACM Press, New York, New York, USA, pp. 32–38. https://doi.org/10.1145/2897829.2897835

Verma, T., Renu, R., Gaur, D., 2014. Tokenization and Filtering Process in RapidMiner. Int. J. Appl. Inf. Syst. 7, 16–18. https://doi.org/10.5120/ijais14-451139

Villaverde, J., Persson, A., Godoy, D., Amandi, A., 2009. Supporting the discovery and labeling of non-taxonomic relationships in ontology learning. Expert Syst. Appl. 36, 10288–10294. https://doi.org/10.1016/j.eswa.2009.01.048

Wagner, D.A., Bennett, M.B., Karban, R., Rouquette, N., Jenkins, S., Ingham, M., 2012. An ontology for State Analysis: Formalizing the mapping to SysML, in: 2012 IEEE Aerospace Conference. IEEE, pp. 1–16. https://doi.org/10.1109/AERO.2012.6187335

Wan, J., Yin, B., Li, D., Celesti, A., Tao, F., Hua, Q., 2018. An ontology-based resource reconfiguration method for manufacturing cyber-physical systems. IEEE/ASME Trans. Mechatronics 23, 2537–2546. https://doi.org/10.1109/TMECH.2018.2814784

Wand, Y., Weber, R., 1990. An ontological model of an information system. IEEE Trans. Softw. Eng. 16, 1282–1292. https://doi.org/10.1109/32.60316

Ward, D., Pichika, H.V., Rossi, M., Sullivan, B.P., 2018a. Assessment and Tailoring of Technical Processes: A Practitioners Experience, in: Mancin, E., Garro, A., Tirone, L., Fierro, D., Gaudenzi, P., Falcone, A. (Eds.), The 4th INCOSE Italia Conference on Systems Engineering. Rome, Italy, pp. 60–70.

Ward, D., Rossi, M., Sullivan, B.P., Pichika, H.V., 2018b. The metamorphosis of systems engineering through the evolution of today's standards, in: 4th IEEE International Symposium on Systems Engineering, ISSE 2018 - Proceedings. https://doi.org/10.1109/SysEng.2018.8544426

Wasson, C., 2015. Systems, Engineering and Systems Engineering, in: System Engineering Analysis, Design, and Development: Concepts, Principles, and Practices. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, p. 880.

Weichhart, G., Guédria, W., Naudet, Y., 2016. Supporting interoperability in complex adaptive enterprise systems: a domain specific language approach. Data Knowl. Eng. 105, 90–106. https://doi.org/10.1016/j.datak.2016.04.001

Wheeler, F.P., Checkland, P., 2000. Systems thinking, systems practice: includes a 30-year retrospective. J. Oper. Res. Soc. 51, 647. https://doi.org/10.2307/254200

Wiens, V., Lohmann, S., Auer, S., 2018. WebVOWL Editor: Device-Independent Visual Ontology Modeling, in: Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks, {CEUR} Workshop Proceedings. CEUR-WS.org.

Wimalasuriya, D.C., Dou, D., 2010. Ontology-based information extraction: an introduction and a survey of current approaches. J. Inf. Sci. 36, 306–323. https://doi.org/10.1177/0165551509360123

Witherell, P., Krishnamurty, S., Grosse, I.R., 2007. Ontologies for supporting engineering design optimization. J. Comput. Inf. Sci. Eng. 7, 141. https://doi.org/10.1115/1.2720882

Wong, W., Liu, W., Bennamoun, M., 2012. Ontology learning from text. ACM Comput. Surv. 44, 1–36. https://doi.org/10.1145/2333112.2333115

Wouters, C., Dillon, T., Rahayu, W., Chang, E., 2005. Large scale ontology visualisation using ontology extraction. Int. J. Web Grid Serv. 1, 113. https://doi.org/10.1504/IJWGS.2005.007552

Wulandari, I.A., Sensuse, D.I., Krisnadhi, A.A., Akmaliah, I.F., Rahayu, P., 2018. Ontologies for decision support system: the study of focus and techniques, in: 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, pp. 609–614. https://doi.org/10.1109/ICITEED.2018.8534947

Yang, L., Cormican, K., Yu, M., 2020. Ontology Learning for Systems Engineering Body of Knowledge. IEEE Trans. Ind. Informatics 1–9. https://doi.org/10.1109/TII.2020.2990953

Yang, L., Cormican, K., Yu, M., 2019a. Ontology-based systems engineering: A state-of-the-art review. Comput. Ind. 111, 148–171. https://doi.org/10.1016/j.compind.2019.05.003

Yang, L., Cormican, K., Yu, M., 2019b. Learning Systems Engineering Domain Ontologies from Text Documents, in: 2019 International Symposium on Systems Engineering (ISSE). IEEE, pp. 1–7. https://doi.org/10.1109/ISSE46696.2019.8984550

Yang, L., Cormican, K., Yu, M., 2017. Towards a methodology for systems engineering ontology development — An ontology for system life cycle processes, in: 2017 IEEE International Systems Engineering Symposium (ISSE). IEEE, Vienna, Austria, pp. 1–7. https://doi.org/10.1109/SysEng.2017.8088299

Yang, L., Cormican, K., Yu, M., 2016. An Ontology Model for Systems Engineering Derived from ISO/IEC/IEEE 15288: 2015: Systems and Software Engineering-System Life Cycle Processes. Int. J. Information, Control Comput. Sci. 11, 1–7. https://doi.org/10.5281/ZENODO.1339576

Yao, H., Liu, C., Zhang, P., Wang, L., 2017. A feature selection method based on synonym merging in text classification system. Eurasip J. Wirel. Commun. Netw. 2017. https://doi.org/10.1186/s13638-017-0950-z

Zavitsanos, E., Paliouras, G., Vouros, G.A., 2011. Gold Standard Evaluation of Ontology Learning Methods through Ontology Transformation and Alignment. IEEE Trans. Knowl. Data Eng. 23, 1635–1648. https://doi.org/10.1109/TKDE.2010.195

Zhang, L.L., 2014. Product configuration: a review of the state-of-the-art and future research. Int. J. Prod. Res. 52, 6381–6398. https://doi.org/10.1080/00207543.2014.942012

Zhou, L., 2007. Ontology learning: state of the art and open issues. Inf. Technol. Manag. 8, 241–252. https://doi.org/10.1007/s10799-007-0019-5

Zhou, W., Liu, Z.T., Zhao, Y., 2007. Ontology learning by clustering based on fuzzy formal concept analysis. Proc. - Int. Comput. Softw. Appl. Conf. 1, 204–208. https://doi.org/10.1109/COMPSAC.2007.161

Zhu, W., He, H., Wang, Z., 2017. Ontology-based mission modeling and analysis for system of systems, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, pp. 538–544. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.85