THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Automatique et informatique

Présentée et soutenue par :

M. SERGE SONFACK SOUNCHIO

le lundi 12 décembre 2022

Titre :

Modélisation et exploitation des connaissances pour les processus d'expertise collaborative

---

Ecole doctorale :

Systèmes (EDSYS)

Unité de recherche :

Laboratoire de Génie de Productions de l'ENIT (E.N.I.T-L.G.P.)

Directeur(s) de Thèse :

M. LAURENT GENESTE

M. BERNARD KAMSU-FOGUEM

Rapporteurs :

M. GAYO DIALLO, UNIVERSITE DE BORDEAUX

M. VINCENT CHEUTET, INSA LYON

Membre(s) du jury :

M. BENOÎT LEBLANC, INP BORDEAUX, Président

M. BERNARD KAMSU-FOGUEM, ECOLE NATIONALE D'INGENIEUR DE TARBES, Membre

M. LAURENT GENESTE, ECOLE NATIONALE D'INGENIEUR DE TARBES, Membre

MME SANJU MISHARA TIWARI, Universidad Autónoma de Tamaulipas, Membre

SUPERVISORS:
Prof. Laurent GENESTE and Prof. Bernard KAMSU-FOGUEM

UNIVERSITY:
University of Toulouse

INSTITUTE:
Toulouse INP

LOCATION: France

This thesis is dedicated to my ancestors, wife, and the SOUNCHIO's family.

ABSTRACT

Nowadays, Expertise Processes are implemented in many fields, particularly in the industry, to evaluate situations, understand problems, or anticipate risks. These processes can also be used upstream of complex and ill-structured problems to assist in understanding them and thus facilitate decision-making. This approach has become so widespread that it has been supported by a standard (NF X 50-110) and a recommendation guide published in 2011 (FDX 50-046).
The approach is mainly based on hypotheses expressed by one or many domain experts and aims to explore all possible aspects of the problem. Subsequently, these hypotheses are progressively validated or not the different phases of the exploratory process with respect to available knowledge of the problem at hand. Thus, experts can understand what is wrong and make decisions or propose solutions by exploring and validating hypotheses with respect to the knowledge available for a problem.

Although the Expertise Process practices and guidelines are defined as a standard, it lacks automatic or semi-automatic tools to assist the domain experts during the different exploratory phases of the process. In addition, in this quasi-manual state, Expertise Processes lack appropriate mechanisms to evaluate expertise, formalize and manage the knowledge produced, such that it can be understood by humans and computed by machines.

Before proposing solutions for these limitations of the current state of Expertise Processes, a review of fundamental and applied studies in logic, knowledge representation for expertise or experience, and collaborative intelligence was carried out to identify the technological building-blocks of the proposed solutions. An analysis of the NF X 50-100 standard was conducted to understand the insights of Expertise Processes and how it can be formally represented and used as experience feedback. Moreover, a study was conducted on past expertise reports from aircraft accidents to find how they can be represented in a machine-readable, general, and extensible format that is domain independent and shareable among systems.

This thesis presents the following studies as contributions to the field of Expertise Processes.

- It proposes formalized knowledge and methodology for collaborative Expertise Processes using hypotheses. This method is illustrated with a use case taken from the field of problem-solving in manufacturing, in which a manufactured product was rejected by clients. The methodology also describes inference mechanisms compatible with the proposed formal representation.

- It presents a non-monotonic collaborative reasoning based on answer set logic programming and the Dempster Shafer Theory. The proposed integration framework is successfully illustrated using a case of auto-mobile diagnosis.

- It describes an ontology for a semantic representation of expertise reports. This contribution yields a base ontology for accident expertise to answer accidents related questions.

First, these contributions have allowed a formal and systematic execution of Expertise Processes, with a human centric motivation. Secondly, they enhance their possible use for further processing according to essential properties such as traceability, transparency, non-monotonic reasoning, and uncertainty, by considering human doubt and experts' limited knowledge of a problem being analyzed. Finally, they provide a human - and machine-readable semantic representation for the expert reports.

Keywords: Expertise process, Knowledge engineering, Epistemic uncertainty, Experience feedback, Non-monotonic reasoning

RÉSUMÉ

Les démarches d'expertise sont aujourd'hui mises en œuvre dans de nombreux domaines, et plus particulièrement dans le domaine industriel, pour évaluer des situations, comprendre des problèmes ou encore anticiper des risques. Placés en amont des problèmes complexes et mal définis, elles servent à la compréhension de ceux-ci et facilitent ainsi les prises de décisions. Ces démarches sont devenues tellement généralisées qu'elles ont fait l'objet d'une norme (NF X 50-110) et d'un guide de recommandation édité en 2011 (FDX 50-046).
Ces démarches reposent principalement sur la formulation d'hypothèses avec un certain doute par un ou plusieurs experts. Par la suite, ces hypothèses vont progressivement être validées ou invalidées au cours des différentes phases de la démarche par rapport aux connaissances disponibles. Ainsi, les certitudes accordées aux hypothèses vont connaître une évolution au cours des dites phases et permettront d'avoir une certitude sur la compréhension d'un problème en fonction des hypothèses valides.

Bien que cette approche d'étude de problèmes ait fait l'objet d'une norme, elle manque d'outils automatiques ou semi-automatiques pour assister les experts du domaine lors des différentes phases exploratoires des problèmes. De plus, cette approche quasi manuelle manque des mécanismes appropriés pour gérer les connaissances produites de manière à ce qu'elles soient compréhensibles par les humains et manipulables par les machines.
Avant de proposer des solutions à ces limites de l'état actuel des processus d'expertise, une revue des études fondamentales et appliquées en logique, en représentation des connaissances pour l'expertise ou l'expérience, et en intelligence collaborative a été réalisée pour identifier les briques technologiques des solutions proposées. Une analyse de la norme NF X 50-100 a été menée pour comprendre les caractéristiques des Processus d'Expertise et comment ils peuvent être représentés formellement et utilisés comme retour d'expérience. Une étude a été menée sur des rapports d'expertise passés d'accidents d'avion pour trouver comment ils peuvent être représentés dans un format lisible par une machine, général et extensible, indépendant du domaine et partageable entre les systèmes.

Cette thèse apporte les contributions suivantes à la démarche d'expertise :

- Une formalisation des connaissances et une méthodologie de résolution collaborative de problèmes en utilisant des hypothèses. Cette méthode est illustrée par un cas d'étude tiré d'un problème de l'industrie de production, dans lequel un produit fabriqué a été rejeté par des clients. La méthode décrit également des mécanismes d'inférence compatibles avec la représentation formelle proposée.
- Un raisonnement collaboratif non-monotone basé sur la programmation logique par l'ensemble et la théorie d'incertitude utilisant les fonctions de croyance.
- Une représentation sémantique des rapports d'expertise basée sur les ontologies.

Premièrement, ces contributions ont permis une exécution formelle et systématique des Processus d'Expertise, avec une motivation centrée sur l'humain. Ensuite, elles favorisent leur utilisation pour un traitement approprié selon des propriétés essentielles telles que la traçabilité, la transparence, le raisonnement non-monotone et l'incertitude, en tenant compte du doute humain et de la connaissance limitée des experts. Enfin, ils fournissent une représentation sémantique lisible par l'homme et la machine pour les expertise réalisées.

Keywords : Processus d'expertise, Ingénierie des connaissances, Incertitude épistémique, Retour d'expérience, Raisonnement non-monotone

ACKNOWLEDGMENTS

My heartfelt thanks to my two Supervisors Prof. Laurent GENESTE and Prof. Bernard KAMSU FOGUEM, for introducing me to the research world and keeping me on track during these three years of Ph.D. I also thank them for their patience, discussion, and help in all aspects to complete this thesis.

I am grateful to the Laboratoire Genie de Production for providing me with infrastructure and resource needed for my research. I would like to acknowledge the support and encouragement of the ICE working group throughout doctoral program.

A special thanks to Prof. V. VAN REIJSWOUD, Prof. R. HOUE NGOUNA, Prof. JUMA LUNGO, Prof. SANJU TIWARI and Mme ELEANOR MAXWELL for their encouragement and advices during this thesis.

I owe much to my wife, my parents the SOUNCHIO family, the extended DJOUGUE family, my in-laws, the ARAMA family, the LEMOGUE family, the Fossong Ellelem community, and friends: NIAZI, KIPKOECH LANGAT, MOHAMMADAMINI,TIZE MHA for their encouragement, moral support and unceasing prayers.

I am grateful to the TSAFACK family, the DJIDJOU family, the NDONGMO family, and the NOUBOUWO family for their assistance when I arrived in France in 2019.

Finally, all praise and glory be to God, the greatest benefactor, Whose help enabled me to complete this thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

ASP Answer Set Programming

FOL First Order Logic

UML Unified Modeling Language

HEG Hypotheses Exploratory Graph

OCL Object Constraint Language

OMG Object Management Group
MDA Model-Driven Architecture
SHACL SHApe Constraint Language
OWL Web Ontology Language
CBR Case-Based Reasoning
AI Artificial Intelligence
KG Knowledge Graph
SAI Symbolic Artificial Intelligence
DDAI Data Driven Artificial Intelliegence
MOF Meta-model Object Facility
BAEO Base Accident Expertise Ontology
DST Dempster Shafer Theory
KG Knowledge Graph
CG Conceptual Graph
FAIR Findable Accessible Interoperable Reusable
IoT Internet of Things
CPS Cyber-Physical-Systems
NFR Non Functional Requirements
SPSP Systematic Problem-Solving Processes
SWRL Semantic Web Rule Language
DL Description Logic
RDF Resource Description Framework
FOD Frame of Discernment
NAF Negation as Failure
OWA Open World Assumption
CWA Closed-World Assumption
SOE Set of Experience
SOEKS Set of Experience Knowledge Structure
RFBSE Requirement Functional Behavior Structure Evolution
DT Digital Twin
I4.0 Industry 4.0
I5.0 Industry 5.0
CBR Case Based Reasoning

1 INTRODUCTION

This thesis focuses on collaborative expertise processes used by experts to understand complex problems in various fields and to support problem-solving or decision-making. It aims to understand this reasoning approach in order to design a formal model, reasoning methodologies based on it, and the integration of human experience in automated reasoning. The studies addressed by this thesis ambition to facilitate collaborative expertise processes and their reuse. In essence, the models, algorithms, and methodologies proposed in this reflection support human reasoning by providing: uncertainty management, knowledge representation during expertise processes, and intelligent computing mechanisms.

This chapter is presented in four sections. First, the context section characterizes expertise processes and presents their role and importance in organizations. The second section identifies and raises questions about the topic to which this thesis provides responses. The third section presents the thesis outline. The last section highlights scientific contributions from issues that were identified earlier.

1.1 CONTEXT

The perpetual quest for development is one of the driving factors of human technological growth (Dulgheru, 2012). This quest led to the First Industrial Revolution between 1760 and 1850, and since then, industries and society have continuously advanced technologically because of the skills and knowledge acquired by humans over time. This technological progress has been driven by great inventions such as electricity, motor cars, airplanes, or radio signals from the first revolution, to innovations such as connected factories, digital twins, and advanced intelligent systems today. Generally speaking, the primary purpose of these technologically challenging contributions is to improve human living conditions, increase production, and speed up productivity while reducing labor efforts (Moll, 2021).

This technology and digital transformation to alleviate some working efforts in sectors such as health, agriculture, aircraft, manufacturing, or automobiles can make us believe that the digitization and automation of companies or society never encountered any difficulties. Nevertheless, they are sometimes subject to complex and unexpected theoretical or practical challenges that can slow down activities or impede productivity. Moreover, a study conducted in Germany highlight that 40% of challenges faced by companies in digitization are those they have never met before (Longard, Schiborr, and Metternich, 2022). These unforeseen impediments can harm humans, cause material and financial losses. Fortunately, some of these problems may be analyzed and resolved using careful methodologies with Systematic Problem-Solving Processes (SPSP) and tools such as Ishikawa, 5 Whys based mostly on discernible symptoms (Meister et al., 2018). Their causes are even easier to search when the goals are clearly stated, and they occur in wholly understood and stable environments. However, this is not always the case because most real-life problems are highly challenging due to increasingly complex environments, no clear methodology to solve them, and vagueness. In addition, they are characterized by multiple and conflicting objectives, uncertainty and are usually evolving (Johnson et al., 2022; Nokes, Schunn, and Chi, 2010). Furthermore, they often have a larger scope, multiple constraints and humans frequently lack complete knowledge to understand them. These types of problems are qualified as Complex Problems (Dörner and Funke, 2017).

A method for solving Complex Problems is by exploring the unknown and complex environment of the problems to gain knowledge and understanding about them in order to make decisions in favor of their resolution. This approach, also used by children unconsciously to solve complex tasks (Ossmy et al., 2022), is referred to as Expertise Processes by the *FN X50-110* French standard. Exploring all plausible understandings of problems aids, on the one hand, in considerably reducing the doubts surrounding them and, on the other hand, collecting and generating knowledge necessary to solve these problems.

Expertise is defined in the French standard *NF X50-110* (Peyrouty, 2010; Pierre Peyrouty, 2011) and the European document *CNS EN 16775 standard* "Expertise activities - General requirements for expertise services" as an activity that objectively provides for each question, an answer, explanations or recommendations using professional judgment, proofs and knowledge. These documents also set guidelines for Expertise Processes in order to improve transparency in the processes, communication, and information sharing among experts during expertise and traceability of Expertise processes.

Authors of (Farrington-Darby and Wilson, 2006) specify that expertise activities, in general, are carried out methodically by experts who are people trained in a specific field and capable of solving complex problems, based on what they have learned or acquired by experience (Bromme, Rambow, and Nückles, 2001). These authors draw attention to the fact that expertise has to be carried out by people with high cognitive capabilities in the domain in which the problem has occurred.

A high-level and generic representation of an Expertise Process is presented in the above standard documents as both **incremental** and **exploratory** guided by hypotheses. Incremental because it is a step-wise process and exploratory because, at each step, all plausible hypotheses are expressed and confronted with the available knowledge for a specific goal to be achieved. Figure 1 depicts an example of an Expertise Process structure that starts with three explored goals and continues the exploration with two of these goals.

The incremental and exploratory methodology usually adopted in Expertise Processes is relevant because these are appropriate means to understand a problem, especially in the context of limited knowledge. The concept of "judgment", present in the definition of expertise, is important because, in the presence of cognitive bias or a high level of uncertainty about available knowledge, expertise can lead to incorrect, incomplete, or biased understandings. So, in essence, expertise can be considered as a

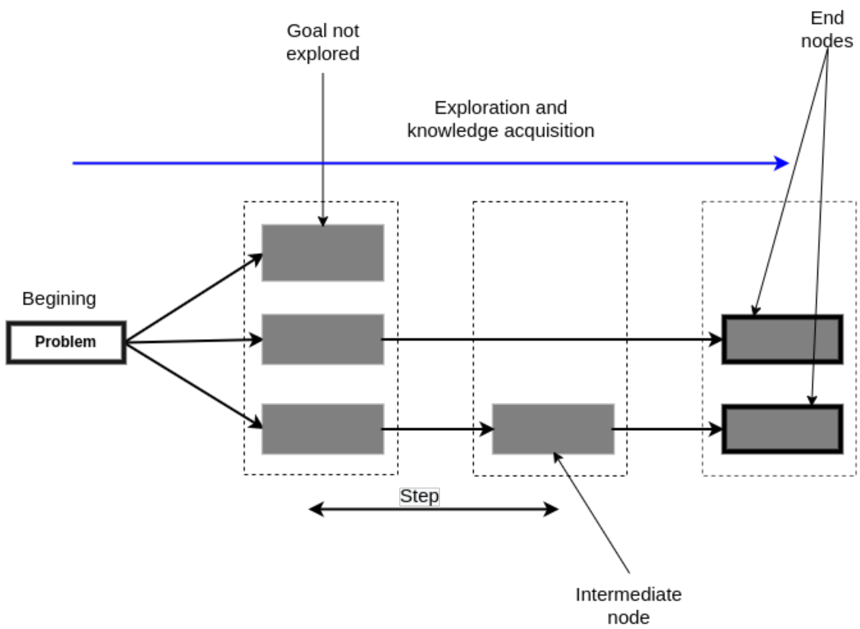


Figure 1: Example of expertise process structure

problem-solving activity given that it encompasses people's efforts using processes or methods to look for a meaningful and feasible interpretation of a problem (Kim and Lim, 2019). Expertise is mainly needed to understand complex problems, which are ill-structured because they have unclear goals and this can lead to multiple solutions or conflicting and competing objectives where there is no clear workflow that leads to a solution. Furthermore, complex problems have a high number of interrelated variables, and there is usually not enough knowledge to clearly understand them (Johnson et al., 2022; Jonassen, 1997; Molnár, Alrababah, and Greiff, 2022).

Some fields where complex problems are found include healthcare, psychology, firefighting, and disaster management. For industry, their continuous upgrade and digitization in order to meet advanced production techniques and improve their efficiency and effectiveness, with intelligent technology operators increases their production system complexity and consequently favoring high failure mode and complex variables' interconnections. These technological factors expose industries to complex problems (Meister et al., 2019; Vartolomei and Avasilcai, 2019).

As a result of the above considerations, expertise can be an essential catalyst for decision-making, taking safety measures, learning, and problem-solving because it facilitates the understanding of complex problems (Patalas-Maliszewska and Kłos, 2019).

Figure 2 describes the predominant position occupied by expertise for subsequent activities such as the design of safety measures or feedback. It also shows that additional knowledge that can come from human perception, learning, or other sources is used to complement human experience (knowledge) during the expertise activity.

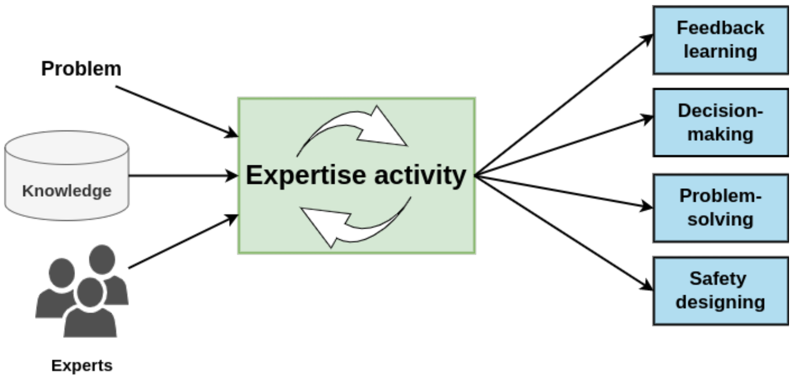


Figure 2: The use of expertise for other activities

Given expertise's central role in unlocking complex problems and its contribution to sustainable and continuous improvement of organizations, it is foreseen as a promising activity in a more and more complex world due to increasing digitization. Moreover, the World Economic Forum alarmed the world on the importance of complex problem-solving skills in its report "The future of jobs report 2018" (Meister et al., 2019).

According to (Navarro and Colbach, 2020), expertise must follow some principles both at the process and the human level in order to achieve quality outcomes. At the process level, these principles are : (1) expertise resources should be *traceable* that is, they should have a well-identified and reliable origin, (2) the method used during the expertise should be *transparent* in the sense that the process is easily understood by humans (3) expertise conclusions should be clear, easy to understand. At the human level, experts have to be guided by the following principles: (1) experts should be independent to avoid bias (2) they should know the domain in which the expertise is being carried out.

Expertise outputs can be reported in various forms. For example, reports can be documents containing knowledge learned during the expertise. It could also be a document containing a recommendation proposed by experts after exploring possible explanations of the problem. The report could also contain the experts' opinions drawn from their experience and what they understood from an expertise activity.
It should be noted that no matter the form in which expertise outcomes are represented, it is usually done using natural language as they are intended for human use.

From a holistic view, expertise has three main phases. The first phase is organizational and consists in describing the problem, setting finances, and selecting experts, the operational phase, brings in the experts. The last phase consists in interpreting or benefiting from the expertise outcomes.

In terms of participation in expertise, experts are not the only actors involved. Expertise activity also involves financial backers who will provide funding for the expertise, project sponsors who call on experts to work on their problem and who are not always the beneficiaries of the expertise, and clients who benefit directly from the knowledge generated by the expertise. Figure 3 illustrates these actors and their roles using an UML use case diagram. Sometimes, a client can inherit the responsibilities of the actors above him and stand as the only interlocutor for experts.

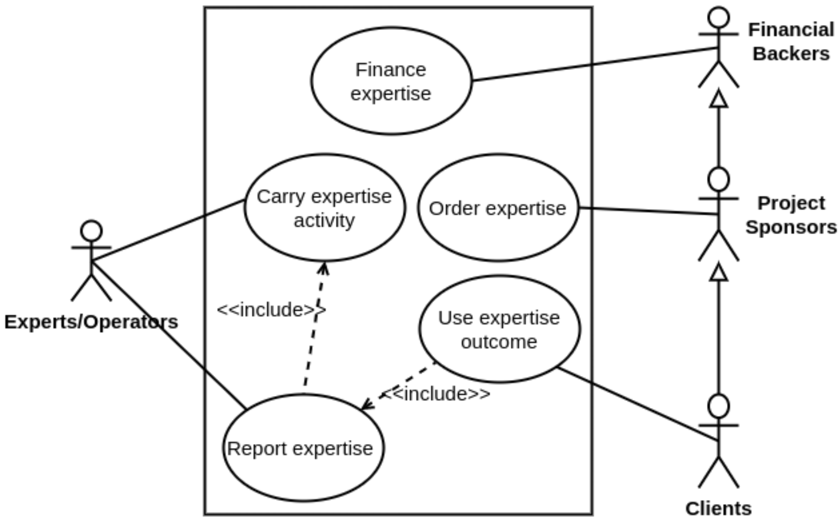


Figure 3: Use case diagram for expertise ecosystem: Actors and their roles

In practical terms, expertise is always done by experts, and in many cases, they come from different fields and backgrounds in which they have acquired extensive and varied experience in their domains. As a result, experts are effective at problem-solving and provided with practical know-how (Shaw and Gaines, 2005).

1.2 PROBLEM AND RESEARCH QUESTIONS

This section has two sub-sections. The first sub-section differentiates expertise as considered in this thesis from concepts such as diagnosis and expertise as knowledge and highlight problems identified for this thesis. The second sub-section focuses on the central question for this thesis that will be answered in subsequent chapters.

1.2.1 *Problem*

From a knowledge point of view, the word "expertise" can be seen as a specific type of tacit knowledge possessed by an expert (Karhu, 2002), which he/she uses to understand problems efficiently. In addition, some authors, such as (Wieten, 2018) emphasize that this knowledge is acquired by interaction or working in a domain. From this standpoint, knowledge obtained from sources such as books or education is not considered expertise. However, considering expertise as a knowledge acquired through interactions or learn in books make it similar to experience, which is considered a specific type of knowledge and is defined as know-how acquired over time by someone, resulting in making the person an expert (Roventa and Spircu, 2009).
These considerations of expertise as knowledge differ entirely from the one this thesis focuses on. This study is interested in expertise as an activity elaborated as a process for understanding problems even if it does not exclude using expertise as knowledge.

Another difference this research would like to highlight is the gap between expertise and diagnosis. Even if expertise and diagnosis may appear similar at first glance, the difference appears when looking at their goals. While expertise focuses on generating knowledge for understanding problems, diagnosis is centered on detecting a systems' failures, disorders, or discrepancies using their observed symptoms (Mariano-Hernández et al., 2021; Marquis, Papini, and Prade, 2014a).
Furthermore, unlike expertise, diagnosis is carried out in a well-defined and stable environment that can be described or modeled accurately. In contrast, expertise is carried out on ill-structured problems or complex problems with unclear goals, high-level of interrelated variables, uncertain environments, and most often, where experts lack complete knowledge of the problems.

To sum up, expertise discussed in this thesis should not be misinterpreted, on the one hand, as someone's skill and knowledge acquired with time by learning or carrying out an activity. On the other hand to diagnoses which generally is intended to identify causes from symptoms in a well-defined environment.

The fact that Expertise Processes rely substantially on human experts makes it challenging, and it will be even more arduous as the world embraces Industrial Revolution 4.0 and 5.0. These technological advancements that are considering changing lives from mass production to mass personalization have an increasing complexity as shown in Figure 4 due to new technological components such as robotics, Artificial Intelligence (AI), 3D printing, augmented reality, cloud computing, edge computing and the desire to facilitate human and smart systems collaboration.

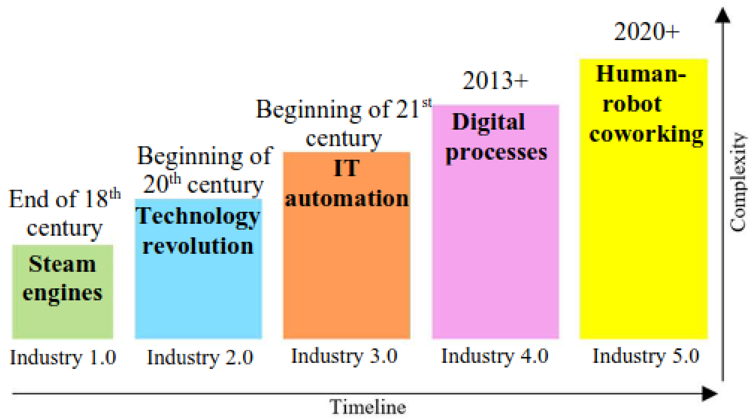


Figure 4: The growth in complexity with upcoming Industrial Revolutions (Taj and Zaman, 2022)

Another challenge is the use of expertise outcomes by both humans and soft intelligent agents that are populating the world. Currently, expertise is reported in natural language, which makes it difficult for smart agents to exploit, whereas the use of machine-readable reports by these agents can allow them to contribute to decision-making and extract hidden knowledge not obvious to humans.

1.2.2 *Research questions*

Having outlined, the differences between expertise as it is considered in this thesis and concepts that may lead to confusion, such as diagnosis or experience knowledge, one central question can be asked.

QUESTION: How can human experts be assisted in collaborative Expertise Processes activities? In other words, is it possible to reduce domain experts' efforts during Expertise Processes? Or is it possible to elaborate digital assistance tools to aid experts during Expertise Processes?
From this central question, three fundamental issues are addressed in this thesis. The first issue is about the Expertise Process formalization and experts' collaboration. The second issue concerns experts uncertainty integration in a reasoning technique and the third concerns the storage of expertise in a computable format.
The challenge is to design the most suitable protocol for human-human and eventually human-machine collaboration under doubt and limited knowledge in order to ease the human problem-solving task.

1.3 THESIS ORGANIZATION

To respond to the question asked in the previous section, the thesis is organized as follows:
Firstly, chapter 2 the state-of-the-art, presents background knowledge such as techniques for knowledge representation, reasoning, uncertainty management, collaborative intelligence, and existing studies addressing topics similar to Expertise Processes such as experience. This chapter is followed by chapter 3, which introduces a framework that formalizes experts' knowledge and methodology for resolving problems collectively using hypotheses. The framework is based on Hypothesis Theory extended with doubt management using possibility theory. It uses a systematic reasoning process over hypotheses exploratory graph to derive conclusions.
Chapter 4, describes a human-machine reasoning procedure for Expertise Process modeling which has the ability to assist experts in their tasks. The human-centered mechanism proposed in this chapter combines experts' beliefs with a declarative knowledge presentation and retractable reasoning based on logic programming coupled with uncertainty management based on belief functions. Fundamentally, the approach gives means to reason by evaluating the logic programming models' beliefs using experts' evidence distributions, thus automating the knowledge-intensive load of the Expertise Process.
Finally, chapter 5 focuses on accident expertise. It provides a semantic schema and reasoning mechanisms to capture and structure accident expertise reports so that humans and machines understand and reason about them.

Figure 5 shows the organization of the thesis on the topic of Expertise Process activity, starting with a reasoning approach, then extraction and formalization of expertise knowledge, and ending with an ontological representation. These chapters (Chapter 3, Chapter 4, Chapter 5) rely on the background knowledge such as Logic,

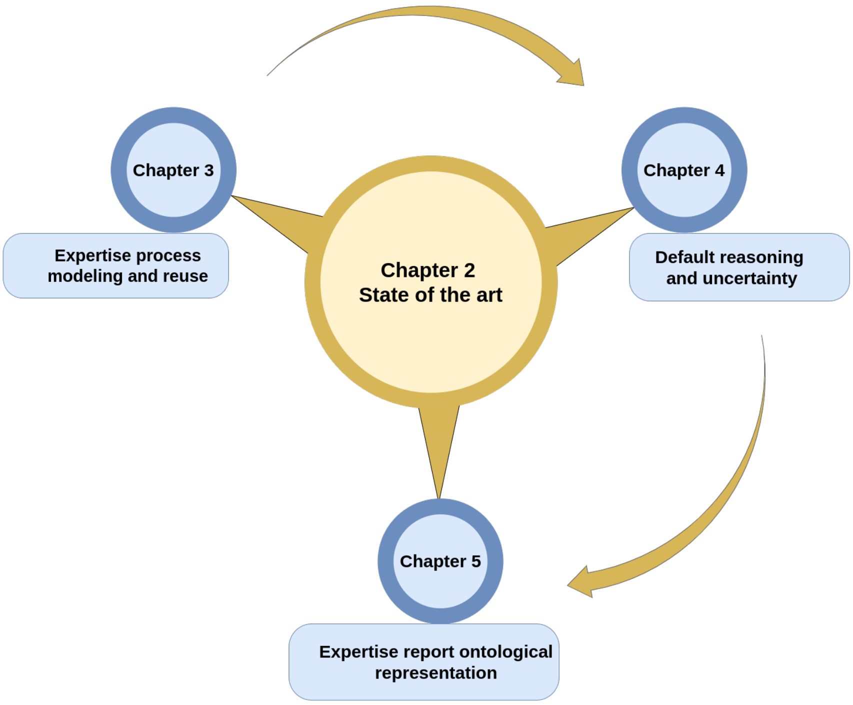


Figure 5: Manuscript organization

Logic Programming, Semantic Web and Ontology, and Experience representation presented in Chapter 2.

1.4 CONTRIBUTIONS

This section exposes scientific contributions published or under review in journals and conferences. They cover issues raised in the thesis.

1.4.1 *Journal publication*

- Sonfack Sounchio, Serge and Geneste, Laurent and Kamsu Foguem, Bernard, Combining expert-based beliefs and answer sets. Applied Intelligence, pp. 1–12, Springer, 2022.
  Published 11 May 2022, https://doi.org/10.1007/s10489-022-03669-z

1.4.2 *Conference publication*

- Sonfack Sounchio, Serge and Geneste, Laurent and Kamsu Foguem, Bernard. Hybridation de l'Answer Set Programming et de la théorie de Dempster Shafer. Journées Francophones d'Ingénierie des Connaissances (IC) Plate-Forme Intelligence Artificielle (PFIA'21), Juin 2021, Bordeaux, France. pp 98-104 https://hal-emse.ccsd.cnrs.fr/emse-03260636

- Sonfack Sounchio, Serge and Geneste, Laurent and Kamsu Foguem, Bernard, Modeling and sharing knowledge in expertise processes. International Conference on Enterprise Systems and Application (I-ESA 2020), novembre 2020, Tarbes, France.

1.4.3 *Journal publication under review*

- Sonfack Sounchio, Serge and Geneste, Laurent and Kamsu Foguem, Bernard, A hypotheses-driven framework for human-machine expertise process. Submitted in **Cognitive Systems Research**.
- Sonfack Sounchio, Serge and Geneste, Laurent and Kamsu Foguem, Bernard, A base ontology for accident expertise. Submitted in **Engineering Application of Artificial Intelligence**.
- Sonfack Sounchio, Serge and Geneste, Laurent and Kamsu Foguem, Bernard, A multilayer graph model of the expertise process knowledge representation. Submitted in **Advance Engineering Informatics**.

2

THE STATE OF THE ART

INTRODUCTION

This chapter elaborates on studies that are close or can contribute to answering the questions asked in the Introduction 1. To surmount, the formalization of expertise processes, concepts of collaborative intelligence, and graph knowledge representation are addressed. Two main uncertainty management concepts are used to deal with experts' doubts. For the issue of making expertise outcomes intelligible and computable by machines, knowledge graphs and ontology will be explored in this chapter.

To cover the above mentions, this chapter is outlined as follows: Firstly, we describe the primary techniques used in designing knowledge for intelligent applications. Secondly, significant theories of uncertainty management are presented. Finally, the fundamentals of collaborative intelligence are handled to complete relevant research fields surrounding the topic of this thesis.

2.1 KNOWLEDGE REPRESENTATION

AI is a sub-field of computer sciences that aims to study and design intelligent agents so that they can mimic human intelligence, such as learning, understanding, producing natural language, and solving problems. AI has two main streams that are Symbolic Artificial Intelligence (SAI) and Data Driven Artificial Intelliegence (DDAI).
The main objective of the symbolic approach is to design knowledge through formal representation and reasoning mechanisms on these representations for intelligent computing. In contrast, data-driven approach is based on knowledge extraction from large data sets.
Used by computer agents to carry out tasks in a human-like manner, knowledge is classified by *Micheal Polanyi* in two main categories. On the one hand, explicit knowledge, which is objective and rational, is easier to acquire, formalize, share and reuse. On the other hand, tacit or implicit knowledge with its cognitive dimension (mental model and beliefs) is challenging to capture and formalize because it is generally personal and sometimes unconscious, relies on social or deep interactions, and can be gained during activities or experiences (Arnett, Wittmann, and Hansen, 2021; Astorga-Vargas et al., 2017; Ikujiro Nonaka, 1995).

This section will focus on SAI instead of DDAI because problem-solving in general and Expertise Processes, in particular, are not carried out enough, which means it will be challenging to think of data learning on them. On the other hand, cases are poorly documented since they do not describe the process but rather the plain text conclusion or are not shared for confidence purposes. Because of this shortage of data for Expertise Processes, it will not be meaningful to explore DDAI, which are methods that rely on massive data.

Another essential aspect to consider for the type of knowledge we wish to design is interpretability. As presented in the previous chapter 1, humans are at the center of expertise.
The following sections will focus on sub-fields of SAI. It will present the logic and graph-based AI for knowledge representation.

2.1.1 *Logical-based knowledge representation*

Using logic to formalize the world is one of the oldest approaches used in AI for knowledge representation and reasoning. It consists in deriving hidden knowledge from logic formulas using well-defined reasoning mechanisms. This vision was proposed by researchers such as McCarthy and Newell in 1968 and 1982, respectively, with the thesis that logic should be used to represent and analyze knowledge (Nebel, 2001) for intelligent reasoning. Accordingly, logic can represent real-world artifacts such as belief, time, action, or planning (Delgrande and Schaub, 2000). As a result, multiple logic and reasoning formalisms have been elaborated to achieve different representations of the world and simulate specific human reasoning behaviors. Commonly, any logic language is described with a syntax based on an alphabet from which sentences are formed, semantics to give meanings of sentences, and inference rules to derive new conclusions. Generally, the set of sentences representing a world is called a knowledge base (Russell, 2021) and is used by inference methods to derive new knowledge. These formalisms can be classified based on their reasoning mechanism as either monotonic or non-monotonic logical reasoning.

2.1.1.1 *Monotonic reasoning approach*

Monotonic reasoning is a safe mode of inferring knowledge hidden in an explicit representation, such that the derived knowledge is also inferred from any consistent increase of the initial knowledge. In other words, the conclusions of these inference methods are true and never retracted once the premises are true.
This form of reasoning is found in most logic languages under classical logic, such as propositional logic or First Order Logic (First Order Logic (FOL)). These logics are defined by syntax, semantics, and inference techniques.
For the case of FOL, its syntax extends the propositional logic syntax with quantifiers, functional symbols, and predicate symbols. An interpretation over a non-empty domain defines its semantics.
Even though this class of logic is used to solve some types of problems, its reasoning strategy poorly manages uncertainty and inconsistency in knowledge representation; that is why an alternative known as non-monotonic was developed to deal with plausible symbolic reasoning.

2.1.1.2 *Non-monotonic reasoning approach*

Non-monotonic reasoning is one of the main characteristics of commonsense reasoning, which consists in deriving from incomplete or inconsistent knowledge, a conclusion that could be retracted with the arrival of new knowledge that conflicts with existing knowledge (Nebel, Rich, and Swartout, 1992). This plausible reasoning

mechanism is based on general patterns such as "*normally, A holds*", "*assume A by default*" or "*in the absence of information to the contrary, assume A*" . These patterns cannot be formalized with classical monotonic logics because of their limitations in handling reasoning with a lack of knowledge and computation.
This way of processing knowledge and drawing conclusions, which may later not be valid as one has precise knowledge about a problem, is also known as defeasible reasoning, which is mostly used when humans have insufficient knowledge about a problem (Brewka, 1991; Reiter, 1988).
Applied in domains where humans excel, such as diagnostic reasoning, natural language processing (Frankish, 2005), non-monotonic reasoning can be achieved on the one hand with various logics such as circumscription, auto-epistemic logic, default logic, and modal non-monotonic logics, and on the other hand with logic programming language computations such as stable model semantics.

2.1.1.3 *Logic programming*

Programming languages are suitable means for humans to share the representation of a world with machines and mechanisms to compute solutions for them (Kowalski, 1974). There is a need to define a specified language for human-machine communication to achieve this purpose, and two main approaches exist. The imperative approach expresses how to solve the problem by providing operational instructions to a language system. In contrast, declarative languages focus on the problem and goal by explicitly describing the problem and goal without processing details. This approach is used by logic programming, based on logic languages consisting of logical symbols with fixed semantics, and functions, predicate symbols that are human-defined with human-dependable semantics. In other words, logic programs are made up of sentences written with symbolic logic (Genesereth and Chaudhri, 2020). Logic programming derives solutions to problems described from well-defined mechanisms called interpretation, which can be procedural or declarative. These interpretations compute the encoded description of the issues being solved and guarantee to find the solutions if they exist.
Some logic programming languages such as Answer Set Programming (ASP) can simulate common-sense reasoning through stable model semantics interpretation.

2.1.2 *Graph-based knowledge representation*

The term Knowledge Graph (KG) was coined by Google and can be defined as the description of real-world entities and their inter-relations, organized in a graph structure (Aggarwal, 2021). This graph structure is user-friendly, offers simple illustrations of complex phenomena, with formal and computable representation.
KG has gained popularity in both academia and industries with applications in fields such as question answering systems, recommendation systems, semantic search, and conversational AI (Hur, Janjua, and Ahmed, 2021). Used for supporting advanced reasoning systems, they assist in sharing knowledge, discovering hidden concepts and learning patterns.
The three primary techniques for constructing KGs:

- the manual method based entirely on domain experts and knowledge engineers,
- the semi-automated approach from which KGs are partially obtained from knowledge bases or raw data using knowledge extraction techniques,
- the automated method relies entirely on machine mining algorithms and natural language processing.

In terms of representation, there are various graph structures to represent knowledge. The most used graph-based structures are property graphs, labeled graphs, or named graphs. Each representation has an expressivity and can be implemented with a specific tool or technology.
The sections below present the most widely used implementations of knowledge graphs and techniques.

2.1.2.1 *Ontologies*

An *Ontology* can be defined as a formal and explicit artifact used to surrogate a domain knowledge (Fionda and Pirrò, 2019; Gruber, 1993; Zhong et al., 2015).
Ontology is one of the building blocks of the semantic web stack, which is guided by the World Wide Web Consortium (W3C). This stack was designed and is constantly updated to fit the new vision of the current Web, which consists in building a machine-readable and human-understandable Web, from which agents can reason or make decisions (Ławrynowicz, 2020). Figure 6 describes the other components of the

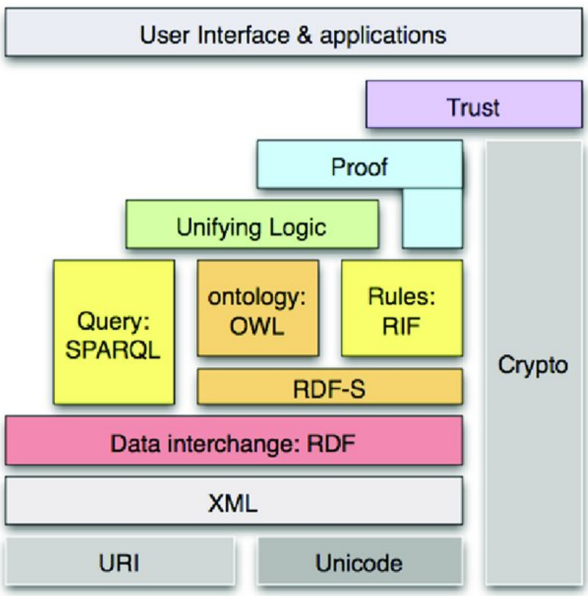


Figure 6: The semantic web stack (Rittgen, 2007)

stack, such as the unified resource identifier (URI), Unicode, XML, data interchange Resource Description Format (RDF), RDF schema, rule interchange format, proof trust, and encryption.
Ontological knowledge is made up of two components which are the *terminological* box denoted by TBox and the *assertions* box denoted by ABox.

The TBox defines concepts and operators, such as equivalence or subsumption, to define new concepts. The ABox defines objects' memberships to concepts in the TBox and relations that exist between them.
Ontology can be formalized with various Description Logic (DL)s, that offer different levels of expressivity and reasoning from the knowledge base. These different DLs match various web ontology OWL languages and reasoners (Chen, Jia, and Xiang, 2020).

Ontologies can be classified into three major groups as shown in Figure 7. This classification is based on how they map or abstract domain concepts and the relationship between these concepts.
From top to bottom, we have, in the first place, top-level ontologies, also called upper-level ontologies, that describe primitive concepts such as entities, process, time, and objects. These upper-level ontologies are made to be extended for specific domains and stand as a base for interoperability. Some examples of these ontologies are the unified foundational ontology (UFO) (Guizzardi et al., 2021), the basic formal ontology (BFO), the general formal ontology (GFO), the DOLCE (a Descriptive Ontology for Linguistic and Cognitive Engineering), Cyc and the suggested upper merge ontology (SUMO) (Mascardi, Cordì, and Rosso, 2007).
The class in the middle corresponds to the domain and task ontologies. This class of ontologies contains concepts that belong to a particular field and guarantee their reuse in these fields.
Finally, application ontologies correspond to a more specific domain than the previous class. Furthermore, while domain ontologies capture domain knowledge with a formal

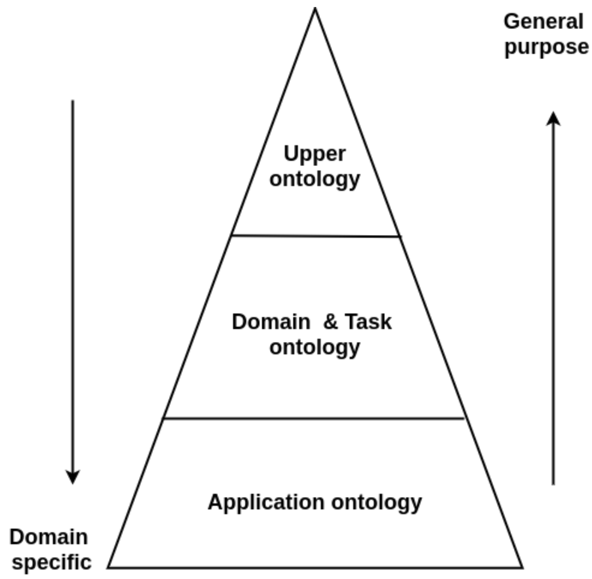


Figure 7: Ontology hierarchy

and consensual vocabulary (Jean, Pierra, and Ait-Ameur, 2007), application ontologies describe components of a domain application and provide semantics to map their integration (Guergour, Driouche, and Boufaïda, 2006; Naubourg et al., 2011).

2.1.2.2 *Conceptual graphs*

Conceptual graphs are a family of graph-based knowledge representation that clearly distinguishes the *ontology knowledge* and the *asserted knowledge* (Croitoru and Compatangelo, 2004). Despite the difficulty of constructing them without prior knowledge, Conceptual Graph (CG)s have good data modeling capacities, are grounded on first-order logic, and have rich knowledge manipulations with standard graphs operations (Faci, Lesot, and Laudy, 2021). Furthermore, CGs are highly expressive and straightforward to use.

Formally a conceptual graph $G = (C, R, E, l)$ is defined over an ontology knowledge or vocabulary $V = (T_C, T_R, I, \sigma)$, where (Aubert, Baget, and Chein, 2006; Schreiber, 2008):

- T_C is the set of concept types, ordered by $\leq$ and has the greatest element $\top$;
- T_R is the set of relation types, ordered by $\leq$ and partitioned into subsets $T_R^1, \ldots, T_R^k$ of relation types arity $1, \ldots, k$;
- I is the set of individual markers, with $*$ as a generic marker;
- σ is a mapping, which assigns a signature to every relation $r \in T_R^j$, $\sigma(r) \in (T_C)^d$;
- (C, R, E) is a bipartite multigraph
  C: is the set of concept nodes
  R: is the set of relation nodes
  E: is the set of edges
- l: is a labeling function of node and edges of (C, R, E), such that:
  - $c \in C$ is labeled by $l(c) = (type(c), marker(c))$,
    $type(c) \in T_c$ and $marker(c) \in I \cup \{*\}$
  - $r \in R$ is labeled by $l(r) \in T_R$

Projection or labeled graph homomorphism is a reasoning mechanism offered by the conceptual graph representation, which defines a generalization/specialization relation over a given graph. In fact, this graph homomorphism determines if a subgraph of a conceptual graph called *query graph* can be deduced from a knowledge base called *factual graph*. In other words, if G_q and G_f are two conceptual graphs over the same vocabulary: There is a projection from $G_q \rightarrow G_f$ if the information represented by G_q can be deduced by the one represented by G_f.

In general, graphs are ideal tools to organize and manage knowledge in a way close to human thinking. Furthermore, they have excellent interoperability potential in addition to their visual and human-understandable aspect of describing the world using graphs.
Graphs can be classified from users' license perspective (proprietary or open), domain (domain-specific and non-domain specific), or type of knowledge represented (commonsense or not) and have storage supports such as Resource Description Framework (RDF) databases, relational databases, or graph databases (Tian et al., 2022).

2.2 EXPERIENCE MANAGEMENT

Experience can be considered as a specific understanding acquired from previous activities of problem-solving (Ruiz, Foguem, and Grabot, 2014; Sun, 2004).
When solving problems or carrying out activities in an enterprise, experiences can be captured from the data transformation or humans, as shown in Figure 8. Humans acquire experience by solving problems from their activities, through errors, or by other colleagues known as experts. This lifetime acquisition of competence while practicing will allow a novice in a field to grow up to an expert level, that is, someone with much experience (Kolodner, 1983). From this author, human experience is encoded at the *episodic memory*, and its magnitude differentiates novices from experts.
The management of experience knowledge (acquire, store, reuse) in a computer-operable form will be an asset for enterprises because they will be able to reuse it if the person leaves the enterprise or is promoted to a different position for his/her quality work done (Song, Jiang, and Liu, 2016). However, the tacit nature of human

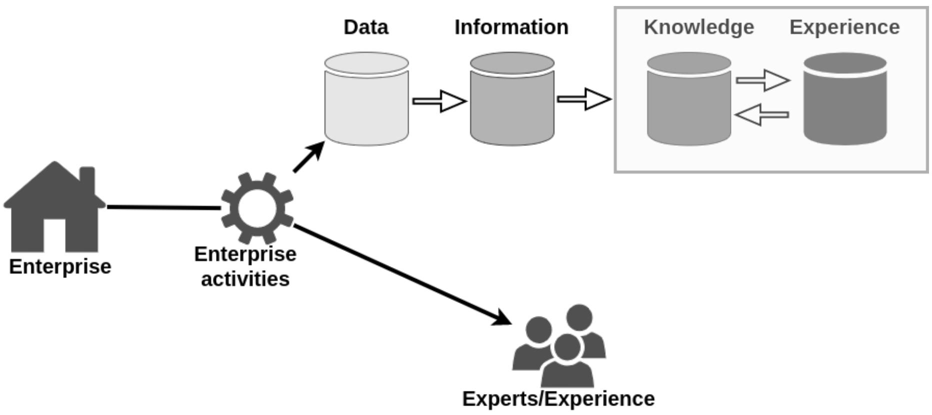


Figure 8: Sources of experience.

experience makes it challenging to acquire and formalize.
In the following paragraphs, attempts of experience representation will be presented.

In the field of agri-food, experts' knowledge gained through experience is highly required for making decisions or carrying out actions. The study presented in (Buche et al., 2019) proposed a system to recommend technical actions for food quality maintenance. This work uses experts' knowledge extracted through interviews and questionnaires. These authors carried out both individual and collective interviews to reduce errors. After that, they use mind maps as an intermediary structure for their experts' knowledge construction. The final representation includes domain and core ontology, a concept graph, and rules in the field of agri-food. Implemented with the concept graph tool called CoGui [1], the constructed knowledge can be queried to know which action to carry for a specific case.
Figure 9 is an example of experts' knowledge presented by (Buche et al., 2019).

Another relevant representation of human expertise is the Set of Experience Knowledge Structure (SOEKS) or Set of Experience (SOE) for short, which is a smart knowledge structure for collecting, storing, improving, and reusing experience of intelligent

1 https://www.lirmm.fr/cogui/

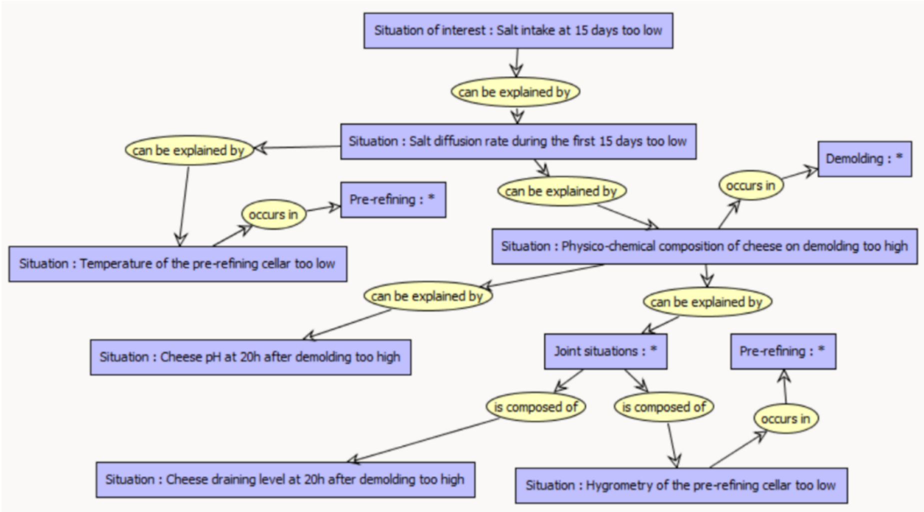


Figure 9: Example of experts' knowledge by (Buche et al., 2019)

decision-making (Shafiq et al., 2017). SOEKS is characterized by: Variables (V) to define components of its environment, functions (F) for relationships between dependent variables and a set of input variables, constraints (C) used for setting limits of feasible solutions, and rules (R) to condition relationships among variables. Rules are presented in the form of "IF-THEN-ELSE" (Shafiq et al., 2015).

As shown in Figure 10, groups of SOEKS are used to store and maintain experi-

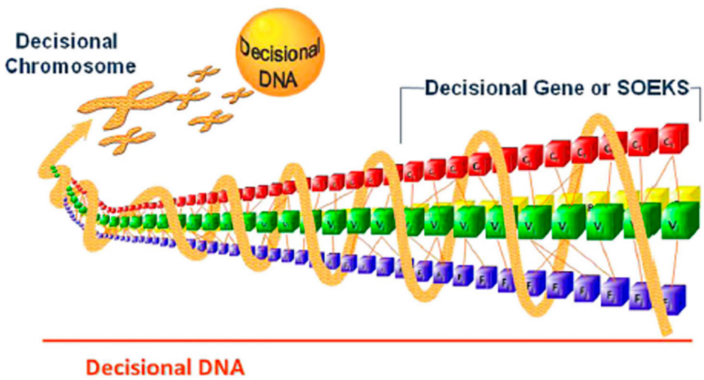


Figure 10: Decision DNA structure by (Waris, Sanin, and Szczerbicki, 2017)

ence and knowledge within an organization. Each group of SOEKS is treated as a chromosome, and a set of chromosomes forms a Decision DNA (DDNA) and transfer knowledge as natural DNA (Shafiq et al., 2014; Waris, Sanin, and Szczerbicki, 2017).

In the domain of manufacturing, (Qin, Wang, and Johnson, 2017) proposed Requirement Functional Behavior Structure Evolution (RFBSE) as a knowledge representation model for acquiring both explicit and tacit knowledge during a design process.
RFBSE is based on these five fundamental elements:

- requirements from market trends or customers,
- functions to describe designed artifacts functionalities,

- behaviors for performing functions,
- structures for design syntheses,
- evolutions for tracking designs and reasoning process changes.

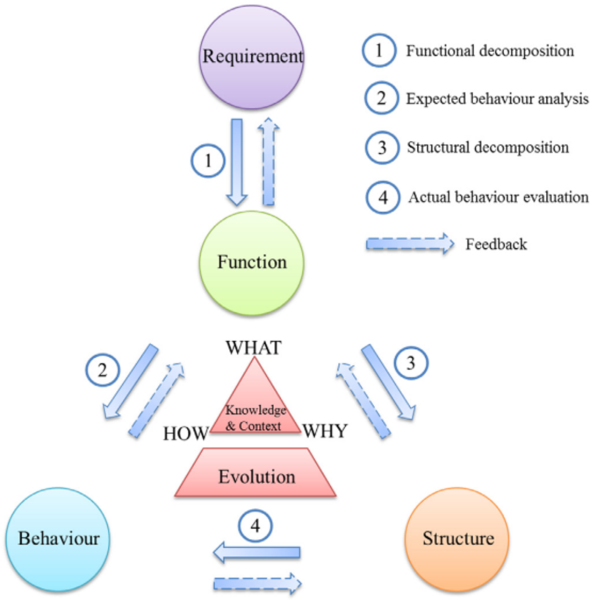


Figure 11: RFBSE knowledge representation scheme (Qin, Wang, and Johnson, 2017)

This approach, as depicted in Figure 11, captures and integrates engineers' know-how, know-what, and know-why during a complete design process that can be later used for decision-making, problem-solving, and designing.

A study carried out by (Sun and Finnie, 2004) proposed a logic-based representation of experience named *experience-based reasoning* (EBR) system. This method uses a multi-inference engine based on eight inference mechanisms to cover various reasoning paradigms humans apply in their activities. They argue the fact that humans use specific reasoning paradigms from their social experiences. These inference rules are fundamentally (1) g-modus ponens, (2) g-rule of trick, (3) g-modus tollens, and (4) g-abduction (Sun and Finnie, 2003) which, according to these authors, cover other aspects of experience reasoning not possible with case-based reasoning (CBR). Table 1 presents these eight inference rules, in which the abduction trick (AT) stands as the "dual" of abduction and can be used for exclusion. The inverses of some rules are motivated by the existence of inverse in logic and serve as common-sense reasoning.

To the best of our knowledge, no related work on knowledge representation intends to represent the knowledge and reasoning process using hypotheses and considering human doubt. Graphical representations map the knowledge well but are generally monotonic and do not describe the resolution steps. Logic representations offer appropriate reasoning mechanisms but do not offer a human visual understanding either. Some experience knowledge representations could capture parts of implicit knowledge, but just as logic representations lack appropriate graphical view and

MP	MT	MPT	A	MTT	AT	IMP	IMPT
P $\frac{P \to Q}{Q}$	$\neg Q$ $\frac{P \to Q}{\neg P}$	P $\frac{P \to Q}{\neg Q}$	Q $\frac{P \to Q}{P}$	$\neg Q$ $\frac{P \to Q}{P}$	Q $\frac{P \to Q}{\neg P}$	$\neg P$ $\frac{P \to Q}{\neg P}$	$\neg Q$ $\frac{P \to Q}{Q}$

Table 1: Experience-based reasoning: Eight inference rules
Modus Ponens: MP, Modus Tollens: MT, MP with Trick: MPT,
Abduction: A, MT with Trick: MTT, Abduction with Trick: AT,
Inverse MP: IMP, Inverse MPT: IMPT

non-monotonicity. However, the ideal representation we would like to achieve has to capture the explicit and implicit knowledge produced during expertise using hypotheses (Salas, Rosen, and DiazGranados, 2010). It also has to materialize the reasoning steps in a human-understandable way.

2.3 COLLABORATIVE INTELLIGENCE

There is no unanimous definition for collaborative intelligence. However, it can be defined as a joint work between two or more agents forming a group to achieve a common goal with sometimes a shared responsibility for the result (Koch and Oulasvirta, 2018). The main objective of this form of intelligence is to ease the task of achieving a common purpose for the group of participants, and this can be done with an organization of tasks so that both parties arrive at a better result and knowledge acquisition (Cheng et al., 2015). Figure 12 depicts a high-level abstraction of collaborative intelligence, showing that agents that collaborate to achieve the group's goal can be different from one another.
According to (Yang, Li, and Jiang, 2021), collaborative intelligence generally has to do

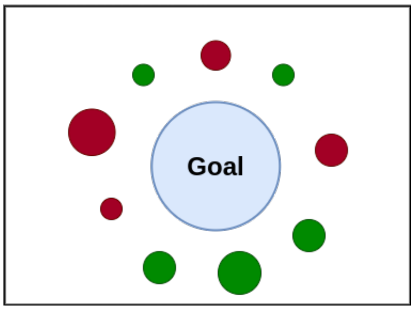


Figure 12: Simple representation of collaborative intelligence

with a diverse group of individual intelligence who uses well-defined mechanisms such as information sharing to achieve their goal. However, when it comes to problem-solving in AI fields, the ability to increase each party's capacity to solve a problem has to be considered (Akata et al., 2020). This idea of capturing more knowledge with collaboration is also supported by (Giannoulis, Kondylakis, and Marakakis, 2019) with a domain expert, as in the case of medical expertise (Doumbouya et al., 2018). In addition, collaborative problem-solving, unlike individual problem-solving, helps to divide the workload among team members and to apply a variety of knowledge, experiences, and events permitting interpersonal stimulation, which can lead to more

creativity, and higher quality solutions (Economic Co-operation and (OECD), 2017). Sometimes the terminology *hybrid* or *collaborative intelligence* is used when the entities involved are humans and machines (Dellermann et al., 2019).
Collaborative intelligence techniques are sometimes used to designate hybrid, collective, or swarm intelligence systems, whereas there are clear distinctions among these forms of intelligence, even though they all involve a group working towards a common goal. On the other hand, it is used to qualify all forms of group intelligence. This last definition encompasses four types of collaboration when considering humans as a particular entity in the collaboration. These groups are illustrated in Table 2. From

	Human	**Machine**
Human	Human-human collaboration	Human-machine collaboration under human control
Machine	Human-machine collaboration under machine control	Machine-machine collaboration

Table 2: Different types of collaborations that can exist within humans and machines

these collaborations involving humans, two variants can be perceived: (1) the *human-machine collaboration under human control* also known as *machine in the loop (MITL)* gives the last decision to human. (2) the *human-machine collaboration under machine control* also known as *human in the loop (HITL)* gives the last decision to the machine.

2.3.1 *Human-machine collaboration*

Over time, humans have developed specific problem-solving skills, such as hypothetical reasoning, intuition, creativity, and induction. Nevertheless, these approaches are less efficient when it comes to reasoning tasks that use algorithms and require a considerable semantic memory (Grigsby, 2018), which on the contrary, are tasks that machines are good at. In addition, human intelligence is adaptable and can achieve acts such as understanding, perceiving, responding to sensory inputs, synthesizing, and summarizing information (Stone et al., 2016), which is still very difficult to have within machine systems. Furthermore, humans can learn new things quickly, adapt and reason independently and develop a gregarious attitude and dynamism (Pupkov, 2019). These important observations grasp our attention to the power of a collaboration between human experts and machine systems to jointly solve problems using hypothetical reasoning methodology.

2.3.2 *Means of collaboration*

An objective of collaborations is to optimize the knowledge of parties in problem-solving. This can be achieved from the cognitive science point of view in the following ways:

1. Collaboration based on *conversational grounding*: according to (Baker et al., 1999), a good collaboration is possible if collaborating entities are able to create a shared knowledge base, beliefs, or assumptions surrounding their goal. The main challenge for this approach is the difficulty of grounding human language in a suitable representation of the real world that can be processed by machines (Chai et al., 2016).

2. Collaboration based on *theory of mind*: this is when collaboration relies on self-understanding and interpretation of others' understandings (Koch and Oulasvirta, 2018). This approach is related to the mental state of the entities involved, which the authors cited, such as beliefs, intentions, knowledge, desires, emotions, or perspectives. One major drawback of this approach is the difficulty for humans to understand machines' mental state or mind (Fussell et al., 2008).

3. Collaboration based on *sub-tasking*: for Epstein (Epstein, 2015), collaborative intelligent systems should partner with humans by sharing sub-tasks that can more efficiently be delegated to persons in order to achieve their goal. This approach of collaborative intelligence was also supported by Pierre Lévy in 1995 when he stated that collaborative intelligence is " a form of universally distributed intelligence, constantly enhanced, coordinated in real-time, and resulting in the effective mobilization of skills " (Suran, Pattanaik, and Draheim, 2020).

2.3.3 *Hybrid Intelligence*

Hybrid Intelligence can be defined in two different forms, first as the framework of building an autonomous system combining a SAI and a DDAI technique, and secondly as the fact of bringing together heterogeneous autonomous or partially autonomous entities to solve a complex problem. In this definition, the group working together should be different and have distinguishable characteristics.

2.3.3.1 *Hybrid intelligence as a combination of techniques.*

This first interpretation of this expression implies the combination of different knowledge representation schemes, decision-making models, and learning strategies to solve a computational task (Abraham and Nath, 2000). These authors specified three essential paradigms that can interact and be integrated rather than merged to build hybrid intelligent systems (HIS):

- Artificial Neural Network, which is adaptive
- Fuzzy logic, with approximate reasoning
- Global optimization algorithm, which is derivative and has free optimization (E. g.genetic algorithms, tabu search)

The integration of the symbolic AI and sub-symbolic model help to take the best of each approach matching the context of needs says (Calegari et al., 2020). This article gives some examples such as neuro-fuzzy, hybrid connectionist-symbolic model, neural-symbolic computing, fuzzy, and connectionism expert systems.

It is also possible to use this approach of hybridization in the area of knowledge representation, working on knowledge-based systems even though it uses symbolic AI. For (Prasad et al., 2012), after mentioning types of knowledge (declarative, procedural, heuristic, meta-knowledge, structural, factual, tacit, priori/prio, posteriori/posteror) and some knowledge representation techniques gave two principles on which hybrid representation should integrate:

1. Representation theory
   It explains what knowledge is to be represented by what formalism.

2. Common semantics for the overall formalism
   It explains in a semantic sound manner the relationship between expressions of different sub-formalisms.

The main objective of this approach is to use the forces or advantages of each technique to solve a common goal, making the hybridization more robust than a single technique.

2.3.3.2 *Hybrid intelligence as a collaboration of heterogeneous agents.*

This second interpretation looks at hybrid intelligence in general as a method for combining complementary intelligence from heterogeneous agents; thus, in particular, a collaboration between *human* and *intelligent system* with the idea to combine their capacities in other to augment each other is also called hybrid intelligence or *collaborative intelligence* (Dellermann et al., 2019). Research on collaborative systems investigates models and algorithms to help develop autonomous systems that can work collaboratively with other systems and with humans (Stone et al., 2016). This approach wasn't envisioned at the beginning of AI in 1955 in what was known as the Dartmouth manifesto, where Claud Shannon, Marvin Minsky, Nathaniel Rochester, and John McCarthy proposed a document where AI termed problem-solving, natural language processing, artificial neural network, complexity theory, machine learning and perception with the purpose that every aspect of learning or intelligence can be simulated by a machine. This document was centered around autonomous intelligence and had no indication that humans and machines could collaborate (Epstein, 2015). This author expresses the fact that the collaborative intelligent system should partner with humans by sharing with them sub-tasks that are more reasonably delegated to persons in other to achieve its goal.
A clear and simple definition of collaborative intelligence was given by Pierre Lévy in 1995 in (Suran, Pattanaik, and Draheim, 2020) as " a form of universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills "
Some important points of the human expert cited by (Abraham and Nath, 2000) are:

- He/she has domain knowledge
- Can reason with uncertainty
- Can adapt to a noisy environment
- Can adapt to a time-varying environment

Collaborative intelligence, when it comes to human-machine interaction, can be seen in two different loops (Dellermann et al., 2019) :

- Artificial intelligence in the loop of human intelligence:
  Which consists in improving human decision or machine prediction assistance by allowing the machine to solve tasks that humans do not want to do themselves.
- Human intelligence in the loop of artificial intelligence:
  Mostly applied to train machine learning models, it relies on humans' assistance to support AI systems in a task it can't solve alone.

It is important to notice that collaborative intelligence is different from purely collective intelligence in the sense that, when it comes to collective intelligence, it involves a group of entities with the same characteristics of the same family acting in an intelligent way: this is also known as swarm intelligence.

2.3.4 *Advantages of Human-machine collaboration*

Collaborative intelligence provides a division of labor that could have been complex to carry out by a single entity. In the case of human-machine collaboration, which consists of delegating some tasks to humans and others to machines, humans have an indisputable contribution to the solution (Epstein, 2015). In fact, human intelligence is so complex that it cannot be completely emulated because it can reason on multiple types of problems, achieve multiple goals at the same time, understand and generate language, perceive and respond to sensory inputs, prove mathematical theorems, play challenging games, synthesize and summarize information, create art and music and even write histories (Stone et al., 2016). Furthermore, it can learn new things very fast, can self-adapt easily, has a gregarious attitude, and creativity (Pupkov, 2019).
Collaborative intelligence can also take advantage of the human ability to handle social and emotional tasks in general, his capacity to derive the implication of AI analysis and translate its information into knowledge. In addition, collaborative intelligence can also benefit from humans' background knowledge and their skills to solve problems (Paschen, Wilson, and Ferreira, 2020).
In other words allowing the collaboration between AI system and human will help systems to overcome some limitations of the machine, like lack of consciousness, lack of emotion, and communication (Mario Raich and Richley, 2019), which are mastered by humans. Merging human and artificial intelligence will yield what these authors call *meta-mind*.

2.4 UNCERTAINTY MANAGEMENT

Uncertainty is defined as the state or condition of being uncertain, that is, being unable to be accurate or not having confidence. It reflects a lack of exact knowledge in providing an information (Parsons and Parsons, 2001; Salicone, 2007). There are two main types of uncertainties. On the one hand there is aleatory uncertainty also known as objective uncertainty based on random experiments and suitable when there is sufficient information. Under this type, there is probability theory. On the other

hand, there is epistemic uncertainty also known subjective and reducible uncertainty appropriate for expressing doubt when there is lack of knowledge of the state of the world.
There are multiple means to model and compute uncertainty; however, some models of uncertainty subsume others. Under this type there are possibility and evidence theories.
This section covers the *possibility theory* and the *evidence theory* because this thesis is human-centred and the context in which its issues mentioned is in need of epistemic/-subjective uncertainty.

2.4.1 *Possibility theory*

Possibility theory was developed as an extension of *probability theory* that was instead designed to manage uncertainty using sets and frequentist distribution, also known as the objective probability. However, frequentist distributions are inappropriate for managing human natural language reasoning under incomplete information. Furthermore, it does not give means to properly express ignorance. In other words, probability theory is not suitable for belief types of uncertainties (Dubois and Prade, 2015).
Possibility theory is a simple mathematical theory that provide means to compute uncertainty from imprecise information or subjective probability. It measures belief with the possibility and necessity dual functions (Alola, 2012), is non-additive as opposed to probability, and has the advantage of providing graded semantics to natural language statements.
Possibility theory is defined as follows:

Let Ω be a state of states of affairs that correspond to the universe of everything that could belong to the state in consideration.
The *possibility distribution function* π that quantifies the degree of possibility is given by:

$$\begin{cases} \pi : \Omega \to L \\ 0 \leq \pi(s) \leq 1 \quad \forall s \in \Omega \\ sup_{s\in\Omega}\{\pi(s)\} = 1 \end{cases}$$

L is a totally ordered scale having 1 as the top and 0 as the bottom and corresponds to the plausible states.
π represents the state of knowledge about the actual state of affairs. If the *possibility distribution function* is normalized, $\exists s \in \Omega$ such that $\pi(s) = 1$

SEMANTICS

- Semantically, the bigger is $\pi(s), s \in \Omega$ the more s is possible and if $\pi(s) = 1$ then, s is totally possible.

- Complete knowledge is considered if one knows the only possible world among those available in the universe. In other words, $\exists s_0, \pi(s_0) = 1$ and $\pi(s) = 0, \forall s \neq s_0 \in \Omega$.

- Complete ignorance is when one thinks all worlds of the universe are possible: $\forall s \in \Omega, \pi(s) = 1$.

If $A \subseteq \Omega$
$\Pi(A) = sup_{s \in A}(\pi(s))$ evaluates to what extent A is consistent with π and corresponds to the *possibility* measure of A also known as *potential possibility*.

$N(A) = inf_{s \notin A}(1 - \pi(s))$
$N(A) = 1 - \Pi(\overline{A})$,
where $\overline{A}$ is the complement of A. It evaluates to what extent A is certainly implied by π and corresponds to the *necessity* measure of A.

$\Pi(\Omega) = N(\Omega) = 1$
$\Pi(\emptyset) = N(\emptyset) = 0$

Maxitivity:

- $\Pi(A \cup B) = max(\Pi(A), \Pi(B))$

Minitivity:

- $N(A \cap B) = min(N(A), N(B))$

2.4.2 *Evidence theory*

Dempster Shafer Theory (DST), also known as belief functions theory, is a generalized approach for reasoning under epistemic uncertainty and the lack of knowledge. Like the possibility theory, it relies on dual functions and subsumes both the probability and possibility theory. This theory offers an essential mechanism for merging pieces of evidence (Denœux, Dubois, and Prade, 2020).
The subsections below elaborate on the fundamentals and the combination rule of the DST.

2.4.2.1 *Fundamentals*

The theory is set out in the following ideas of *mass function*, *belief function* and *plausibility* (Liu and Yager, 2008; Reineking, 2014; Shafer, 1976, 1986; Yager and Liu, 2008), which when given a question of interest (for instance a problem to solve), assigns a mass of evidence to all considered hypothetical answers. This theory is formalized as follows:
Let $\Theta = \{\theta_1, \ldots, \theta_n\}$ be a finite set of possible answers or hypotheses to a question called a frame of discernment (FOD), which is a finite set of mutually exclusive elements in the domain.
2^Θ is the power set from this FOD, which corresponds to the set of all subsets of Θ. Θ can be considered as a set of possibilities, with exactly one of which corresponding to the truth
$2^\Theta = \{A | A \subseteq \Theta\}$,
given a set A, which represents a statement or proposition that the truth lies in A.

A belief function allows basic belief numbers or mass numbers to be assigned to an entire set of points in Θ without further division. This corresponds to assigning a belief value to each hypothesis based on one or more pieces of evidence.

The value of $m(A)$ represents the amount of belief strictly committed to hypothesis A. The entire belief is divided into one or more evidence numbers $m(A)$ and allocated to one or more subsets A.
$m : 2^\Theta \rightarrow [0,1]$ is a mass distribution function used to represent a piece of evidence regarding some variable $A \in 2^\Theta$
$m(\emptyset) = 0$ if m is a normalized mass function that corresponds to the closed-world assumption and
$\sum_{A \subset \Theta} m(A) = 1$
$\forall A \in 2^\Theta$ if $m(A) > 0$, it is called a *focal set* of m.

Given a basic belief number $m(A)$, the belief and plausibility functions over Θ are defined as follows:

- Belief function
  The belief function $Bel()$, also known as lower probability
  $\begin{cases} Bel(A) = \sum_{B \subseteq A} m(B) \\ m(\emptyset) = 0 \end{cases}$
  $Bel(A)$ can be interpreted as one's degree of belief that the truth lies in A.

- Plausibility function
  The plausibility function $Pl()$ of hypothesis A, also called the upper probability, is the amount of belief not strictly committed to the complement of $\bar{A}$
  $\begin{cases} Pl : 2^\Theta \rightarrow [0,1] \\ Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\overline{A}), A \subseteq \Theta \end{cases}$

2.4.2.2 *DS combination rule*

Combining uncertain information can be an effective way of merging multiple agents' uncertainties . This issue was addressed in the DST as a *combination rule*. The DST offers a powerful mechanism for combining different bodies of evidence from the same frame.
Given distinct basic beliefs m_1 and m_2 over a common frame, the combined mass is equal to (Yager and Liu, 2008)

$$m(A) = (m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A \neq \emptyset} m_1(B) m_2(C)}{\sum_{B \cap C = \emptyset} m_1(B) m_2(C)}$$

Consequently, this combination rule takes advantage of the redundancy and complementarity of evidence sources to obtain a more precise distribution.

The following example shows an application of the DST combination rule.
Suppose we have a car to expertise from which all possible explanations belong to the set:
$\Theta = \{A1 = lowfuel, A2 = fuelpumpbad, A3 = batterybad\}$.
Two experts, from their experience in the domain, can express the following distributions of evidence:
Table 3 and 4 below show their distributions:

$m_1(\{A1\})$	$m_1(\{A1, A2\})$	$m_1(\{A3\})$	$m_1(\{\Theta\})$
0.3	0.4	0.2	0.1

Table 3: First expert distribution

$m_2(\{A2\})$	$m_2(\{A2, A3\})$	$m_2(\{A3\})$	$m_2(\{\Theta\})$
0.2	0.3	0.3	0.2

Table 4: Second expert distribution

Considering these two distributions, the mass of evidence associated with $\{A3\}$ is equal to:

$m(\{A3\}) = m_1 \oplus m_2(\{A3\})$

$m_1 \oplus m_2(\{A3\}) = \frac{(0.2*0.3)+(0.2*0.3)+(0.1*0.3)+(0.2*0.2)}{(0.3*0.2)+(0.2*0.2)+(0.3*0.3)+(0.3*0.3)+(0.3*0.4)} = \frac{0.19}{0.4}$

$m(\{A3\}) = m_1 \oplus m_2(\{A3\}) = 0.47$

Under the above distributions, the combined mass of $\{A3\}$ was greater than that of the individual sources.

For this thesis where experts are intended to collaborate during Expertise Processes, this rule is of significant importance because it will be applied to merge experts' beliefs during their collaboration with the same expertise.

2.4.3 *Uncertainty and logic*

This section presents some relevant studies that combine logical reasoning and uncertainty in general and with the DST for uncertainty management in particular.

Some important studies have been carried out to bridge logical reasoning and uncertainty in general and DST in particular. This is the case in (Núnez et al., 2013), whose approach sets an interval of uncertainty to first-order logic formulas, such that De Morgan's law is preserved under uncertain logic. This becomes classical logic when the lower bound of the interval is equal to the upper bound. In terms of semantics, the lower bound of this interval supports the truth of the formula, whereas ignorance is the difference between the upper and lower bounds. Based on this mechanism, a Frame of Discernment (FOD) was built, and appropriate operations were used to combine the support masses for the first-order logical operators (AND, OR).
Similar approaches have been used with logic programming languages such as ASP. An excellent example of this is the possibilistic Answer Set Programming (ASP) (PASP), which combines possibility theory and ASP by setting a certainty value to ASP declarations. For this approach, the certainty of the conclusion after computation is equal to the lowest certainty of the rules used during the computation (Nicolas et al., 2006). This semantic was not satisfactory because models were penalized by minimum certainty, and therefore (Bauters et al., 2012a) proposed a new semantic in which the certainties of rules in an ASP program define a set of constraints over a possibility distribution, such that the certainty of an atom appearing in a model depends on those of its body. Another combination of possibility theory and ASP was given by (Bauters et al., 2012b), which uses Boolean states for truth and necessity measures to evaluate the plausibility of these truths.

CONCLUSION AND REMARKS

This chapter explores relevant concepts related to collaborative Expertise Processes. This exploration includes significant knowledge representation and reasoning mechanisms, collaborative intelligence, uncertainty management techniques and combination of uncertainty and logic. It starts with monotonic and non-monotonic logic-based representations based on well-defined formal languages and reasoning methods. The main difference between the two approaches is that the latter conclusions are retractable.

The second category of representations addressed in this chapter is graph-based representations. The ontological and conceptual representations were considered for their differences in technological stack and reasoning mechanisms. The ontological knowledge representation relies on sub-branches of description logic and can be formalized in RDF, whereas the conceptual representation utilizes graph theory algorithms and homomorphism for reasoning.

After these knowledge representations, collaborative intelligence is addressed and is distinguished from collective, swarm, and hybrid intelligence.

The last category is the management of doubt in knowledge. For this group of approaches, possibility and belief function theory were considered because they overcome most uncertainty management challenges, such as probability and fuzzy logic.

2.5 RESEARCH QUESTIONS

Having gone through these notions, which are limited to the desired representation of Expertise Processes but can be essential for their construction, the central question remains: how can human experts be assisted in collaborative Expertise Process activities? In other words, is it possible to reduce domain experts' efforts in carrying out the Expertise Process by assisting them?

From this central question, three fundamental issues were addressed from different angles. These perspectives are (i) knowledge acquisition, (ii) knowledge representation, and (iii) collaborative reasoning. The questions asked for these three key perspectives are:

QUESTION 1 Since expertise should be transparent, traceable, and involves the collaboration of experts from different technical fields, how can such knowledge be acquired and formalized for future uses?

The challenge of this question is to have a human-readable representation that machines can compute during and after the Expertise Process to derive hidden knowledge while fulfilling the mentioned commitment.

QUESTION 2 How can human experts' beliefs be integrated with automated reasoning approaches suitable for Expertise Processes?

Expertise is a human-centered activity in which experts use their experience in the context of limited knowledge regarding a problem. For this reason, considering their beliefs is essential to reasoning mechanisms that can assist them.

QUESTION 3 Can past expertise reports be formalized in a machine-readable representation in order to ease their understanding and learn from them through systematic reasoning?
This latch's primary goal is to transform past expertise reports writing in natural language, which is difficult to process, to learn by humans or reuse in automated systems.

After elaborating on the relevant studies that stand as components that can assist in formalizing some dimensions of Expertise Processes, three issues were retained for this thesis. First, how can Expertise Processes be formalized to permit human-assisted deep analysis of problems? The second is how can humans and machines collaborate in solving a problem when there is limited knowledge and doubt about it? The third is how machines can gain an understanding of expertise reports.

3 EXPERTISE PROCESS MODELING AND REUSE

3.1 INTRODUCTION

Problem-solving is an important activity in industry, it serves as a way of continuously improving products and processes. Structured approaches such as Expertise Processes are often implemented to allow an in-depth analysis of problems, to search for plausible understanding or explanation, to design corrective measures, to make decisions, and to take preventive actions. Generally speaking, experts involved in problem-understanding put forward different hypotheses based on the elements of knowledge at their disposal, which will be progressively confirmed or refuted at each addition of new knowledge. This work is often the result of a collaboration between experts from different fields who bring and share their vision of the problem and, put forward the hypotheses they consider relevant to explain it.
Even though a normative reference frame on expertise approaches has been defined in the French NF X50-110 standard and the European CNS EN 16775 standard, there are no models or tools to support, share and reuse expertise.
Thus, this chapter introduces the foundations of a formal representation and reasoning mechanism for human experts and machine collaborative Expertise Processes. These foundations support a hypotheses-driven process using experts' doubts and hypothesis theory. The proposed solution will assist experts in three ways: Firstly, it proposes a shared formal representation of their reasoning process based on hypotheses. Secondly, it defines an inference mechanism based on hypotheses in order to draw an explanation from expertise representation. Finally, it reuses formalized processes to infer and learn hidden knowledge.

3.1.1 *Hypotheses-driven problem solving*

This section defines the notion of hypothesis and sets the definition for a reasoning process based on hypotheses.

3.1.1.1 *Hypothesis definition*

Defining the concept of hypotheses is essential to understanding the hypothesis-driven reasoning approach. The ancient Greek philosopher Aristotle defined a *Hypothesis* as "a judgment, affirmative or negative that is merely assumed without being certain; and thus it is a statement that can be used as the basis of an inference only insofar as it is conceded, and so rest, upon homo logia." (Rescher, 1968); Trochim and Donnelly (Trochim and Donnelly, 2001) think it is a specific statement of prediction that describes in real terms (as opposed to theoretical terms) what someone expects will happen in a study or is the cause of a problem (Hurley, 2014). A more recent definition is given by (Ashley, 2007) as a tentative assumption to draw out its normative, logical, or empirical consequences. From these definitions, one can discover that hypotheses are doubtful and predictive thoughts which can be used to analyse problems.

Nevertheless, while reasoning based on hypotheses is well known in the field of science (Kell and Oliver, 2004), it is a challenge to integrate it into systems (Aikins, 1979). In summary, hypothesis can be used to infer knowledge and solve problems, and its use to derive solutions to a given problem can be called hypothesis-driven problem-solving. This suggests that the associated method guides the verification or test of expressed hypotheses in accordance or not with what is known of a problem. Nonetheless, for most real-life problems, for example in the manufacturing sector, when confronted with problems, experts are able to express hypotheses because of their experiences, skills, and high domain knowledge (Ericsson, Hoffman, and Kozbelt, 2018). It is these qualities which make experts rather than non-experts, the most appropriate participants in the process of hypotheses-driven expertise in their field.

3.1.1.2 *Hypothetical reasoning*

Hypothetical reasoning is a method of solving problems by expressing hypotheses and reasoning over them (Hurley, 2014). It is a method used to reason under incomplete knowledge circumstances and try to provide an explanation for the hypotheses (Minutolo, Esposito, and De Pietro, 2016).
This approach of reasoning is, for instance, used by medical practitioners (Hoffman, 2007) and can be resumed in following steps:

1. Collection of context information
   This can be done by simple observations, measurements, or experiments. This information supports any suspicion or warning about a certain case.

2. Hypotheses formulations
   They are proposed based on practitioners' knowledge and beliefs according to questions they ask in order to solve the problem.

3. Reasoning
   Knowledge is computed based on context information or what is known of the domain using an inference mechanism like induction or deduction

4. Hypotheses verification
   After the reasoning step, hypotheses are confirmed or not, and conclusions help to understand the main problem.

3.1.2 *Hypothesis Expertise Process*

This section describes the reasoning process elaborated in this chapter as a protocol of human-machine collaboration.

The reasoning phase of the Hypotheses-driven Expertise Process or Hypotheses' Expertise Process for short is an iterative process, which has as an entry point the problem to be solved, and it involves experts who work collectively and collaboratively with a machine system towards a common goal. Iterations involve experts who work collectively and collaboratively with a machine system for a common goal. The iterative process is described in Figure 13 as follows:

- Step A: Experts ask questions to understand the problem at hand.
- Step B: Experts express hypotheses related to questions of step A.
- Step C: Additional knowledge of the context of the problem being solved is considered.
- Step D: Automatic reasoning is carried out using hypotheses and additional knowledge.

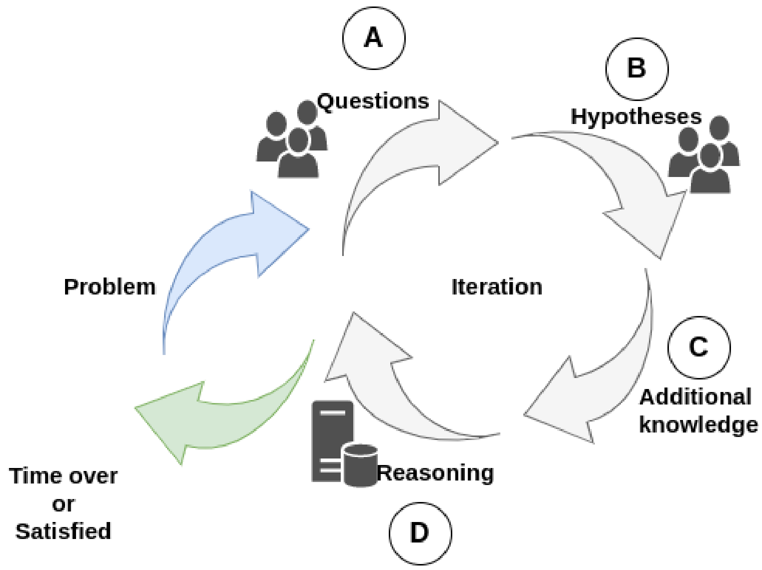


Figure 13: Hypotheses Expertise Reasoning Cycle

The Hypothesis-driven Reasoning Expertise Process yields at the end a Hypotheses Exploratory Graph (HEG) that will be exploited in the next sections for reasoning. In general, each iteration of the Hypothesis-driven Reasoning Expertise Process modifies the HEG at different points: Firstly, in terms of the number of available nodes, since each new Hypothesis introduced will create a new node on the graph. Secondly, in terms of node states, the status of its nodes can change, depending on whether they are confirmed or not due to the available additional knowledge. Figure 13 describes these hypotheses' Expertise Process reasoning cycle, while Figure 14 shows details of the questions phase (*A*) corresponding to steps *1*, *2* and the hypothesis phase (*B*) corresponding to steps *3*, *4*.

Figure 14 depicts how domain experts collaboratively select the questions and hypotheses which will be used for the Expertise Process. These selection processes are greatly inspired by (Doumbouya et al., 2018), with a voting process for aggregating experts' individual preferences into collective preferences, and a threshold mechanism to limit their numbers (Marquis, Papini, and Prade, 2014b).

- In steps one (1) and two (2) a question or multiple questions are selected based on the voting scoring rule. This question selection procedure will allow multiple experts to choose collectively the questions which should be used to understand

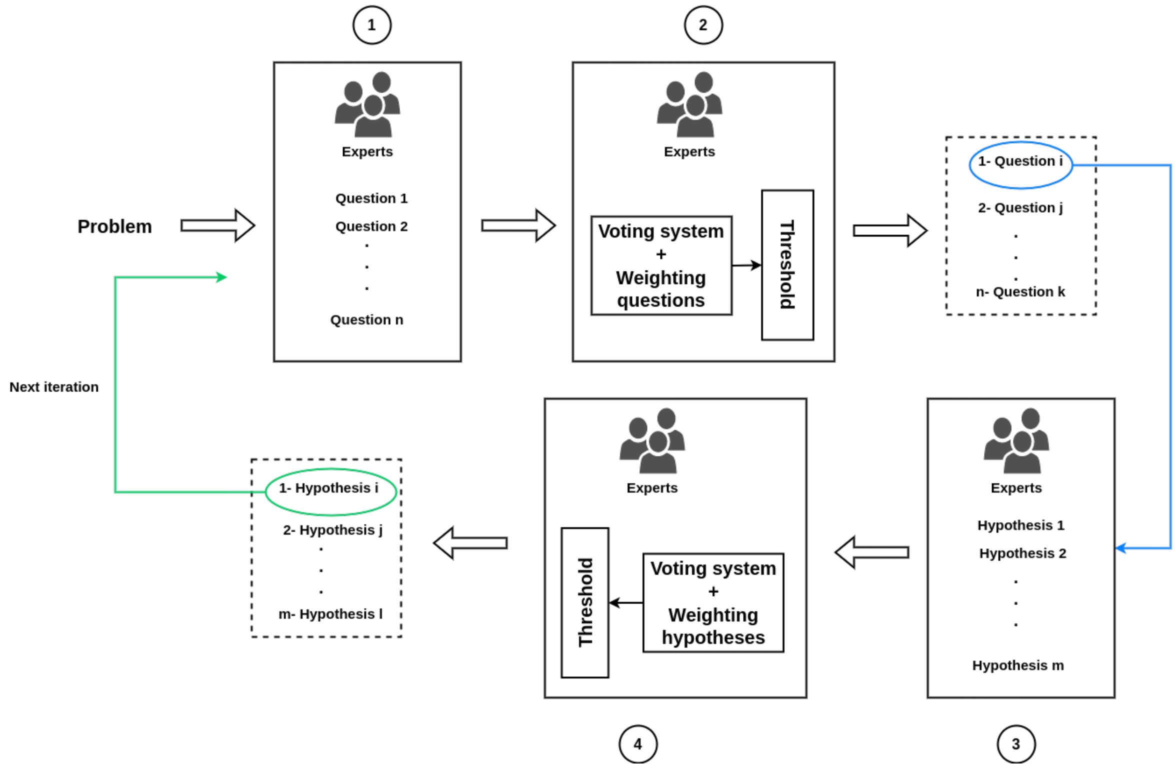


Figure 14: Collaboration of domain experts for questions and hypotheses expressed for the Expertise Process

the problem at hand.
The selection procedure is as follows:

Let $E = \{e_1, \ldots, e_m\}$ be experts working together asking relevant questions.

Let $coef_{e_j} \in]0,1]$ be the level of expert e_j, such that if e_j is more experienced than e_i then, $coef_{e_j} > coef_{e_i}$.
$coef_{e_j}$ forms an ordered set of the experts' levels.

Let $\{q_i^{\epsilon}\}_{1 \leq i \leq n, \epsilon \in \{e_1, \ldots, e_m\}}$ be the questions such that $q_i^{e_j}$ is the question asked by an expert e_j among m experts.

The scoring function used is $score()$ defined as:
$score(q_i^{\epsilon}) = s_j, \epsilon \in \{e_1, \ldots, e_m\}$
where s_j belongs to the scoring rate $s_1 = 1, \ldots, s_n = 0$.

The $QFinal\_score$ of question is given by:

$$QFinal\_score(q_i^{e_j}) = \gamma * coef_{e_j} + \beta * [\frac{\sum_{\epsilon=e_1}^{e_m} score(q_i^{\epsilon}) * coef_{\epsilon}}{\sum_{\epsilon=e_1}^{e_m} coef_{\epsilon}}], \tag{1}$$

$\gamma + \beta = 1$

γ, and β are respectively the coefficient of importance given to experts' level and a question vote, respectively.

Selected questions will be those with highest *QFinal_score*. However, a predefined threshold can be used to filter the number of questions.

For example, if three experts named 1, 2, and 3 with level novice, competent, and expert respectively are to select a question for the problem: *Product KW831 is rejected by customers*, the selection procedure will go as described in Figure 15.

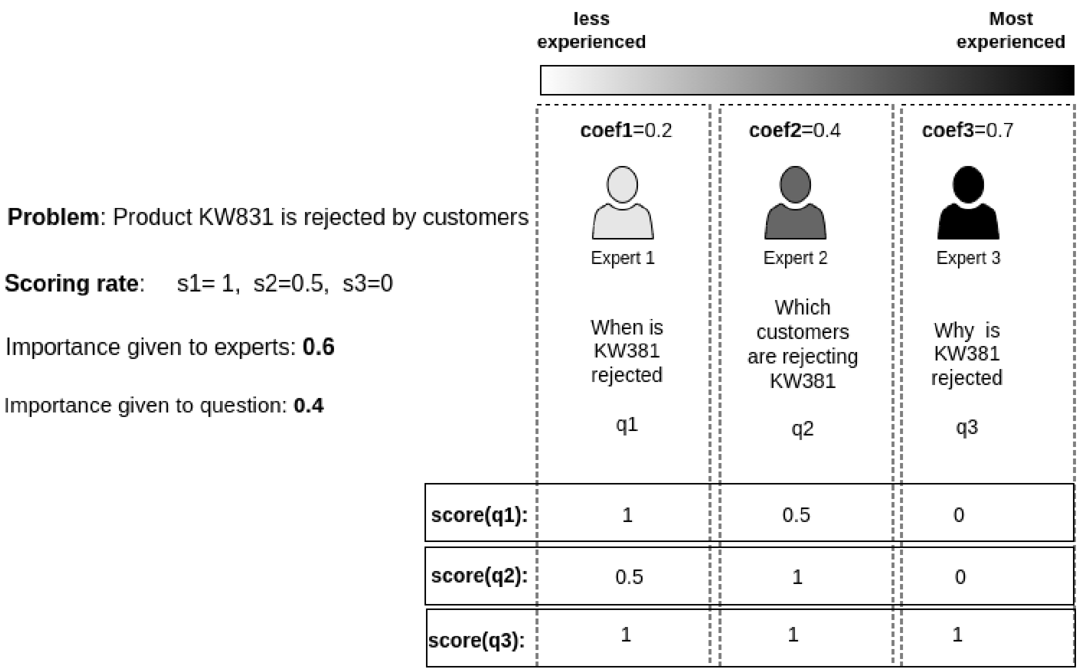


Figure 15: Example of selecting questions

The $QFinal\_score(q1) = 0.6 * 0.2 + 0.4 * \frac{1*0.2+0.5*0.4+0*0.7}{0.2+0.4+0.7} = 0.24$

The $QFinal\_score(q2) = 0.6 * 0.4 + 0.4 * \frac{0.5*0.2+1*0.4+0*0.7}{0.2+0.4+0.7} = 0.39$

The $QFinal\_score(q3) = 0.6 * 0.7 + 0.4 * \frac{1*0.2+1*0.4+1*0.7}{0.2+0.4+0.7} = 0.82$

For this example, question $q3$ expressed by expert 3 is the one to be selected.

In the example above, three levels were assigned to experts' experiences. However, the following levels may be used: *Novice, Beginner, Advanced Beginner, Competent, Proficient, or Expert*. This set of experts' levels must be an ordered set. The list of levels is ordered, and the weights of these levels range between $]0, 1]$.

These levels of experience may be based on experts' years of experience or levels certified by organizations.
In conclusion, selected questions will have a maximum *QFinal_score*.

- In steps three (3) and four (4) a hypothesis or hypotheses are selected with respect to the question at hand. This phase uses the following formula to give weight to each of these hypotheses, similar to the previous formula for selecting a question. In addition, the doubt expressed by an expert on its hypothesis is also counted in the overall score.

$$HFinal\_score(h_i^{e_j}) = \gamma * coef_{e_j} + \alpha * \pi_{h_i^{e_j}} + \beta * [\frac{\sum_{\epsilon=e_1}^{e_m} score_e(h_i^{\epsilon}) * coef_{\epsilon}}{\sum_{\epsilon=e_1}^{e_m} coef_e}], \tag{2}$$

$\alpha + \gamma + \beta = 1$

$coef_{e_j}$ is the level of the expert that expressed $h_i^{e_j}$,
$\pi_{h_i^{e_j}}$ the doubt given to hypothesis $h_i^{e_j}$ by expert e_j.
α, γ, and β are respectively the coefficient of importance given to the doubts, the experts' level, and the scores of experts.

This iterative process (see Figure 13) guarantees a dynamic property to the HEG and consequently Expertise Processes, bringing out the question of when to stop.
The process may be ended in two possible ways:

1. Time as indicator:
   A length of time can be set as the endpoint for the Expertise Process. This will be suitable in situations where the exploration could take more time than available.

2. Satisfaction as indicator:
   This is when the expertise process is stopped because one is satisfied with the solution produced.

In general, an expertise process may be described by the following algorithm 1: The proposed methodology is domain-independent and produces a *hypotheses exploratory graph* (HEG) which is a contextualized knowledge, similar to experience (Bergmann, 2002; Ruiz, Foguem, and Grabot, 2014; Sun, 2004; Sun and Finnie, 2004). The sections that follow describe the collaborative aspect of the methodology, its formalization, and how it can be processed to derive more refined knowledge.

It should be noted that hypothetical reasoning could be misinterpreted as argumentation for the followings: Firstly, both come from human cognition and are also used in scientific reasoning and daily life for solving problems (Dung, 1995). Secondly, both capture some strengths such as uncertainty when reasoning.
However, they are fundamentally different as shown in the following points : (1) Argumentation is an approach for reasoning that uses arguments (premises or reasons and conclusions or claims) with conflicting information, to identify the most acceptable reasoning (Besnard and Hunter, 2008; Morveli-Espinoza et al., 2019). While

Algorithm 1: Hypothesis-driven process algorithm

```
ALGORITHM(Graph, Experts, KBS):
if the stopped time criterion is given then
    set stop time
else
    set stop to satisfactory
end
while stop time not reached or not satisfied do
    Experts ask questions;
    Select questions;
    forall Selected questions do
        Experts express Hypotheses;
        Select hypotheses per questions;
    end
    if additional knowledge then
        add it to KBS;
    end
    Reason on Graph;
end
```

hypothetical reasoning is based on a process of collecting information and asking questions from which hypotheses are derived. (2) In terms of processes: generally, in argumentation, what matters is not the rationality of a statement or its explanation, but the fact that it argues successfully against an attacking argument (Amgoud and Ben-Naim, 2018; Dung, 1995), which is not the case for hypothetical reasoning.

The following sections define two formalizations that will be used to render computable Expertise Processes. The first section elaborates on a logical representation of the hypotheses, knowledge, and reasoning. The second section focuses on integrating linguistic and possibility representations to enhance human doubt computations. These two concepts are paramount for the formal description of the Hypotheses Exploratory Graph (HEG) proposed in this chapter. HEG stands as the central knowledge structure of the proposed Expertise Process mechanism.

3.2 HYPOTHESES REASONING FORMALIZATION

Moving on from the design of the reasoning mechanism, this section continues with the formal representation of the knowledge used in the hypotheses exploratory process and the integration of linguistic doubts.

3.2.1 *Hypothesis logic*

Hypothesis logic, denoted by $\mathcal{H}$, is a bimodal logic that uses the preferential model of non-monotonic reasoning to represent an appropriate state of knowledge based on hypotheses. $\mathcal{H}$ subsumes the Default Logic and uses the L operator that has properties of the modal system $\mathcal{T}$ and $[H]$ that has those of the modal system $\mathcal{K}$ to extend classical predicate logic (Siegel and Schwind, 1993; Siegel et al., 2017, 2020).

- Known formulas
  They are expressed with the L operator: Lf, where f is any formula. Known formulas correspond to *what is known* or *proved/stated*
- Truth
  These are any first-order logic formula not containing a modal operator. Truth formulas are *true without necessarily being known or assumed*
- Hypothesis
  They are expressed with the H operator: Hf, where f is any formula. Hypotheses correspond to *assumptions which can be made*
  The model operator $H = L\neg L\neg$ and is the dual of $[H]$
  Hf means f is a *hypothesis* whereas $[H]f$ means $\neg f$ *is not a hypothesis*

3.2.1.1 *Syntax and axioms*

The language of hypothesis logic similarly to First Order Logic (FOL) consists of the following alphabet.

- Variables: $x, y, \ldots$
- Connectives or logical symbols: $\neg, \wedge, \vee, \rightarrow$
- Quantifiers: $\forall, \exists$
- n place function symbols: F/n
- n place predicate symbols: P/n

The language of hypothesis logic $\mathcal{L}(\mathcal{H})$, has the following definitions and axioms (Schwind and Siegel, 1994; Siegel and Schwind, 1993).

- $\mathcal{L}(\mathcal{H})$ contains the FOL
  In fact:
  - Any formula of FOL belongs to $\mathcal{L}(\mathcal{H})$;
  - $\forall f, g \in \mathcal{L}(\mathcal{H})\ \neg f,\ f \wedge g,\ f \vee g,\ f \rightarrow g,\ Lf,\ [H]f,\ Hf, \in \mathcal{L}(\mathcal{H})$
- If f is a formula, then f, Lf, Hf, $[H]f$, $\neg Lf$, $\neg[H]f$ and $\neg Hf$ are formulas in $\mathcal{L}(\mathcal{H})$
- All rules and axioms of FOL are also rules and axioms in $\mathcal{H}$
- Axioms
  Let f and g be FOL formulas.
  1. If $L(f \rightarrow g)$ then, $Lf \rightarrow Lg$
  2. If f is true then, $L\neg f$ (If a formula is *true*, therefore its *negation* is *known*)
  3. $\forall^x(L(P(x)))$ then, $L(\forall^x P(x))$

4. $[H](f \rightarrow g)$ then, $[H]f \rightarrow [H]g$
5. $\forall^x([H]P(x))$ then, $[H](\forall^x P(x))$
6. Lf then, $[H]f$

- Inferences
  1. If $L(f \rightarrow g)$ and Lf then, Lg
  2. If a formula is *known*, therefore it is *true*
     $Lf \rightarrow f$
  3. If f is a tautology, then f is known, but facts that are merely true are not necessarily known
  4. If hypothesis f is made, then $\neg f$ is not known
  5. $f \wedge g$ is known if and only if f is known and g is known:
     $L(f \wedge g) \iff Lf \wedge Lg$
  6. Making the hypothesis f does not mean knowing f
  7. Not making the hypothesis f does not mean knowing its negation
  8. If f is not hypothesized then , $\neg f$ is known: $\neg Hf \rightarrow L\neg f$
  9. The L operator is not distributive: $L(f \cup g) \neq Lf \cup Lg$

3.2.1.2 *Hypothesis Theory and extension*

A hypothesis theory $\mathcal{HT} = \{F, HY\}$ is defined with a set of formulas F in $\mathcal{H}$ and HY a set of Hypothesis.

Definition 3.1 *Extension*
An extension E of $\mathcal{HT} = (F, HY)$ is the biggest subset of F and hypotheses HY', such that HY' is consistent with F.

This extension can be found with the following recursive equation as extension of default theory (Voorbraak, 1991).
$E = \cup_{i \geq 0} E_i$
$E_0 = F$
for $i > 0$:$E_{i+1} = Th(E_i) \cup \{h : h \in HY \quad and \quad \neg h \notin E\}$

The Th operator is based on the fixed point definition and uses maximal consistent sets. Th is *non-monotonic* because an extension is defined by adding hypotheses, and in some cases, newly added formulas in a knowledge base can prevent some previously admitted hypotheses from belonging to the extension E (Siegel et al., 2020).
Extensions can be classified into two groups:

- An *extension* E is a *stable extension* if it satisfies the *coherence property*:
  $\forall Hf, \neg Hf \in E \rightarrow L\neg f \in E$. In other words, whenever it is forbidden to assume f, $\neg f$ is proven.
- An *extension* E is a *ghost extension* if: $\exists Hf, \neg Hf \in E$ and $L\neg f \notin E$

3.2.2 *Linguistic terms and uncertainty*

To facilitate experts' interaction with the proposed hypotheses-driven expertise process, a linguistic expression was designed and integrated with possibility theory to consider qualitative doubts when exploring problems or making decisions (Lan et al., 2015). For this purpose, the proposed approach uses linguistic term sets as an entry point of uncertainty management.
To have a realistic system, this study uses linguistic terms with the following properties (Xu, 2012).

- Unbalanced terms: Naturally, humans do not uniformly quantify terms.
- Symmetric information: For each term, there is an opposite term of equal magnitude.
- Continuous possibility values: Terms are given possibility values which indicate how certain the person is as (s)he expresses them.

Definition 3.2 ***Possibility Linguistic Term Sets (PsLTS)***
Let S be a linguistic term set.
$S = \{s_t | t = -\tau, \ldots, -1, 0, 1, \ldots, \tau\}$, *where* $\tau > 0$ *is a positive integer and the cardinality of* $|S| = 2\tau + 1$
S is a finite and totally ordered discrete term set. Its elements correspond to the possible values of a linguistic variable.
S supports the following (Xu, 2004):

1. *Ordered set:* $s_i \geq s_j$ *if* $i \geq j$
2. *Negation operator:* $\forall s_i \in S, \exists sj \in S | neg(s_i) = sj$
3. *Max operator:* $max(s_i, s_j) = s_i \quad if \quad s_i \geq s_j$
4. *Min operator:* $min(s_i, s_j) = s_i \quad if \quad s_i \leq s_j$

Example 3.1 *Example of linguistic terms set*
S = { s_{-4} *= Certainly False (CF),* s_{-3} *=Almost Certainly False (ACF),*
s_{-2} *=Highly Uncertain (HU),* s_{-1} *=Probably False (PF),* s_0 *=Undecided,*
s_1 *=Probably True (PT),* s_2 *=Highly Likely (HL),*
s_3 *=Almost Certainly True (ACT),* s_4 *=Certainly True (CT) }.*

3.3 EXPERTISE PROCESS FRAMEWORK

After elaborating on essential helpful concepts for hypotheses reasoning processes, this section describes the data structure that captures the knowledge emanating from it. In addition, reasoning mechanisms from this knowledge structure, queries, and doubt computations of this knowledge structure are presented.

3.3.1 *HEG structure*

The structure of the expertise process knowledge is divided into two main components: the core HEG and its support. The subsections below concern these components.

3.3.1.1 *Core Hypotheses Exploratory Graph (HEG)*

The HEG is a *directed acyclic* graph with unique relation called *question* among edges which are *hypotheses*. A HEG can be *complete* or not, its construction is an iterative process that begins with the problem and grows as experts attempt to explain the problem at hand.

The following steps show how the HEG definition is derived:

- $G_0 = (V_0, E_0, \emptyset, K_0)$, where
  $V_0 = Problem$ corresponds to the initial problem,
  $E_0 = \emptyset$ means there are no edges,
  K_0 stands for the initial knowledge available at the beginning of the expertise. There are neither questions nor hypotheses at this step, and there was no previous graph; that is why there is an empty set at the fourth component.
- First iteration: $G_1 = (V_1, E_1, G_0, K_1)$:
  $V_1 = \{V_0, h_{1,i}\}_i$ and $E_0 \subseteq E_1$, where $h_{1,i}$ are hypotheses expressed at the first iteration.
  $K_1 = \{K_0, \triangle K_1\}$ with $\triangle K_1$ corresponding to additional knowledge at this iteration. This graph results from the first iteration of the expertise process.
- The second iteration of the expertise process is given by:
  $G_2 = (V_2, E_2, G_1, K_2)$ where,
  $V_2 = \{P, h_{1,i}, h_{2,j}\}_{i,j}$, where $h_{2,j}$ are hypotheses expressed at the second iteration
  $K_2 = \{K_0, K_1, \triangle K_2\}$ and
  $E_1 \subseteq E_2$.
- The HEG can be organized as follows:
  $G_k = (\{P, h_{i,j}\}_{i<=k,j}, E_k, G_{k-1}, K_k)$ which corresponds to :

$$\begin{cases} G_0 = (V_0, E_0, \emptyset, K_0) \\ G_n = (V_n, E_n, G_{n-1}, K_n) \end{cases} \tag{3}$$

To sum up, HEG deals with the following vocabulary: **Observation** which identifies all additional knowledge used during the graph construction, **hypothesis**, **problem** corresponding to the initial node of the graph, **question**.

The definition below integrates linguistic set terms and possibility theory for simple computation of human doubt in the expertise process.

Definition 3.3 ***Possibility linguistic distribution***

Experts' doubts are taken into consideration with the use of possibility theory, from which hypotheses are described with:
Let Ω be the universe of all hypotheses that can be expressed by experts.
Let U be the set of understanding that hypotheses try to explain.
The possibility linguistic distribution function defined on Ω is given by:

$\pi^* = L \circ \pi : \Omega \longrightarrow [0,1] * [0,1]$
$\pi : \Omega \longrightarrow S$, *where S is a linguistic term set.*
$L : S \longrightarrow [0,1] * [0,1]$

$$L(s_i) = \begin{cases} (1 - l(s_i), 0) & if \quad s_i < Undecided \\ (1, l(s_i)) & if \quad s_i > Undecided \\ (1, 0) & if \quad s_i = Undecided \end{cases},$$

$l(s_i) \in [0,1]$ is a predefined function to set quantities on linguistic terms.

Example 3.2 *Possibility linguistic distribution*
For the understanding of the rejection of a manufacturing product by customers, the following model is illustrated.

Let Ω be the universe which contains all possible hypotheses that can be asked to understand the problem mentioned.
u = "Why were KW831 products rejected by customers?",
a point of interest.

A subset $A = \{h_1, h_2\} \subseteq \Omega$,

where:
h_1 = *"It is almost certainly true that it is due to faulty measurement tools."*
h_2 = *"It is highly likely that it is due to non-compliance with the manufacturing plan."*

Given the following linguistic distribution:
$l(CF) = l(CT) = 1$, $l(ACF) = l(ACT) = 0.75$, $l(HU) = l(HL) = 0.5$, $l(PF) = l(PT) = 0.25$

$L(CF) = (0,0)$, $L(ACF) = (0.25, 0)$, $L(HU) = (0.5, 0)$, $L(PF) = (0.25, 0)$, *terms less than Undecided (U)*

$L(CT) = (1,1)$, $L(ACT) = (1, 0.75)$, $L(HL) = (1, 0.5)$, $L(PT) = (1, 0.25)$, *terms greater than Undecided (U)*

$L(U) = (1,0)$

For example,
$\pi^*(h_1) = L \circ \pi(h_1)$
$\pi^*(h_1) = L(ACT)$
$\pi^*(h_1) = (1, 0.75)$

From this example, saying that one is *Almost Certain* a hypothesis $h_{i,j}$ is false, reflects a 75% confidence of the falsity of $h_{i,j}$, which give a *possibility* of 0.25 for this hypothesis to be true and a *necessity* of 0, that corresponds to 0.75 of possibility for every any other hypothesis to be true.

An event $A \subseteq \Omega$, $A = \{h_{i,j}\}_{i,j}$ of mutually exclusive hypotheses which corresponds a given understanding $u_i \in U$ of a problem.
$u_i = A$ means that hypotheses in A are intended to explain u_i
$\pi^*(h_{i,j}) = (x, y)$ means that the hypothesis $h_{i,j}$ has x possibility to be valid and y necessity.

From an event A, one can compute for each iteration of the graph :

- Possibility of an event A:
  $\Pi(A) = sup_{h_{i,j} \in A}\{\pi^*(h_{i,j})\}$
  Evaluates to what extent there are hypotheses in A which are possible.

- Necessity of an event A:
  $N(A) = 1 - \Pi(\overline{A})$, where $\overline{A}$ is the complement of A
  Evaluates to what extent none of the hypotheses in A is possible.

To process hypotheses and their terms, the following operators are defined. Their combinations will permit decision-making on HEG at the end of the exploratory process. These operators will also be important during reasoning steps.

$$\begin{gathered} The \quad OR(\vee) \quad of \quad two \quad linguistic \quad terms \quad s_i \quad and \quad s_j \\ is \quad given \quad by: \quad (s_i \vee s_j) = max(s_i, s_j) \end{gathered} \tag{4}$$

$$\begin{gathered} The \quad AND(\wedge) \quad of \quad two \quad linguistic \quad terms \quad s_i \quad and \quad s_j \\ is \quad given \quad by: \quad (s_i \wedge s_j) = min(s_i, s_j) \end{gathered} \tag{5}$$

For the case of possibility and necessity values, the following operators will be used to combine them during reasoning iterations or at the end of the Expertise Process to make decisions:
$\forall A_i, A_j$ sets of hypotheses of a HEG,

$$The \quad possibility \quad of \quad \Pi(A_i \vee A_j) = max(\Pi(A_i), \Pi(A_j)) \tag{6}$$

$$The \quad necessity \quad of \quad N(A_i \wedge A_j) = min(N(A_i), N(A_j)) \tag{7}$$

After reviewing linguistic and uncertainty tools used by experts to express their doubt, the subsequent sections will elaborate on details of the HEG followed by how to use these tools for decision-making.

3.3.1.2 *HEG nodes and edges*

HEGs are directed acyclic graphs in which nodes and relations among nodes have specific semantics.
The relation among nodes is defined as follows:

Definition 3.4 ***HEG triples***
The triple $< h_{i,j}, e, h_{k,l} >$, *with* $i < k$ *of a HEG stands for:* $h_{k,l}$ *can be an explanation to* $h_{i,j}$ *under question* e.

Importantly, hypothesis nodes are characterized by three elements which are: $h_{i,j} = (Hf, \xi_n, \pi(h_{i,j}))$

- Hf is a hypothesis as defined by the hypothesis logic.
- $\xi_n = (\xi_i, \xi_{i+1}, \ldots, \xi_n)$: where $\xi_m \in \{Valid, Unknown\}$, $i \leq m \leq n$ is the status of $h_{i,j}$ at iteration m.
- $\pi(h_{i,j}) \in S$, S is a linguistic term set.
  $\pi(h_{i,j})$ is its possibility linguistic term corresponding to the doubt associated with the Hypothesis at the last iteration.

Semantically, for a given iteration n, $h_{i,j} = (Hf, \xi_n, \pi(h_{i,j}))$ means that hypothesis $h_{i,j}$ with logical formula Hf, and status $\xi_n \in \{Valid, Unknown\}$ was expressed with linguistic doubt $\pi(h_{i,j})$ corresponding to how certain was an agent of its hypothesis with respect to a question. The values $\pi^*(h_{i,j})$ correspond to how the agent is sure of this Hypothesis.

3.3.1.3 *HEG's support*

This section defines supports for HEGs to semantically enhance the knowledge described by HEGs. The *support* of a HEG adds meaning to its vertices and edges such that it will be possible to infer new knowledge from the HEG. This support, similar to the one proposed by (Mugnier, 2000) for conceptual graphs, provides meaning for hypotheses and questions through taxonomies and linguistic possibilities. In other words, it describes the domain knowledge used for constructing a HEG.

Definition 3.5 ***Support***
A support of a HEG $S = (T_{HEG}, T_H, W, \pi^*)$, with:

- *T_{HEG} is a taxonomy that gives semantics to components of the HEG.*
- *T_H is a taxonomy that gives semantics to hypotheses of a HEG.*
- *W is the set containing classes of questions having {What, Why, Where, Who, When, How } used to specify the type of question. These classes of questions are also known as the 5W1Hs*
- *π^* is the possibility linguistic distribution.*

This classification of questions with the 5W1H is used in order to externalize different dimensions of experts' knowledge (Huang and Kuo, 2003).
Figures 16 and 17 depict respectively the HEG taxonomy and a hypothesis taxonomy. Extending or defining a hypothesis taxonomy that captures a problem better will be of great advantage.
The *support* gives semantics to both questions and hypotheses. Furthermore, HEG's *support* assists experts' collaboration as its taxonomies set a common understanding for humans working on the same problem. Authors of (Meléndez et al., 2018) use the same approach for collaborative experience in industrial processes.
From the above support, the following definition emerges for questions of a HEG. A question is defined by

$$q := W(q) \tag{8}$$

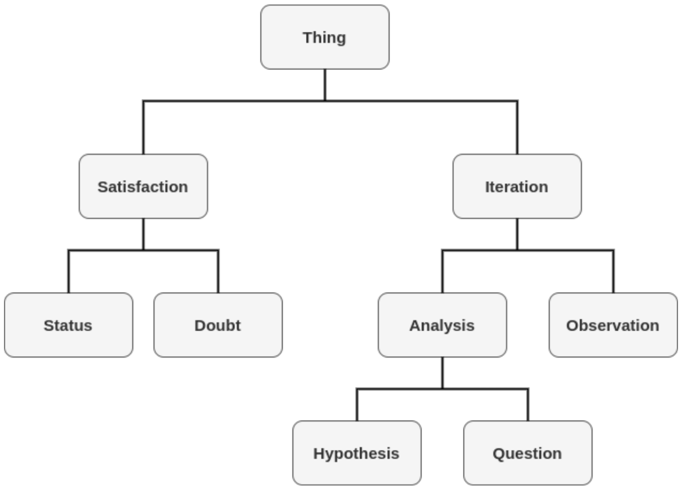


Figure 16: HEG taxonomy

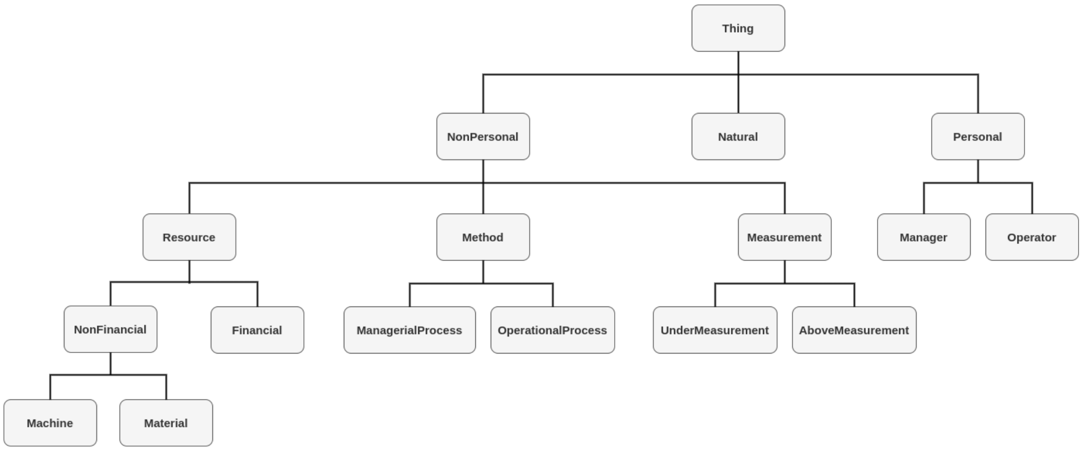


Figure 17: Hypotheses taxonomy

, where *W* is a class among the 5W1Hs.

Example 3.3 ***Example taken from an expertise report[1] carried by BEA[2]***
Hypothesis1*: The pilot certainly deviated from the instrumental approach path.*
Question*: Why did the deviation happen?*
Hypothesis2*: It is probably true, that the pilot relied on erroneous external visual references acquired shortly before the decision altitude.*

From this example, *Hypothesis1* and *Hypothesis2* can be labeled with *Operator* concept and question *Question* with label *Why*. The use of {*Valid* and *Unknown*} and not {*Valid* and *Invalid*} for hypotheses statutes was on purpose. In fact, it is assumed that a hypothesis is not valid due to insufficient knowledge. This reasoning approach corresponds to the Open World Assumption (OWA), which is opposed to the Closed-World Assumption (CWA), from which what is not true (Valid for the proposed case) is false (Invalid) (Grimm and Motik, 2005). This choice of the OWA is not only motivated by the lack of complete knowledge about the world but also by the importance of having a non-monotonic mechanism on hypotheses statutes.

1 Accident of the Piper PA34-200T Seneca III registration HB-LSD on December 7, 2016 in Basel - Mulhouse
2 https://bea.aero/

Property 3.1 ***Open World Assumption (OWA)***
HEG adheres to the Open World Assumption (OWA): non-existing triples, as well as non-valid hypotheses, have unknown status.

3.3.2 *Reasoning over HEG*

From the knowledge structure of the previous section, reasoning mechanisms to infer hidden knowledge are presented in the sections below.

3.3.2.1 *Reasoning process*

The reasoning process over a HEG is based on the Hypothesis Theory. This theory defines three types of information which are: (1) truth made of FOL formulas, (2) known formulas expressed using the L operator, and (3) hypotheses that are defined with the H operator. More precisely, the additional knowledge of the graph is composed of the first two types (truth, and known formulas), and the vertices are hypotheses.
The reasoning process consists in checking if the hypothesis theory defined by a HEG has an extension for its additional knowledge from its iterations. In other words, if there is a subset of vertices that is consistent with this additional knowledge.
Let G_n be a HEG, this implies verifying if there is a $V' \subseteq V_n$ such that $(K_n \cup V')$ is the extension of K_n in V_n. However, at each reasoning process, the statutes of hypotheses (nodes) belonging to the extension set (V') are *valid* and stay unchanged if they are valid at the previous iteration. Nevertheless, if a hypothesis was previously in the set of extensions and did not belong at the current iteration, its status changes from valid to *unknown*. This status stays unchanged if it was not in the previous hypothesis extension set.

Example 3.4 *Valid and unknown status of hypotheses*
Assuming G is a HEG with two hypotheses h_1, h_2 and an additional knowledge K_1:
If $K_1 = \{Lp, Lp \wedge Hq \rightarrow Lr, \neg Hr\}$ which stands for:
$\{$p is known, If p is known and q is a hypothesis then r is known, r is not a hypothesis $\}$
$h_1 = Hq$,
$h_2 = Hr$
then, the extension on G is $E = K_1 \cup \{h_1\} = \{Lp, Lr, \neg Hr\}$.
From this extension, it can be concluded that h_1 is a Valid hypothesis whereas h_2 has an Unknown status.

3.3.2.2 *Definitions and properties from HEGs*

The following definitions and properties can be obtained from the above formalization of the Expertise Process and its reasoning mechanism.

Definition 3.6 ***Goal-directed HEG***
*A HEG is considered **goal-directed** if each of its iterations intends to achieve a specific goal.*

Definition 3.7 ***Valid hypothesis***
A hypothesis h is valid for a HEG G_n if h belong to the sub-set of hypotheses of G_n's extension. In other words, If G_n is a HEG, h a hypothesis of G_n and E be the extension of G_n, then h is valid for G_n if $h \in E$

Definition 3.8 ***Valid path***
A valid path ρ *of a HEG* G_n *is any sequence of valid hypotheses of the graph starting from the first iteration to the last iteration and containing exactly one valid hypothesis at each iteration. Let* G_n *be a HEG of* $n \in \mathbb{N}, n \geq 1$ *iterations.*

$$\rho = \{(h_{1,j}, h_{2,k}, \ldots, h_{n,l})\ h_{i,j} \in V_n \quad and \quad h_{i,j} \quad is \quad valid \quad in \quad G_n\} \tag{9}$$

A valid path can also be defined as a directed path starting from the first iteration to the last iteration and containing exactly one valid Hypothesis at each iteration.

Definition 3.9 ***Successful expertise***
Expertise is successful if the HEG obtained from it has an extension with at least a valid path that has its hypotheses in the set of hypotheses of this extension.

Definition 3.10 ***Having expertise***
An agent is set to have expertise after an exploratory reasoning process if consciously, the graph obtained after the Expertise Process is a successful expertise.

Definition 3.11 ***Valid iteration***
An iteration of an expertise process is said to be valid if it has at least one valid Hypothesis.

Property 3.2
If G_n *is a successful expertise, then each iteration of* G_n *has at least a valid hypothesis. The reciprocal of this is not always true.*

Property 3.3
Let $G_n = (V_n, E_n, G_{n-1}, K_n), n \in \mathbb{N}, n \geq 1$ *be a HEG and* $V' \subseteq V_n$ *such that* $(K_n \cup V')$ *is the extension of* K_n *in* V_n*.*
If $V' = \emptyset$ *then* G_n *is not a successful expertise.*

Property 3.4
Let $G_n = (V_n, E_n, G_{n-1}, K_n), n \in \mathbb{N}, n \geq 1$ *be a HEG and* $V' \subseteq V_n$ *such that* $(K_n \cup V')$ *is the extension of* K_n *in* V_n*.*
If $V' = V_n$ *then* G_n *is a successful expertise and has multiple valid paths.*

3.3.3 *Querying hypotheses exploratory graph*

In order to draw some new inferences from a HEG, this section presents a simple language to query specific knowledge from a HEG. The three types of knowledge that can be extracted from this graph are: The explanation or valid paths, the questions that were asked, and the hypotheses expressed.

- **Inference explanations or valid paths**
  Suppose G is a HEG obtained after multiple iterations, h_i, h_j hypotheses of G and $i, j \in \mathbb{N}$ some iterations numbers.
  The following queries can be made on G

Code 1: Querying all valid paths

```
FROM G
EXPLAIN *
```

Code 2: Querying all valid paths having hypothesis h_i

```
FROM G
EXPLAIN *
HAVING h_i|q
```

- Query all explanations of the problem
  This query in Code 1 looks for all explanations, also known as valid paths of the problem being expertized from the graph G.

- Query specific explanations
  This query in Code 2 looks for all explanations, also known as valid paths of the problem being expertized having a given hypothesis h_i or question q. The $HAVING$ clause can be used with operators such as AND, OR. If used with AND ($h_i \quad AND \quad h_j \quad AND...$), each valid path must have this list of hypotheses.
  If used with OR ($h_i \quad OR \quad h_j \quad OR...$), each valid path must have at least one of the listed hypotheses.
  Same as for hypotheses, operators AND and OR are used for questions. Particularly, the hypotheses or questions used at the $HAVING$ clause can come from sub-queries.
- Query a number of explanations
  This query in Code 3 looks for $i \in \mathbb{N}$ number of explanations from G. This query can be extended to a given number of explanations having some specific hypotheses.

- **Query questions**
  Querying this component of a HEG assists in finding the questions asked during the Expertise Process.
  - Query all questions from the graph
    This query in Code 4 will return all questions from the graph.
  - Query all questions at iteration i
    This query in Code 5 will return all questions asked at the iteration i of the graph. One can also provide a list of iterations $(i, j, k, ...)$ to return all

Code 3: Querying i number of valid paths

```
FROM G
EXPLAIN i
[HAVING ...]
```

Code 4: Query all questions used in a HEG

```
FROM G
QUESTION *
```

Code 5: Query all question at iteration i from a HEG

```
FROM G
QUESTION *
AT i
```

questions at each given iteration.
Used with AND, the AT clause will correspond to an interval as follows:
$AT \quad i \quad AND \quad j = [i, j]$
$AT \quad i \quad AND \quad * = [i, \rightarrow]$
$AT \quad * \quad AND \quad j = [\leftarrow, j]$

- Query questions from iteration i to iteration j
This query in Code 6 will return $TRUE$ if it exists in the graph G an $FALSE$ if not.
The question clause can also have a list of questions separated by AND or OR operators. If used with AND ($q_1 \quad AND \quad q_2 \quad AND$...), the query will return $TRUE$ if all these questions were asked in G and $FALSE$ if not.
If used with OR ($q_1 \quad OR \quad q_2 \quad OR$...), the query will return $TRUE$ if at least one of the listed questions was asked in G.

- **Query observations**
Observations correspond to additional knowledge used during the Expertise Process. Just like hypotheses or questions, it is important to know the knowledge used for a problem.
  - Query all available knowledge
  This query in Code 7 will return knowledge available at all iterations of the graph.
  - Query knowledge at specific iteration
  This query in Code 8 will return knowledge available at iterations described in the AT clause.

- **Query hypotheses**
This query in Code 9 without the AT clause will return all hypotheses used in the graph.

Code 6: Querying all valid paths

```
FROM G
QUESTION q
[AT ...]
```

Code 7: Query all observations of a HEG

```
FROM G
OBSERVATION *
```

Code 8: Query all observations of a HEG at iteration i

```
FROM G
OBSERVATION *
[AT ...]
```

If the *AT* clause is used, the query will return hypotheses as specified in this clause.

3.3.4 *Doubt updating mechanism*

Concerning the reasoning mechanism proposed above, the doubt given to hypotheses is not static; it can be increased or decreased based on their validity (or unknown) status after reasoning. Indeed, each time a hypothesis status is *unknown* after reasoning, it increases disruption in information about the expertise process, thereby strengthening its doubt (Durmaz, Demir, and Sezen, 2021). This mechanism is done in two steps as follows:

- Initially, the doubt of a hypothesis is set by experts, but if it is not the case, its default status value is *undecided*. However, whatever the doubt of a hypothesis, if its status becomes *valid* after reasoning, then its doubt reduces.
  Whereas, if its status becomes *unknown* after reasoning, then its doubt remains unchanged.
  An exception for the above update rules is the case of the initial *undecided* Hypothesis, which status becomes *unknown* after reasoning. For this case, there is an increase in doubt.

  Example 3.5 *Hypotheses doubt updates*
  In the case of a stepwise update, the doubt will decrease from its initial value of ***undecided*** *to* ***probably true*** *if it becomes valid, but if its status was* ***unknown****, then its doubt increases and becomes* ***probably false****.*

- The doubt of hypotheses changes over the expertise process at each iteration.
  In general, hypotheses' doubts remain unchanged if their statutes does not change. But, if their statutes change from *valid* to *unknown*, their doubts increase and decrease otherwise.

Code 9: Query all observations of a HEG at iteration i

```
FROM G
HYPOTHESIS *
[AT ...]
```

These doubts updates can be defined by the following function and summarized in table:
Let S be a linguistic term set.
$S = \{s_t | t = -\tau, \dots, -1, 0, 1, \dots, \tau\}$, where $\tau > 0$,

$\Phi_{update} : \{Unknown, Valid\}^2 * L \rightarrow L$
$\Phi_{update}(\xi_i, \xi_{i+1}, L(s_k))$ where $s_k \in S$, ξ_i is the state before reasoning (iteration i), $L(s_k)$ the linguistic doubt at iteration i and ξ_{i+1} the state after reasoning (iteration $i+1$). This function returns the linguistic doubt at iteration $i+1$.

$$\Phi_{update}(\xi_i, \xi_{i+1}, L(s_k)) = \begin{cases} L(s_{min\{k+1,\tau\}}) : if \quad \xi_i = Unknown \quad and \quad \xi_{i+1} = Valid \\ L(s_k) : if \quad (\xi_i = Valid \quad and \quad \xi_{i+1} = Valid) \\ or \\ \quad (\xi_i = Unknown \quad and \quad \xi_{i+1} = Unknown) \\ L(s_{max\{-\tau,k-1\}}) : if \quad \xi_i = Valid \quad and \quad \xi_{i+1} = Unknown \\ \Phi_{update}(Unknown, Unknown, L(s_0)) = L(s_{-1}) : \\ In \quad case \quad of \quad ignorance \end{cases} \tag{10}$$

Table 5 summarizes the output of this update function of Formula 10.

Iteration i	**Iteration $i+1$**	**Explanation**
From $L(s_k)$	to $L(s_k)$	Doubt unchanged
From $L(s_k)$	to $L(s_{k-1})$	Doubt increased
From $L(s_k)$	to $L(s_{k+1})$	Doubt decreased

Table 5: Doubt update from iteration i to iteration $i+1$

3.3.5 *Trust over HEG*

The context in which experts explore potential explanations of a problem is one in which there is limited knowledge. Therefore, evaluating the trust of the results at hand is important, especially if they have to choose the most plausible outcome after explorations.
To compute the overall uncertainty, this section presents an integration of possibility and logic theories. The approach proposed consists of defining possibility and necessity for each type of formula of the hypothesis logic.

$$\begin{cases} Truth \quad formulas \quad (\Pi = 1, N = 1) \\ Known \quad formulas \quad (\Pi = 1, N \approx 1) \\ Hypothesis \quad formulas \end{cases} \tag{11}$$

In addition to the context in which expertise takes place, it is essential to note that it is a human-centered task; therefore, the trust manipulation should be easy to express

and understand by them.
In order to fulfill the above requirements, trust will be designed with linguistic terms as follows:
Let $S = \{certainly\ false\ (CF), almost\ certainly\ false\ (ACF), highly\ unlikely\ (HU), probably\ false\ (PF), undecided, (U) probably\ true\ (PT), highly\ likely\ (HL), almost\ certainly\ true\ (ACT), certainly\ true\ (CT)\}$
be our linguistic terms set.
(s_{-4} = *certainly false*, ..., s_0 = *undecided*, ..., s_{+4} = *certainly true*),
that is compiled from qualitative possibility values proposed by (Walker, 2006).
S has the following characteristics:

- S is an ordered set: $s_\alpha > s_\beta$, if $\alpha > \beta$;
- The negation operator $neg(s_\alpha) = s_\beta$ such that $\alpha + \beta = 0$

Let π^* be a possibility linguistic distribution of a HEG and a linguistic term set S.
This uncertainty management on hypotheses expressed by experts can be done either by pairwise dependency among hypotheses or in consideration of independent doubts.

Considering a simple case of a uniform step scale on the linguistic terms, the following association of possibility on linguistic terms can be obtained.

Example 3.6 *An example of linguistic terms distribution*
$L(S) = \{CF : (0,0), ACF : (0.25,0), HU : (0.5,0), PF : (0.75,0), U : (1,0), PT : (1,0.25), HL : (1,0.5), ACT : (1,0.75), CT : (1,1)\}$.

However, it is possible to have different scales and possibility distribution for the linguistic terms.

3.3.5.1 *Doubt of possible explanations*

To compute the possibility of expertises, the possibilities of its paths are first computed.
Suppose G_n is a HEG and ρ is a valid path of G_n. The possibility of ρ is given by the following expression.

$$\Pi(\rho) = sup_{h_{i,j} \in \rho}\{\pi^*(h_{i,j}) / h_{i,j} \in \rho\} \qquad (12)$$

This study derives the final possibility from some principles defined by (Yager, 1995).
These strategies are as follows.

- ***Pessimistic strategy***.
  The uncertainty of a *successful expertise* is equal to the *minimum* possibility of its valid paths.

  $$N^-(G_n) = min\{\Pi(\rho_i) / \rho_i \in G_n\} \qquad (13)$$

- ***Optimistic strategy***.
  The uncertainty of a *successful expertise* is equal to the *maximum* possibility of its valid paths.

  $$N^+(G_n) = max\{\Pi(\rho_i) / \rho_i \in G_n\} \qquad (14)$$

- ***Balanced strategy***
  The uncertainty of a *successful expertise* is equal to the *average* possibility of its valid paths.

$$N(G_n) = \frac{\sum \Pi(\rho_i)}{N} / \rho_i \in G_n, \quad N \ \ is \ \ the \ \ number \ \ valid \ \ paths \tag{15}$$

From the above necessity of a valid path and a HEG, it is possible to define a preference relation on paths and graphs.

Definition 3.12 ***Preference among graphs***
Let $G1_n$ and $G2_n$ be two HEGs obtained from the same problem with the same number of iterations n.
$G2_n$ is preferred to $G1_n$ ($G1_n \succ G2_n$) if the uncertainty of $G2_n$ is less than the uncertainty of $G1_n$

Similarly:

Definition 3.13 ***Preference among valid paths***
Let ρ_i and ρ_j be two valid paths from a HEG.
ρ_i is preferred to ρ_j ($\rho_j \succ \rho_i$) if the uncertainty of ρ_i is less than the uncertainty of ρ_j

3.3.6 *Illustration of the proposed methodology*

The proposed approach is demonstrated in a real-world case in a manufacturing company to show how it works. For this illustration, experts were asked to use the proposed approach to look for explanations of why customers returned an article that an enterprise manufactures.

- **Iteration 0**: It corresponds to the problem.
- **Iteration 1**:
  - Question: Why were KW831 products rejected by customers?
  - Hypotheses:
    * $h_{1,1}$
      · **Hypothesis**:It is *almost certainly true* that it is due to faulty measurement tools.
    * $h_{1,2}$
      · **Hypothesis**: It is *highly likely* that it is due to non-compliance with the manufacturing plan.
    * $h_{1,3}$
      · **Hypothesis**: It is *highly likely* that it is due to the over-tightening of its parts.
  - Observation:
    * Some operators were not trained to use measurement tools, so some could not measure KW831 components well.

  * Measurement tools are new and were tested well before usage; therefore they are not faulty.
- Reasoning
  * $h_{1,1}$
    · **Hypothesis**: *almost certainly true* that it is due to faulty measurement tools.
    · **Status**: Unknown.
  * $h_{1,2}$
    · **Hypothesis**: It is *almost certainly true* that it is due to non-compliance with the manufacturing plan.
    · **Status**: Valid.
  * $h_{1,3}$
    · **Hypothesis**: It is *highly likely* that it is due to the over-tightening of its parts.
    · **Status**: Unknown.

  Remarks: For the reasoning process, hypotheses with *Unknown* status are those that were not supported by observations, whereas *Valid* hypotheses are those that are consistent with observations.
  This mechanism is used at every iteration.

• **Iteration 2**:
- Question: Why were these recently manufactured KW831s poorly tightened?
- Hypotheses:
  * $h_{2,1}$
    · **Hypothesis**: It is *probably true* that it is because operators poorly did the work.
- Observation:
  * Only recently manufactured KW831 are rejected by customers.
- Reasoning:
  * $h_{1,1}$
    · **Hypothesis**:It is *almost certainly true* that it is due to faulty measurement tools.
    · **Status**: Unknown.
  * $h_{1,2}$
    · **Hypothesis**: It is *highly likely* that it is due to non-compliance with the manufacturing plan.
    · **Status**: Unknown.
  * $h_{1,3}$

      · **Hypothesis**: It is *highly likely* that it is due to the over-tightening of its parts.
      · **Status**: Unknown.
    * $h_{2,1}$
      · **Hypothesis**: It is *probably true* that it is because operators poorly did the work.
      · **Status**: Unknown.

- **Iteration 3**:
  – Question: Why were the dimensions of the KW831 parts not respected?
  – Hypotheses:
    * $h_{3,1}$
      · **Hypothesis**: It is *probably true* that it is due to measuring errors.
  – Question: Why are these newly recruited operators not good?
    * $h_{3,2}$
      · **Hypothesis**: It is *certainly true* that operators may not have been well trained on the production line.
  – Observation:
    * There are newly recruited operators, so they could poorly mount or measure KW831 components.
    * Operators worked under pressure in order to deliver KW831 products on time, so it is possible to have manufacturing errors.
    * Newly recruited operators are inexperienced workers.
  – Reasoning:
    * $h_{1,1}$
      · **Hypothesis**:It is *almost certainly true* that it is due to faulty measurement tools.
      · **Status**: Unknown.
    * $h_{1,2}$
      · **Hypothesis**: It is *almost certainly true* that it is due to non-compliance with the manufacturing plan.
      · **Status**: Valid.
    * $h_{1,3}$
      · **Hypothesis**: It is *almost certainly true* that it is due to the over-tightening of its parts.
      · **Status**: Valid.
    * $h_{2,1}$
      · **Hypothesis**: It is *highly likely* that it is because operators poorly did the work.

- **Status**: Valid.

* $h_{3,1}$
  - **Hypothesis**: It is *undecided* that it is due to measuring errors.
  - **Status**: Unknown.
* $h_{3,2}$
  - **Hypothesis**: It is *certainly true* that operators may not have been well trained on the production line.
  - **Status**: Valid.

From this exploratory process, the hypotheses statutes, doubts and the reasoning steps can be tracked. The graph in Figure 18 shows how the statutes of hypotheses changed for each iteration, and the graph in Figure 19 shows the final statutes of the HEG described in this illustration.

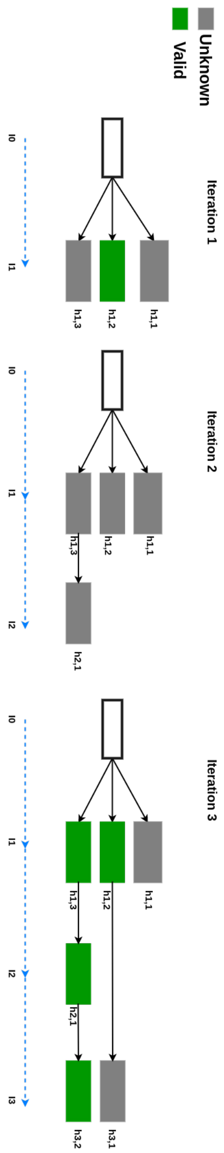


Figure 18: Hypotheses statutes at each iteration. For final graph see figure 19

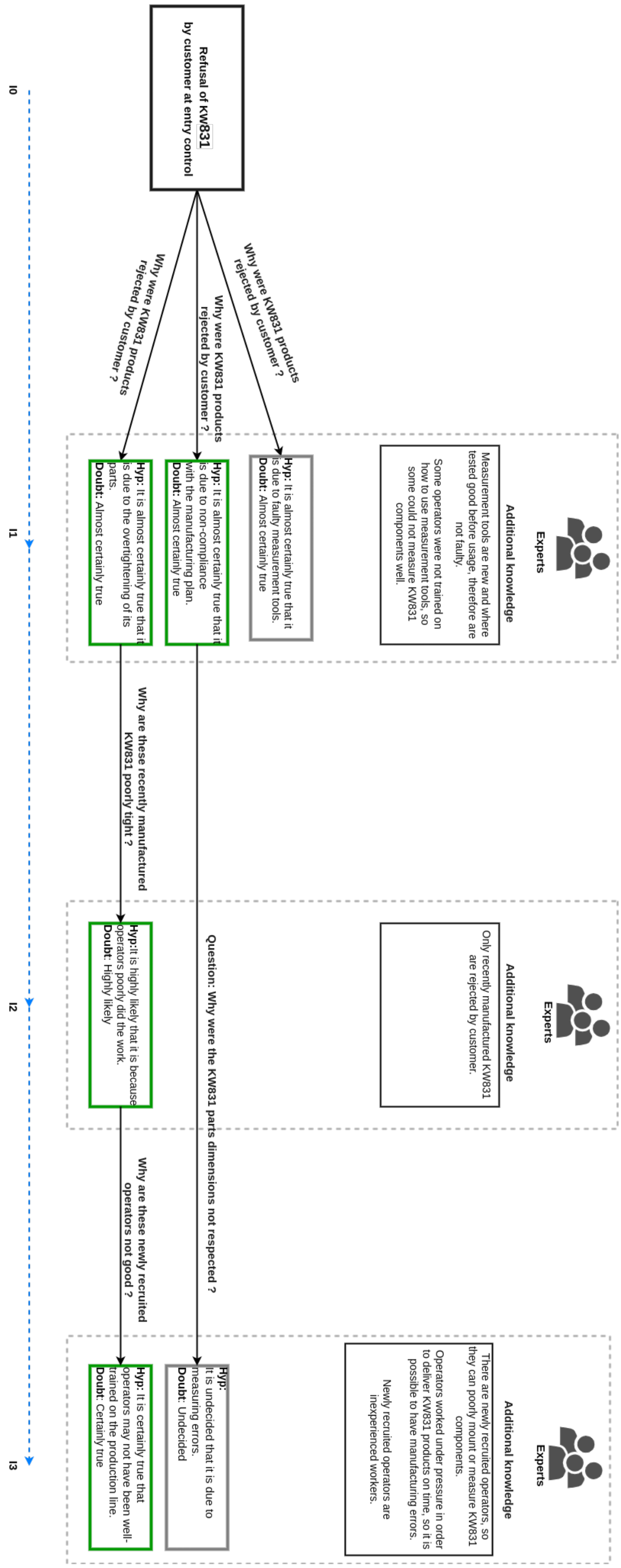


Figure 19: Final representation of the HEG described in this illustration. For process see figure 18

For this example, Table 6 shows how the doubts are updated with respect to their status at each iteration, and Table 7 shows how hypotheses' statutes change at each iteration.

Hyp	Initial doubt	Iteration 1	Iteration 2	Iteration 3
$h_{1,1}$	Almost certainly true	Almost certainly true	Almost certainly true	Almost certainly true
$h_{1,2}$	Highly likely	Almost certainly true	Highly likely	Almost certainly true
$h_{1,3}$	Highly likely	Highly likely	Highly likely	Almost certainly true
$h_{2,1}$	Probably true	-	Probably true	Highly likely
$h_{3,1}$	Probably true	-	-	Undecided
$h_{3,2}$	Certainly true	-	-	Certainly true

Table 6: Hypotheses doubt evolution with respect to iterations

Hypotheses	Initial Status	Iteration 1	Iteration 2	Iteration 3
$h_{1,1}$	Unknown	Unknown	Unknown	Unknown
$h_{1,2}$	Unknown	Valid	Unknown	Valid
$h_{1,3}$	Unknown	Unknown	Unknown	Valid
$h_{2,1}$	Unknown	-	Unknown	Valid
$h_{3,1}$	Unknown	-	-	Unknown
$h_{3,2}$	Unknown	-	-	Valid

Table 7: Hypotheses statutes evolution with respect to iterations

3.3.7 *Knowledge derived from* HEG

Deriving new knowledge from a HEG is a valuable asset of the proposed approach. To achieve these values, graph analysis, semantic mapping and causal reasoning techniques were applied on HEG.
Details of these techniques are described in the following sub-sections.

3.3.7.1 *Valid nodes' importance or ranking*

Graph ranking or network ranking is a widely used technique in economic, social, or political domains with various applications in daily activities. It determines the importance of nodes in a graph based on elements that characterize then. Simple methods that can be used are ranking nodes according to their degree of connectivity, which corresponds to the set of nodes that are adjacent to it (Desouki, Röder, and Ngonga Ngomo, 2019; Van Den Brink and Rusinowska, 2021).

A similar method is used in this chapter on HEGs. In fact, after an Expertise Process, it is essential to determine valid critical hypotheses that participated significantly in the process, which can stand as primary hypotheses for the problem or given priority for an upcoming, similar problem. The importance of a hypothesis in a HEG, like the ranking of nodes used in graph structures, defines a complete preorder of hypotheses (Van Den Brink and Rusinowska, 2021). It determines the importance of a hypothesis relative to elements of the HEG in which it belongs.

This study defines valid hypotheses' importance by considering the number of incoming and outgoing questions of a hypothesis but also by the validity of the hypotheses of these questions. For a given hypothesis h of a HEG, this importance is given by the following formula:

$$R(h) = (out + in) * \alpha + \beta \tag{16}$$

$\alpha = (1 - \frac{1}{N})$, N is the total number of questions coming or going out of the Hypothesis. This value gives credit to hypotheses for the number of questions edges coming or going out of them.

$\beta = \frac{I-i}{I}$, I is the total number of iterations of the HEG, and i is the iteration on which the Hypothesis belongs.

$out = \frac{nv_o}{n_o}$, nv_o is the number of valid hypotheses from out-going questions and n_o is the total number of out-going questions.

For a hypthesis h,

$n_o = |\{e \quad question : \exists h_{i,j}, < h, e, h_{i,j} > \in \text{HEG}\}|$

This value credits hypotheses for the number of valid hypotheses attached to their outgoing questions.

$in = \frac{nv_i}{n_i}$, nv_i is the number of valid hypotheses from incoming questions and n_i is the total number of incoming questions.

For a hypthesis h,

$n_i = |\{e \quad question : \exists h_{i,}, < h_{i,}, e, h > \in \text{HEG}\}|$

This value credits hypotheses for the number of valid hypotheses attached to their incoming questions.

In summary, the importance of a hypothesis depends on one hand on its surrounding (out-going and incoming hypotheses) and, on the other hand, on the general structure of the Expertise Process (number of iterations, number of questions asked). Hypotheses in HEG with more exploration are favored than those appearing in less explored problems. The proposed measurement is different from node importance evaluations such as *node centrality measure* (degree centrality, closeness centrality) and *node betweenness centrality* because they rely only on the neighboring node. In contrast, the proposed evaluation also depends on the graph structure (Dudkina et al., 2021). However, applying *PageRanking* on HEG will be challenging since it has a unidirectional path.

$$\lim_{N \to \infty} \alpha = 1$$

Hypotheses closer to the end of the exploration are less favored than those close to the problem.

$$\lim_{i \to I} \beta = 0$$

For particular cases, the problem is considered to have unknown status.

3.3.7.2 *Causal inference*

From a HEG, hidden knowledge can be revealed from the behavior of hypotheses regarding additional knowledge used during the Expertise Process. This is possible because each hypothesis $h_{i,j}$ has its sequence of states $\xi_n = (\xi_i, \xi_{i+1}, \ldots, \xi_n)$ that vary from iteration to iterations based on additional knowledge.
This hidden knowledge is embedded in the cause-effect graph between hypotheses and additional knowledge that can be extracted from a HEG by considering the change of states over iterations of hypotheses as effects of additional knowledge.
These causal graphs obtained from HEGs are learned from a table of hypotheses and additional knowledge where rows correspond to successive iterations and cells with either 0 (zero) or 1 (one). For hypotheses, 0 corresponds to an unknown status while 1 corresponds to a valid status. For knowledge cells, 0 corresponds to the absence of additional knowledge, and 1 is when the knowledge is present. This causal graph can be leaned using algorithms based on *Bayesian Networks* such as *Non-combinatorial Optimization via Trace Exponential and Augmented largRangian for Structure learning (NO TEARS)* used in the python library called *CausalNex* (Zheng et al., 2018). In particular, *CausalNex* gives means to define constraints on the causal graph by adding additional knowledge from experts. For example, in the presented case in this chapter, knowledge-knowledge cause effects were removed.

This causal graph and the derived knowledge are valuable for decision-making and cause analysis because knowing that a particular knowledge will cause a change in the hypothesis status can help make decisions with respect to this hypothesis in becoming valid. From the illustration of section 3.3.6 the table below is obtained.
The causal graph obtained from the data in Table 8 is shown in Figure 20. This figure

Iterations	h11	h12	h13	h21	h31	h32	k1	k2	k3
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0
2	0	0	0	0	0	0	1	1	0
3	0	1	1	1	0	1	1	1	1

Table 8: Hypotheses validity based on available knowledge

is learned on the causal data generated after an expertise process.

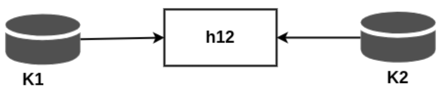


Figure 20: Causal graph showing that hypotheses h12 depends on knowledge k1 and k2

Figure 20 describes that knowledge expressed at iterations one and two are the cause of hypothesis $h_{1,2}$ validation: $\{k_1, k_2\} \rightarrow h_{1,2}$
In other words, this relation is teaching us that

"Training operators to measure KW831 components" will permit "KW831 parts to be in accordance with manufacturing plans".

In general:

If in a causal graph a hypothesis $h_{i,j}$ is validated because of a knowledge $\{k_i\}_{i\in\mathbb{N}}$ then $h_{i,j}$ can be avoided in the falseness of $\{k_i\}_{i\in\mathbb{N}}$

The falseness of knowledge ensures that this knowledge is not observed.

3.3.7.3 *Semantic explanation learning*

Another important knowledge inferred from a HEG emanates from the graph and its semantics mapping. Valid hypotheses of a HEG are mapped with concepts from hypotheses taxonomy (E. g.Figure 17).
From this mapping, one can learn that the likely causes of a problem are the labeled concepts of valid hypotheses of the HEG. The number of occurrences of a given concept even comforts more its likelihood.
For example, from the problem described at section 3.3.6, conclusion is that the problem was caused by humans who are *Operators* and *Measurement* problem. Figure 21 is the graph learned from the example illustrated earlier. From this graph, one can learn that most rejection problems may have been caused by operators and very less by measurement problems.
Mainly, it will be acceptable to prioritize human causes because it has more occurrences (three) compared to the measurement that is only one. Figure 22 shows the last iteration and the label of all hypotheses of the graph obtained from the example.

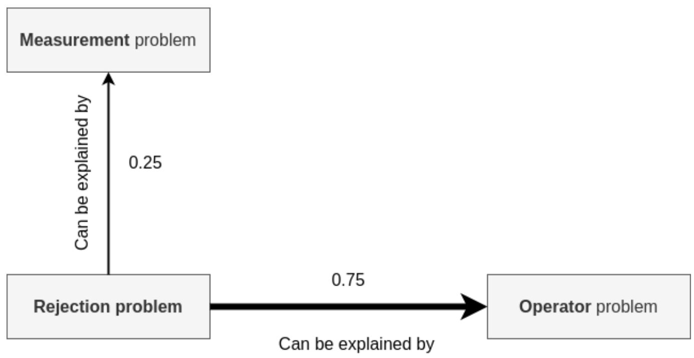


Figure 21: Semantic explanation learning

A numeric evaluation for these concepts in the case of a single HEG is given by the following formula:

$$CE(C_i) = \frac{n_{c_i}}{N_c} \tag{17}$$

, where n_{c_i} is the number of occurrences of concept c_i in the HEG and N_c is the total number of identified concepts occurring in the HEG.

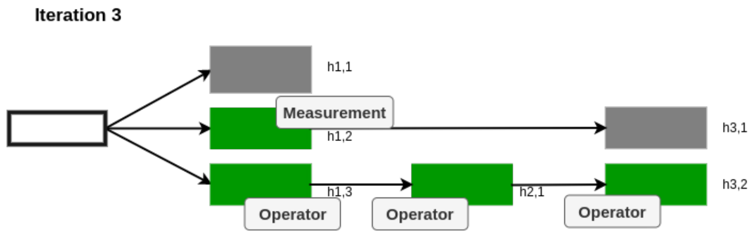


Figure 22: A labeled HEG using the proposed hypotheses taxonomy

In the case of accumulated Expertise Processes from the same problem, the concept evaluation will be given by:

$$CE = \frac{\sum_{j=1}^{g}(\frac{n_{c_i}^{j}}{N_{cj}})}{\sum N_{cj}} \tag{18}$$

, where $1 \leq j \leq g$ is the number of HEG having the same problem,
N_{cj} the total concepts at graph j,
N_c^j is the total concepts of each HEG.
The semantic explanation learning graph can be considered as a projection of a HEG. This newly learned graph encodes human experts' implicit knowledge and confirms common assumptions about a field of study. The graph is different from lexical (WordNet) or factual knowledge graphs (Wikidata). As a result, the learned graph can be considered as a *commonsens knowledge graph* and most importantly, it can be used to solve problems in same domain as the expertise (Zang et al., 2013).

Figure 23 describes the learning cycle based on the HEG, where lessons are learned from HEG analysis, knowledge-hypothesis causal structure, and semantic mapping. These lessons can be used for decision-making, accelerate expertise in urgent cases, or when there is a high time constraint. They can also be used to facilitate future Expertise Processes.

3.4 CONCLUSION

Motivated by the desire to bring together experts' learning skills, experience, and reasoning capabilities on the one hand and machine's computational speed and artificial intelligence, on the other hand, to the domain of Expertise Processes, this chapter proposes a foundation for acquisition, representation, and reasoning with hypotheses in collaborative Expertise Processes.
To describe the proposed approach, this chapter started by reviewing hypothetical reasoning for problem-solving and human-machine collaboration techniques, then presented a framework based on an extended hypotheses theory with experts' doubt, their collective exploration to understand a problem, and an elaborated iterative process. A case from a manufacturing company where one of its products was rejected by customers illustrates the application of the proposed study.

The methodology presented in this chapter takes great advantage of the human cognition system and defeasible reasoning by applying the Hypotheses Theory. As a result,

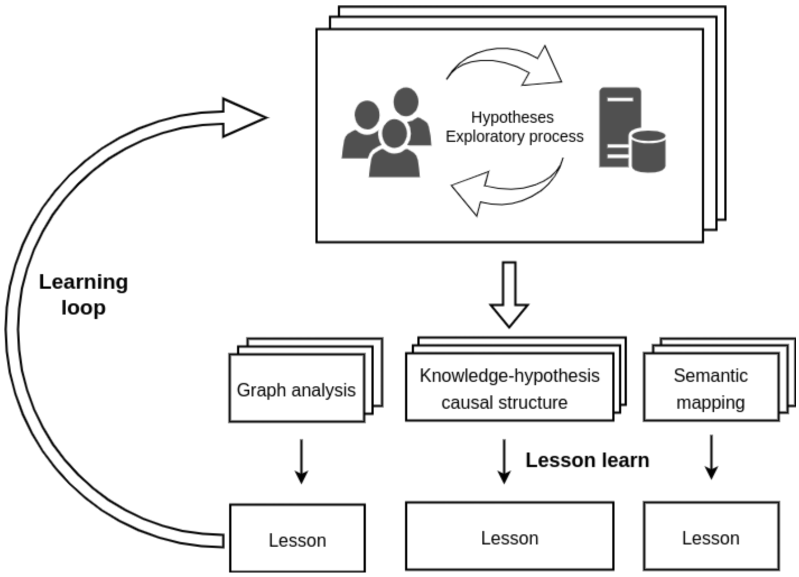


Figure 23: Learning from the HEG

it produces an exploratory graph of hypotheses (HEG), which embeds knowledge and describes the Expertise Process while considering experts' doubt. In addition, this approach gives means firstly to objectively evaluate expertise processes by use of metrics based on the *number hypotheses*, *valid hypotheses rate*, *number iteration* and *valid iteration rate*. Secondly, to monitor the expertise process, using the defined *hypotheses-validation graph*. These tools are relevant for decision-making in the context of the Expertise Process. Finally, the proposed approach can derive both a hypotheses-knowledge causality and a semantic causal graph that experts can use for preventive decisions and upcoming expertise.

Moreover, this foundation of Expertise Processes based on hypotheses is domain independent and can be used in various sectors like railway, automotive, maritime, or construction industries, to name a few. Its usage will permit experience sharing and increase efficiency in expertise while reducing errors, time, and financial expenses. As a result, the proposed approach allows a formal and explicit description of Expertise Processes while facilitating experts' implications in a collaborative environment under limited knowledge.

In the chapters that will follow, studies will investigate, on the one hand, how to integrate experts' beliefs into logical reasoning and, on the other hand, the construction of an ontology as a means to offer more reasoning compared to the taxonomy presented earlier in this chapter for HEG semantic mapping.

4 EXPERTS' BELIEFS AND REASONING

4.1 INTRODUCTION

Companies regularly face problems that reduce productivity when carrying out their main activities. Most of the time, they want to consider experts' certitude in their proposals of solutions. As a result, they seek skilled and experienced people in the field at hand to guarantee reliable solutions. They call for experts because they are effective in problem-solving using their practical knowledge and experience (Shaw and Gaines, 2005) to reduce the level of the doubt in proposed solutions.

In a context that requires experts to propose solutions with some doubts, this chapter proposes an approach to assist them in carrying out this activity efficiently and systematically when there is uncertainty owing to limited knowledge. This approach mixes two required characteristics: first, the ability to use defeasible or non-monotonous reasoning to reason over limited knowledge, and second, to explicitly integrate the doubt of experts into the available knowledge. For the first aspect, answer set programming (ASP) attracted our attention because it appears to be close to the reasoning used in Expertise Processes. ASP is a systematic combinatory method based on predicate logic; however, unlike other inference mechanisms, it uses defeasible reasoning and searches for all possible solutions to a problem. For the second aspect, to address experts' preferences and doubts, this study uses the Dempster Shafer Theory (DST), which subsumes traditional and Bayesian probability in such a way that it manages ignorance and uncertainty using evidence and a well-defined combination rule for merging beliefs (Lu and He, 2017).

In addition, these methods have been implemented as software libraries which are extremely useful in the automation of problem-solving or solving complex problems. However, these systematic tools and ASP, in particular, lack procedures that consider human experts' experience for the possible solutions they propose.
In fact, human experts possess essential skills such as common-sense and know-how, often referred to as tacit knowledge, which is still challenging to encode in machine-readable formats but are used intensively during complex problem-solving (Baporikar, 2020).
It is clear that supporting ASP with human experts' points of view could make it more reliable and will guide users towards realistic solutions partially drawn from experts' intuitions, common sense, or experience. This result emanating from an artificial intelligence system under human control is an essential element in the new vision of AI, in which humans and intelligent systems interact to complement each other's strengths and provide more trustworthy results (Riedl, 2019).
This study aims to create a systematic methodology for expertise that couples human experts' experience with the ASP method and DST. It extends ASP by using expert beliefs and uses DST to define preferences for models that are all solutions to the

problem described in ASP, obtained from its solver. In other words, this procedure sets the metric based on experts' beliefs from which some ASP models will be preferable over others.

The remainder of this chapter is structured as follows. First, section 4.2 presents the background knowledge used in defining the approach and a literature review on methods that combine logic and uncertainty. ASP and DST are presented in this section. Second, section 4.3 describes how to consider experts' beliefs regarding answer sets. An illustrative example follows this section in section 4.4. Finally, section 4.5 describes the characteristics of the approach compared with existing methods before presenting the conclusion and envisioning future directions.

4.2 LOGIC PROGRAMMING AND UNCERTAINTY MANAGEMENT

To describe the proposed procedure, ASP and DST which are the fundamental building blocks of the method, are first presented. Subsequently, existing methods that integrate logic and uncertainty are presented.

4.2.1 *Answer Set Programming (ASP)*

ASP is a declarative programming paradigm that consists in describing a problem in terms of logical rules, constraints, and facts, searching for all possible solution sets that are also known as models, and using a solver that relies on a guess and check procedure (Fandinno and Schulz, 2019; Gebser et al., 2012). This reasoning approach is based on instantiated atoms (without variable), and stable model semantics (see 4.2.1.2), similar to those of constraint satisfaction programming used in inductive reasoning. In fact, these characteristics favor the use of ASP in the domains of knowledge representation and reasoning. In particular, it has been used to tackle problems in domains like product configuration, decision support, music composition, and team building, to name but a few (Dodaro and Maratea, 2017; Gebser et al., 2011). Moreover, its mechanism is suitable for solving optimization problems and derives its roots from logic programming, and non-monotonous reasoning (Janssen et al., 2012). Concerning this solving mechanism, its non-monotonicity is achieved in the form of negation, called *negation as failure* or *default negation* (Kakas, 1994), making it suitable for common-sense reasoning, which is a way of reasoning similar to human reasoning during expertise processes with the ability to retract conclusions, based on new knowledge.
These characteristics make ASP a good tool for exploratory problem resolution, such as expertise processes, because additional information and knowledge is acquired as the task proceeds, and hypotheses expressed by experts can be retracted along the way.
This section presents two main components to better understand ASP: the language syntax and the semantics .

4.2.1.1 *ASP language syntax*

For its syntax, ASP can be expressed in propositional logic as well as first-order logic (Riguzzi, 2018), which makes it simpler and easier to define constraints compared to other programs. In general, the core ASP language includes the following representations (Gebser et al., 2012):

- Normal rules
  A rule r is in the form:

$$a_0 \leftarrow b_1, \ldots, b_m, notb_{m+1}, \ldots, notb_n, \tag{19}$$

  where $0 \leq m \leq n$ and a_0, b_i an atom with $1 \leq i \leq n$
  The atom $\{a_0\}$ is called the *head* of the rule, and is denoted by $head(r)$.
  The set $\{b_1, \ldots, b_m, notb_{m+1}, \ldots, notb_n\}$ is called the *body* and is denoted by $body(r)$. This body can be divided into two parts as follows:
  The set of atoms without negation: $body^+(r) = \{b_1, \ldots, b_m\}$
  The set of atoms with negation: $body^-(r) = \{b_{m+1}, \ldots, b_n\}$

- Ruled out rules
  These are rules that remove grounded atoms from the models based on expressed constraints.
  Integrity constraint

$$\leftarrow b_1, \ldots, b_m, notb_{m+1}, \ldots, notb_n, \tag{20}$$

  where $0 \leq m \leq n$ and b_i an atom with $1 \leq i \leq n$
  The atoms satisfying their body literals are removed from the models.

- Ruled in rules
  These rules add grounded atoms that correspond to predicates without variables to the stable models based on expressed constraints.

Cardinality rule

$$a_0 \leftarrow l\{b_1; \ldots; b_m; notb_{m+1}; \ldots; notb_n\}, \tag{21}$$

where $0 \leq m \leq n$, $l \in \mathbf{N}$, $l \leq n$ and a_0, b_i are atoms with $1 \leq i \leq n$
It adds a_0 to a model if the l cardinal subset of its body is contained in the model; where l represents the lower bound of the cardinality rule.

Generalized cardinality rule

$$a_0 \leftarrow l\{b_1; \ldots; b_m; notb_{m+1}; \ldots; notb_n\}u, \tag{22}$$

where $0 \leq m \leq n$, $l, u \in \mathbf{N}$ and a_0, b_i are atoms with $1 \leq i \leq n$, identical to *cardinality rule* except that it has an upper bound.

The synoptic example of car expertise in Figure 24 shows how these rules affect the answer set.
Figure 24 is the example program without any rule.
Figure 24b is the same example when the cardinality rule is triggered.
Figure 24c is the same example program with its cardinality rule, which was not triggered.

```
slowstart(p19):- batteriesindicator(p19).
batteriesindicator(p19).

Answer: 1
slowstart(p19) batteriesindicator(p19)
```

(a) without cardinality rule

```
batteriesbad(p19):- 1{batteriescorrosion(p19);
                        batteriesindicator(p19)}2.
slowstart(p19):- batteriesindicator(p19).
batteriesindicator(p19).

Answer: 1
batteriesindicator(p19) batteriesbad(p19) slowstart(p19)
```

(b) with activated cardinality rule

```
batteriesbad(p19):- 1{batteriescorrosion(p19);
                      batteriesindicator(p19)}2.
slowstart(p19):- thickoil(p19).
thickoil(p19).

Answer: 1
thickoil(p19) slowstart(p19)
```

(c) With inactivated cardinality rule

Figure 24: Cardinality rule example

From the examples (Figure 24), it can be observed that **batterybad(p19)** is added to the answer (case 24b) because at least one of the bodies of its cardinality rule (**batteryindicator(p19)**) belongs to the model, whereas it is not the case for 24c, where the cardinality rule is not activated and 24a that does not follow this rule.

Choice rule

$$l\{a_1;\ldots;a_m;nota_{m+1};\ldots;nota_n\}u \leftarrow b_1,\ldots,b_m,$$
$$notb_{m+1},\ldots,notb_n,$$

where $0 \leq m \leq n$ and a_i, b_j an atom with $1 \leq i,j \leq n$
If the body of this rule holds, a minimum of l and a maximum of u subsets of the head are added to the stable models.

Figure 25 illustrates this with a simple example, which shows that if the car **p19** has difficulty starting, then it is either due to a bad fuel pump (*fuelpumpbad(p19)*, low fuel (*lowfuel(p19)*), or both.

```
slowstart(p19).

Answer: 1
slowstart(p19)
```

(a) without choice rule

```
1{lowfuel(p19); fuelpumpbad(p19)}2:- slowstart(p19).
slowstart(p19).

Answer: 1
slowstart(p19) lowfuel(p19)
Answer: 2
slowstart(p19) fuelpumpbad(p19)
Answer: 3
slowstart(p19) fuelpumpbad(p19) lowfuel(p19)
```

(b) With choice rule

Figure 25: Choice rule example

From Figure 25b, with the choice rule, one can experience an increased number of answer sets containing all the possibilities for the choice rule.

4.2.1.2 *ASP semantics*

ASP semantics is based on *stable model* semantics, which define how to compute the solution sets of a program, also called its models. In simple terms, this semantic derives sets of grounded atoms that satisfy the rules of the program. This process is performed using an ASP solver after the grounding phase of the problem.
Typically, a set S of grounded atoms is a model of program P if:
$head(r) \in S$,
whenever $body^+(r) \subseteq S$ and $body^-(r) \cap S = \emptyset$
for all rules r of program P (Gebser et al., 2012).
As this procedure does not capture the important property of the *minimal set* (with respect to $\subseteq$), it was redefined in two steps:

1. The **Gelfond-Lifschitz reduction** which transforms an ASP program to a definite program from a given set of grounded atoms. This reduction is as follows: Given a set S of grounded atoms and grounded program P, it transforms P into a definite program as follows:
   - Deleting all the rules in P which have negative atoms appearing in S
   - Removing all Negation as Failure (NAF)-literals in the body of remaining rules. NAF-literals are those preceded by default negation "not".

   The resulting program is denoted P^S
   $P^S = \{head(r) \leftarrow body^+(r)/r \in P, body^-(r) \cap S = \emptyset\}$

2. Deduction
   The deduction of a grounded program P, denoted by $Cl(P)$, is the set of grounded atoms that are consequently deduced from P.

In conclusion, a set S of grounded atoms is a stable model of program P if and only if $S = Cl(P^S)$. Moreover, this model is minimal.
Let us point out that, stable model semantics have non-monotonic characteristics and are quite different from those of classical logic programming languages, such as Prolog, which are goal-oriented and based on backward chaining query evaluations (Niemelä, 1999).

4.2.2 *Logic and uncertainty*

First, to the best of our knowledge, research communities have poorly addressed the Expertise Process, as described in the introduction. In fact, most researchers look at expertise as a specific type of knowledge acquired during activities over a long period of time, which makes a person highly effective in performing a task compared with those who have not acquired this knowledge. (Barley, Treem, and Leonardi, 2020) showed with an application in the health sector that even for domains like coordinating experts, this specific knowledge is essential for good coordination.

In general, approaches that combine logic and uncertainty models, such as probability or possibility, are carried out similarly to those presented at section 2.4.3, that is, by assigning an uncertainty value to logic sentences. However, these approaches have limitations when managing conflicting evidence, or intervals (Núñez et al., 2018). These authors overcame these limitations by extending first-order logic formulas with DST and by assigning an uncertainty interval corresponding to the support one has for a formula being true or false.
The lower bound of the interval for a given domain quantifies the mass of belief that one has for a formula to be true, the upper bound accounts for its plausibility of it being true, and the difference between these two values quantifies the ignorance one has of a formula. After assigning the mass, the mass fusion method was defined to systematically combine the masses for logical operations.

In summary, methods combining logic or logic programming and uncertainty are based on the same principle, which comprises either quantifying the truth of formulas or the belief in their truthfulness. However, they do not describe how these quantities were obtained, either systematically or manually. In addition, these studies do not consider human expectations regarding the result of the problem being computed by logical reasoning.
In particular, the fact that we could not find studies related to expertise processes as activities for problem-solving in the context of limited or lack of knowledge drew our attention.

4.3 EXPERTS' BELIEFS ON ANSWER SETS

This chapter proposes a procedure that takes advantage of both theories within the domain of expertise. Its design is based on (1) knowledge representation and reasoning with ASP, which is a widely used language for reasoning under a lack of knowledge, and (2) the DST for reasoning under uncertainty, from which it is derived quantitative uncertainties from experts' qualitative preferences. This will be done by extending the approach presented by (Malo et al., 2013) from a single expert to multiple experts and (3) a novel mechanism that integrates experts' beliefs and ASP.

Starting from a problem encoded in ASP, the proposed approach will enable experts to collaborate on the same expertise and to select the best answer set from ASP based on their background knowledge beliefs.

Let P be a problem encoded in ASP, for which experts' preferences on its answer sets have to be considered.
Let H_P denote the Herbrand base of P, which corresponds to the set of all ground atoms obtained from predicates appearing in P (Lloyd, 2012) and $Ans(P)$ its answer sets.
Let $E_1, E_2, \ldots, E_n$, $n \in \mathbf{N}$ denote the n experts involved in solving problem P and who have experience in solving problems similar to P.
The method proposed has the following steps:

- **Step 1**: Experts' preferences elicitation
  This step consists in obtaining the experts' preferences based on their beliefs from the Herbrand base H_P of the problem P to be solved. These subsets of H_P correspond to experts' expectations of what could be solutions to the problem based on their experience and current observation of this problem.
  Therefore, they use their past knowledge and know-how to select distinct preference subsets from the Herbrand base H_P:
  - If experts have different preferences:
    Expert E_1: $\{F_1^1, F_2^1, \ldots, F_{m_1}^1\}$
    Expert E_2: $\{F_1^2, F_2^2, \ldots, F_{m_2}^2\}$
    ...
    Expert E_n: $\{F_1^n, F_2^n, \ldots, F_{m_n}^n\}$, $F_j^i \subseteq H_P$,
    $F_j^i \neq F_k^i$, $1 \leq j \leq m_i$ and $j \neq k$
    $\forall_{1 \leq j \leq m_i} F_j^i$, meaning that each expert preference set is distinct.
    F_j^i are distinct preference subsets obtained iteratively from experts.
  - If experts have the same preferences: this means they consensually agree on the same subsets of preferences.
    In this case, just one set of preferred subsets will be used.
- **Step 2**: Preference frame of discernment
  After selecting their preferences, what remains is the least preferred set (B^i).
  Let us denote them by $\{2^{H_P} \setminus E_i\} = B^i$,
  which constitutes the least preferred candidate.

From the previous steps, the new frame can be defined as follows:
$\Theta = \{F_j^i, \{\cap B^i\}\}_{1 \leq i \leq n}$, which comprises distinct preference sets from all experts and distinct subsets of the Herbrand base H_P. On the contrary, $\{\cap B^i\}$ guarantees that there is no preference set in this set.

- **Step 3**: Quantifying experts' beliefs
  This step of the procedure comprises assigning masses of evidence to preference sets provided by experts in the previously defined FOD.
  The basic belief assignments from qualitative to quantitative values are defined utilizing the methodology described in (Malo et al., 2013). The approach consists of eliminating the least credible hypotheses iteratively from the FOD and providing evidence based on how easy it was to choose the least likely solutions. The value of how easy it was to select these sets ranged from 1 to 10, with 1 indicating that the easiest effort was needed to select a given set.

  If $N = count(F_j^i)_{1 \leq i \leq n}$ is the number of preferences expressed by expert E_i, then he/she will have no more than $N - 1$ iterations to define his/her evidence.
  For each iteration, a unique subset $S_i \subseteq \Theta, i \leq N$ is selected by an expert based on their experience with the problem being solved.
  For the ease of selection of this subset, the value given by the expert is shared as a mass between the remaining set and frame:
  If the given value is $\alpha \in 1$ to 10, then $m_i(\{\Theta \setminus S_i\}) = \frac{10-\alpha}{10}$,
  $m_i(\Theta) = \frac{\alpha}{10}$
  From these basic assignments, the mass distribution function $m()$ of FOD Θ for all experts using the DST combination rule can be computed.

- **Step 4**: Compute beliefs on answer sets
  This last step is as follows. First, answer sets are obtained from the ASP. Second, the beliefs of subsets of preference sets and, hence, those of answer sets are evaluated.
  The belief of answer sets is computed from the belief of experts' preference sets as follows:
  If $A \in Ans(P)$ is an answer set, then:

$$Bel(A) = \sum_{\{F_j^i\} \subseteq A} m(\{F_j^i\}) \tag{23}$$

$$Pl(A) = \sum_{\{F_j^i\} \cap A \neq \emptyset} m(\{F_j^i\}) \tag{24}$$

  where $\{F_j^i\}$ is a sub-set of preferences.

The procedure and its different steps are shown in Figure 26.

It should be noted that experts' experiences have an impact on the model's beliefs in various ways. If experts have the same experience and choose sets closer and similar

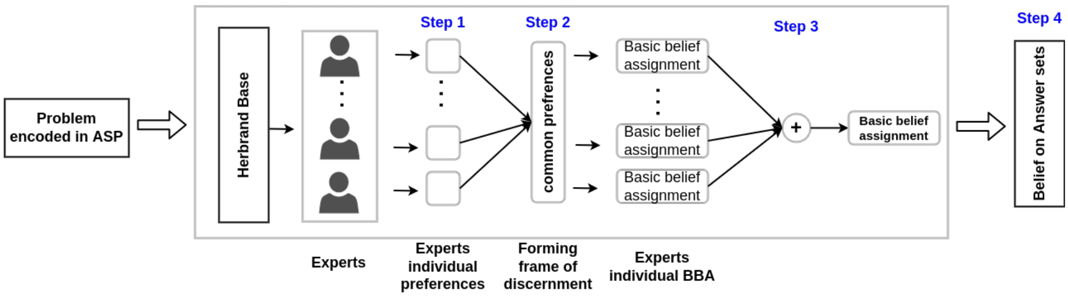


Figure 26: Experts' beliefs on answer sets elicitation process

to the models, then the beliefs of these models computed by the approach should be close to their expectations.
On the contrary, if experts have entirely different experiences, which can be illustrated by the fact that they have opposing views on the problem at hand, the model's belief will be affected by a firm belief in the entire frame expressing total ignorance or, on the empty set, because the intersection of their choices will be empty, which is a consequence of the combination rule.

4.4 ILLUSTRATION

This illustration shows how the elaborated method works for a given problem through the steps of the proposed procedure.
The problem of this use case is described as follows:

- Knowledge of car start:
  If it is not believed that the fuel pump is bad, and it is believed that there is fuel, and it is believed that the battery is less than ten (10) months old, then it is believed the car will start.

- Knowledge of car slow start:
  If it is not believed that the car battery is bad, and it is believed that the battery is more than 12 months old and that the car oil is thick, then it is believed the car has a slow start.

- Knowledge of bad fuel pump:
  If it is believed that there is heat in the car at a certain temperature, sputter, and noise from the tank of the car, and fuel pressure is less than 30 PSI, then it is believed that the fuel pump is bad.

- Knowledge of bad batteries:
  If it is believed that the car has a slow start and the dash light is on, and there is corrosion on the battery and the battery indicator signals when driving, then it is believed that this battery is bad.

- If it is believed that the car has a slow start, then the problem could be due to a bad battery, low fuel, or bad fuel pump.

The ASP program associated with this problem is shown in Code 10, with a car **p19**, which has some problems.

This example considers that two experts (Expert1 and Expert2) are involved in expertise.
After encoding the problem in ASP, its Herbrand base is derived. The Herbrand base for problem P is:
$H_P = \{battery(p19, 15), fuelpumpbad(p19),$
$lowfuel(p19), thickoil(p19), batterybad(p19),$
$slowstart(p19), pressure(p19, 20), sputter(p19),$
$noisetank(p19), heat(p19, 50),$
$batteryindicator(p19), batterycorrosion(p19),$
$dashlight(p19)\}$

- **Step 1**: Experts' preferences
  The preferences of two experts (Expert1, Expert2) from their experience in the domain of car diagnostics are expressed as follows:
  - Expert 1
    $F_1^1 = \{batterybad(p19), batterycorrosion(p19)\},$
    $F_2^1 = \{lowfuel(p19)\}$
  - Expert 2
    $F_1^2 = \{fuelpumpbad(p19), thickoil(p19),$
    $noisetank(p19)\},$
    $F_2^2 = \{lowfuel(p19)\}$

  Letters A_1, A_2, and A_3 for sets F_1^1, F_1^2, and F_2^1, stand respectively for
  $A_1 = \{batterybad(p19), batterycorrosion(p19)\}$
  $A_2 = \{fuelpumpbad(p19), thickoil(p19),$
  $noisetank(p19)\}$
  $A_3 = \{lowfuel(p19)\}$

- **Step 2**: Preference frame of discernment
  From the preferences of experts:
  $B^1 = 2^{H_P} \setminus \{A_1, A_3\}$ from Expert 1
  $B^2 = 2^{H_P} \setminus \{A_2, A_3\}$ from Expert 2
  Putting it all together, the FOD is:
  $\Theta = \{A_1, A_2, A_3, \{B^1 \cap B^2\}\}$

- **Step 3**: Quantifying experts' beliefs
  After expressing their preferences, the two experts' distributions of evidence on the FOD are listed in table 9, and 10 respectively.

  From Tables 9 and 10, Tables 11 and 12 are obtained as evidence distributions for Experts 1 and 2, respectively:

Code 10: ASP encoding of car expertise

```
battery(p19, 15).
% p19 is a car
% Car p19 has a 15 month battery

thickoil(p19).
% Car p19 has thick engine oil

sputter(p19).
% Car p19 is sputtering

dashlight(p19).

pressure(p19, 20).
% p19 pressure is 20 PSI

heat(p19, 50).
% Car p19 heats at 50 degree

lowfuel(p19).
% Car p19 has low fuel

noisetank(p19).
% Car p19 has a noisy tank

batterycorrosion(p19).
% Car p19 has corrosion on its batteries

batteryindicator(p19).
% Car p19 board is indicating low batteries

slowstart(p19).
% Car p19 has a slow start

quickstart(X) :- battery(X, Month ), not fuelpumpbad(X), not lowfuel(X),Month < 10.
% Knowledge of car start

slowstart(X) :- battery(X, Month ), thickoil(X), not batterybad(X), Month > 12.
% Knowledge of slow car starts

fuelpumpbad(X) :- heat(X, Temperature), noisetank(X), sputter(X),pressure(X, Pressure),
                    Pressure < 30, Temperature > 60.
%Knowledge of bad fuel pump

batterybad(X) :- dashlight(X),batterycorrosion(X), batteryindicator(X).
%Knowledge of bad batteries

1{batterybad(X); lowfuel(X); fuelpumpbad(X)}3:- slowstart(X).
```

Expert 1	Mass distribution
Which of the preference sets is the least trustworthy ? Expert's answer: $\{B\}$ How easy was it to remove $\{B\}$? Expert's answer: 8	After removing $\{B\}$ from the preference set, evidence is shared among $\{A_1, A_2, A_3\}$ and $\{\Theta\}$ m1($\{A_1, A_2, A_3\}$) = 0.2 m1($\{\Theta\}$) = 0.8
Which of the preference sets is the least trustworthy ? Expert's answer: $\{A_2, B\}$ How easy was it to remove $\{A_2, B\}$? Expert's answer: 6	After removing $\{A_2, B\}$ from the preference set, evidence is shared among $\{A_1, A_3\}$ and $\{\Theta\}$ m2($\{A_1, A_3\}$) = 0.4 m2($\{\Theta\}$) = 0.6
Which of the preference sets is the least trustworthy ? Expert's answer: $\{A_3, A_2, B\}$ How easy was it to remove $\{A_3, A_2, B\}$? Expert's answer: 4	After removing $\{A_3, A_2, B\}$ from the preference set, evidence is shared among $\{A_1\}$ and $\{\Theta\}$ m3($\{A_1\}$) = 0.6 m3($\{\Theta\}$) = 0.4

Table 9: Basic belief distribution applied by Expert 1

Expert 2	Mass distribution
Which of the preference sets is the least trustworthy ? Expert's answer: $\{B\}$ How easy was it to remove $\{B\}$? Expert's answer: 9	After removing $\{B\}$ from the preference set, evidence is shared among $\{A_1, A_2, A_3\}$ and $\{\Theta\}$ m1($\{A_1, A_2, A_3\}$) = 0.1 m1($\{\Theta\}$) = 0.9
Which of the preference sets is the least trustworthy ? Expert's answer: $\{A_2, B\}$ How easy was it to remove $\{A_2, B\}$? Expert's answer: 6	After removing $\{A_2, B\}$ from the preference set, evidence is shared among $\{A_1, A_3\}$ and $\{\Theta\}$ m2($\{A_1, A_3\}$) = 0.4 m2($\{\Theta\}$) = 0.6

Table 10: Basic belief distribution applied by Expert 2

$m(\{A_1\})$	$m(\{A_1, A_3\})$	$m(\{A_1, A_2, A_3\})$	$m(\{\Theta\})$
0.6	0.16	0.04	0.2

Table 11: Final distribution of masses for Expert 1

$m(\{A_1, A_3\})$	$m(\{A_1, A_2, A_3\})$	$m(\{\Theta\})$
0.4	0.06	0.54

Table 12: Final distribution of masses for Expert 2

Using the DST combination rule, these experts' distinct distributions were combined into a single distribution of evidence, as shown in Table 13

$m(\{A_1\})$	$m(\{A_1, A_3\})$	$m(\{A_1, A_2, A_3\})$	$m(\{\Theta\})$
0.6	0.25	0.036	0.1

Table 13: Final distribution of masses

- **Step 4**: Compute beliefs on answer sets
  Now let us compute the answer set of the problem:
  Answer: 1
  $Ans1 = battery(p19, 15), lowfuel(p19),$
  $thickoil(p19), batterybad(p19),$
  $slowstart(p19), pressure(p19, 20),$
  $sputter(p19), noisetank(p19),$
  $heat(p19, 50), batteryindicator(p19),$
  $batterycorrosion(p19), dashlight(p19)$

  Answer: 2
  $Ans2 = battery(p19, 15), lowfuel(p19), thickoil(p19),$
  $batterybad(p19), slowstart(p19),$
  $pressure(p19, 20), sputter(p19),$
  $noisetank(p19), heat(p19, 50),$
  $batteryindicator(p19), batterycorrosion(p19)$
  $dashlight(p19), fuelpumpbad(p19)$

  The last task of the procedure is to evaluate their beliefs:
  - Answer set $Ans1$
    Among the sets on which there are distribution of masses, the largest that covers $Ans1$ is $\{A_1, A_3\}$ which means that:
    $Bel(Ans1) = 0.6 + 0.25 = 0.85$

  - Answer set $Ans2$
    $Bel(Ans2) = 0.6 + 0.25 + 0.036 = 0.89$

  Therefore, from the experts' experience in this case of car expertise, it would be preferable to choose **Ans1** over *Ans2*.

  There are high and close values of beliefs from this example because, on the one hand, experts' belief sets were close in terms of elements they contained, and, on the other hand, these sets were close to the answer sets. On the contrary, if experts had expressed significantly different belief sets, which in addition had fewer elements in common with the answer sets, one would have expected low beliefs.

From the above, the following algorithm 2 is proposed to easily determine the answer set with the greatest belief:

Algorithm 2: Find model with the highest belief

Result: Model with highest belief
Encode problem in ASP ;
Get Herbrand base;
for *each expert* **do**
  Define preferences from Herbrand base;
end
Form common frame of discernment;
for *each expert* **do**
  Elicit evidence distribution;
end
Combine mass of evidence;
Compute answer sets;
Compute answer sets' beliefs;
Return model with the highest belief;

4.5 DISCUSSION

The approach described and illustrated in the previous sections extends and covers certain limits of human and systematic reasoning.
On the one hand, it combines both the ASP methodology for problem-solving and human experts' beliefs and experiences obtained from their past activities. It makes use of both strengths and provides beliefs on answer sets, which cannot be obtained by each party individually.
On the other hand, the DST alone cannot be applied in this context of exploratory problem-solving because it does not provide any mechanism to search for all solutions to a problem. In essence, the strengths of both theories are combined, and Table 14 below summarizes the strengths of the proposed procedure over the two other methods considered individually.
Another important aspect of this approach is its ability to accept divergent viewpoints

Approach	Solution to problem	Belief on sets
ASP	yes	no
Experts/DST	no	yes
Proposed approach	**yes**	**yes**

Table 14: Contributions of experts' beliefs on answer sets

from experts working on the same problem, which means more evidence for possible answer sets. In fact, having many experts working together does not demean the result; however, it has a positive effect on the procedure because it increases the evidence distribution space.

Furthermore, the proposed means of collaboration are pretty different from most human-machine collaborations. Most human-machine collaborations, such as robot collaboration or virtual agents, are based on interactions between humans and machines through natural languages, voice, and gestures to ensure safety and teamwork for high productivity (Sowa, Przegalinska, and Ciechanowski, 2021). Other collaborative systems are based on third parties, such as augmented reality (AR) software or components working on task allocation and adaptive control between humans and machines, to ameliorate their interaction with physical tasks, which in most cases cannot be fully automated (Baroroh, Chu, and Wang, 2020; Bettoni et al., 2020). Conversely, this chapter proposes a system for task reasoning in which humans actively contribute to the knowledge needed to produce the final result. Undoubtedly, without human experts' experience and beliefs, it will not be possible to compute the selection of models produced by ASP using the proposed method.

This chapter used DST simple support function to compute beliefs on answer sets, which is not as specific as mass distributions because only one proper subset of the frame is provided with evidence. Moreover, this method does not consider experts' collaboration, whereas the approach presented here provides a means for that. Thus, the proposed approach is a more accurate and collaboratively usable method in the context of Expertise Processes.

Another attempt for ASP and DST hybridization is the method presented in (Al Machot, Mayr, and Ranasinghe, 2018), where the activity recognition approach is represented by ASP, and DST is used to merge data from sensors. Two main differences distinguish this approach from the method proposed in this chapter.

- This approach is purely domain-specific and particularly for activity recognition, while the one of this chapter is domain-independent.
- These authors' mass distribution of evidence is based on past sensors' spatial and temporal data features, whereas the one proposed in this chapter is based on human experts' collaboration and experience.

Working with group decision-making preferences and uncertainty is a study path in which linguistic decision-making excels. This theory is based on human language word computing, and sometimes with uncertainty attached to linguistic information (Xu, 2005, 2012). This model is practical, and many operators have been developed to cover their combination, but applying it in this context is not appropriate because DS evidence theory was not conceived to compute words (Pang, Wang, and Xu, 2016). In addition, designing probabilistic or hesitant fuzzy linguistic approaches (Liao et al., 2018) for a combinatory problem resolution, such as ASP, brings in more complexity. In fact, it is challenging to capture human experts' preferences for a large number of possibilities and aggregate them.

4.6 CONCLUSION

This chapter presents a generalized method that considers human experts' beliefs from their experience on answer sets derived from a problem encoded in the ASP. This is achieved using the DST and a mechanism to compute answer set beliefs.

To describe this procedure, ASP is first presented as a language actively used in knowledge representation that supports a non-monotonic reasoning paradigm. After that, the fundamentals of DST are presented. It is a generalized method for uncertainty representation. Using this, a new procedure to quantify experts' qualitative beliefs using numeric values was presented and used in conjunction with ASP models.

Finally, the method was illustrated with an example in the domain of car expertise, in which models' beliefs were computed considering the experience of two experts collaborating on the same problem.

The discussion section shows how the proposed method differs from existing methods, first, by its general applicability in all domains because it can be applied whenever domain knowledge can be encoded in ASP. Second, it is human-centered because it uses human experts' beliefs, experiences, and the possibility that they can collaborate on the same problem. Third, unlike existing human-machine interactions, the developed approach implements collaborative reasoning, not task collaboration. Finally, the study sets out a method to elucidate expert evidence that is more accurate than the simple support functions proposed by others.

5 EXPERTISE ONTOLOGICAL REPRESENTATION

5.1 INTRODUCTION

This study was motivated firstly by the importance of expertise in companies and society as acknowledged by French and European standard document *NF X50-110*, the *CNS EN 16775 standard* "Expertise activities - General requirements for expertise services". These documents set norms on how expertise should be carried out and the roles of different actors involved in the process.
Secondly, the need of semantic support for Expertise Processes beyond taxonomy as used in Chapter 3 for semantic mapping, the lack of appropriate representation of expertise knowledge and the benefit of making it accessible and reusable by humans and machines. Undoubtedly, expertise helps design safety measures and learn lessons to reduce or avoid risk in the case of accidents. Furthermore, this knowledge is exploited in understanding new problems; therefore, it helps to reduce the burden of this knowledge-intensive task and, as a result, speeds up expertise processes.
Thirdly, the desire to enhance communication among the stakeholders involved in the expertise process, such as financial backers, project sponsors, clients, and experts.
Besides this, experts from various fields who must work together because problems are multi-facet or need different domain knowledge face difficulties understanding each order.
Finally, it is worth using expertise when designing tools such as safety or awareness systems to avoid accidents in a field of activity.

The contribution of this chapter in the field of accident expertise is in particular for the following reasons. First, expertise is a vast field with numerous categories such as accident, risk, and safety. Each category can be further divided into subcategories. For example, the accident domain can be subdivided into aircraft, automobiles, and housing. As a result, collating all the main categories can be a laborious task and can lead to a meaningless result because each category differs from its counterpart in knowledge, vocabulary, and objectives.
Second, accident expertise is relevant because accidents cause important material and human loss. For example, a mutual insurance company in Chile registered 625,050 work accidents from 2015 to 2019 with thousands of deaths and disabilities (Bravo et al., 2022), and the World Health Organization (WHO) counted about 1.25 Million deaths in road accidents (Baskara et al., 2019). Another alarming statistic that motivated the choice of accident expertise is that in Canada, almost 3000 people are killed yearly in road traffic accidents (Wang and Wang, 2011). As accidents are not rare, the proposed design would like to model accident expertise such that sharing or reusing them by humans and systems will be easier. Without a doubt, proper reuse of past accident expertise knowledge can help in saving time, money, and most importantly, human lives.
Third, because accident expertise involves interaction between experts from various fields, there is a need for common understanding. For example, in aircraft accident

expertise, experts from aviation, weather, safety, and mechanical engineering work together to understand what happened. This aspect calls for the need of a common vocabulary and reasoning mechanism, so that experts can understand each other. Finally, accident expertise is a field in need of support because it is a knowledge-intensive activity that must be implemented when there is no clear understanding or knowledge to answer questions about accidents. Nevertheless, their outcomes help support new expertise, building safety systems, learning lessons, or making decisions.

This study's contribution from the observed breaches in accident expertise is an ontology, that is, a formal description, concise vocabulary, precise semantics, and reasoning (Panagiotopoulos et al., 2012). As a result, this study aims to build a Basic Accident Expertise Ontology (BAEO) for accident expertise. First, this ontology will act as a shared vocabulary and knowledge for experts. Secondly, it will act as a base for representing domain-specific accident expertise.
In order to ease the BAEO integration and reuse, the chapter utilize the UML and the Model-Driven Architecture (MDA) that are appreciated in systems designing for their high level of detail representation (De Lope et al., 2021; Malgouyres and Motet, 2006). MDA is an Object Management Group (OMG) standard that uses abstract views called models at four (04) different layers of representation to design systems. The most abstract layer is the meta-meta-model, followed by the meta-model, the third layer is the model, and finally, the instance of the model (Paolone et al., 2020). Figure 27 shows the different layers of the MDA architecture. The meta-meta model is self-defined and

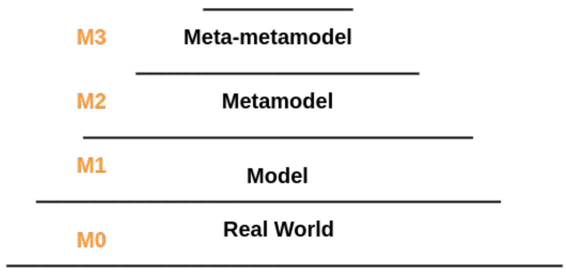


Figure 27: Model Driven Architecture from (Gašević, Djurić, and Devedžić, 2006)

is used to describe the meta-model, while the meta-model is used to express a valid model.

The proposed BAEO presented in this study belongs to the second layer of the ontology hierarchy. This chapter will show its construction in subsequent sections, illustrate its utilization from a case of accident expertise taken from the Bureau d'Ênquetes et d'Analyse (BEA)[1] online database, and demonstrate its reuse.

The next section of the work (section 5.2) presents studies of ontologies related to accident and expertise. After this, section 5.3 describes the design and implementation of the proposed BAEO, followed by an illustration and reuse. The last section before the conclusion is the discussion section 5.5 where differences between BAEO and some accident ontologies are shown.

1 https://bea.aero/

5.2 ONTOLOGY IN THE DOMAIN OF ACCIDENT EXPERTISE

This section presents relevant studies in domains that bridge both ontology and accident expertise.

In the domain of accident scenarios, (Maalel et al., 2012) proposed an ontology to acquire expert knowledge in analysis and safety assessment processes in railroad accidents and incidents. The ontology includes essential knowledge such as contexts, events, and causes but is strictly limited to the domain of railroad accidents. The conceptualization proposed by these authors helps in modeling scenarios that describe actions that lead to dangerous situations.

Another attempt to model expertise knowledge is the study of safety presented by (Kaindl et al., 2016). These authors used an engineering ontology building approach, *ISO 26262* and *EN 50126* standard concepts from the railway and automobile industry to construct ontology for safety. Motivated by the harmonization of safety assessment, they provide a taxonomy containing concepts such as *risks*, *harms*, and *hazards* from the railway domain. This ontology cannot capture accident knowledge such as causes, consequences, or events.

In order to make aviation accident reporting easier within the European Coordination Centre for Accident and Incident Reporting Systems (ECCAIRS), (Křemen et al., 2017) used the terminology of an existing accident reporting information system to construct an ontology for occurrences, events, factors, and aircraft descriptions. The constructed ontology uses Unified Foundational Ontology (UFO) as its base. Because it was intended for reporting, the ontology does not represent the knowledge of cause and consequences, with this limiting its utilization for accident expertise.

In road accidents, tools such as Mivar Expert System (MES) analyze road accidents and determine optimal parameter values for accident simulations. MES reduces experts' difficulties in reconstructing vehicle accidents and accelerates decision-making (Chuvikov et al., 2019). Nevertheless, the final artifact produced by experts is still unexploited and not considered by the Mivar system.

Similarly, a study from (Wu et al., 2020) used ontology in the transport field. The authors coupled ontology and natural language processing (NLP) techniques to describe subway accidents. This description allowed them to retrieve similar cases of accidents to support decision-making when faced with new cases. Ontology built using this approach formalized the semantics of unstructured and semi-structured documents related to subway accidents and regulations, which are not appropriate for our case study.

Another study in the transport sector is the work of (Barrachina et al., 2012b) that used accident information and data from the General Estimates System (GES) to design an ontology for vehicle accidents. The ontology allows sharing, integrating, and reusing knowledge about vehicles involved in road accidents. Likewise, for an interoperability solution, (Barrachina et al., 2012a) proposed an ontology for car accidents through Vehicular Ad hoc Networks (VANETs). Their study defines a shared understanding of a car accident environment such that it can be shared with other vehicles. These ontologies are based on four main concepts: vehicle, accident, occupant, and environment.

In summary, ontologies developed in the above literature have drawbacks for accident expertise despite their contributions to accident-related fields. On the one hand, because these ontologies are domain-specific, it is challenging to use them for different accident expertise knowledge. On the other hand, they only partially capture aspects of accident expertise such as cause-effects knowledge for some and event for others.
This study aims to overcome the above limits using an ontology for accident expertise. Furthermore, it focuses on a base ontology to facilitate the construction and integration of specific domain accident expertise knowledge.

5.3 ONTOLOGY FOR ACCIDENT EXPERTISE

The main problem is the absence of a dedicated ontology for accident expertise knowledge. However, this knowledge is essential for the construction and integration of specific fields of accident expertise, such as railways, aircraft, automobiles, and building construction. To achieve the above mentions, this study elaborated a high-level ontology that can be used to represent accident expertise knowledge from multiple areas. Figure 28 shows the overview architecture to be achieved in this study. BAEO

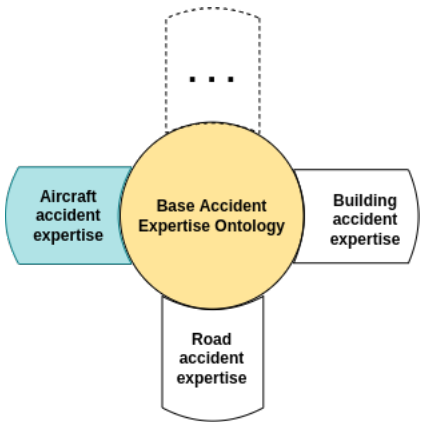


Figure 28: Basic expertise ontology architecture

will stand as a base ontology that can be extended for specific domains of accident expertise.
This study bases the construction of the proposed ontology on a manual approach because, on the one hand, automated techniques can produce redundant and inconsistent conceptualization or even lack semantics due to noisy data (Hur, Janjua, and Ahmed, 2021). On the other hand, it is challenging to access all areas of expertise, and if that was the case, it would lead to the building of isolated ontologies for each area which would be even more challenging.
For this reason, this study will rely on a manual middle-out approach for conceptualizing that starts with essential concepts and move toward a high-level conceptualization. This approach will allow us to (1) reduce the granularity since this work intends to construct a high-level ontology, (2) avoid inconsistencies due to low-level details, (3) have better control over concepts, (4) have more stable models, and (5) have fewer reworks and efforts (Uschold and Gruninger, 1996).
The methodology for ontology construction employed in this study utilizes approaches

described by (Hassan and Mokhtar, 2021; Martínez-García et al., 2020). The following mechanisms were added to the fundamental steps proposed by these authors. First, an iterative and cyclic process is defined. This cycle corresponds to the main steps of the proposed methodology. Second, UML profile and model-driven architecture as tools for the conceptualization were adopted. These tools will enhance integration from specific areas of accident expertise and increase understanding or usability. Finally, semantic web technology was used to make the ontology machine-readable.
In fact, UML profile serves as a bridge between UML and OWL, allowing the semantic web to use UML for designing. Clearly, UML profile is an extension of classic UML with mechanisms such as stereotypes, tag definition, or tagged values, which makes it an ideal tool for designing ontologies (Djurić et al., 2004) and for which there are rules to transform their representations to OWL (Vo and Hoang, 2020).
Table 15 below shows the main differences between UML and OWL (Jetlund, Onstein, and Huang, 2019).
The methodology employed for this study is cyclic and iterative. The following detail

UML	Semantic Web/OWL
Based on closed world assumptions (CWA)	Based on open world assumptions (OWA)
Instances of a class have the same number of properties	Instances of a class have flexible number of properties
Models are defined in a closed environment with less flexibility	Easy and flexible means to link models
Does not have set based principles	Relies on DL and set theory

Table 15: Differences between UML and OWL

describes its phases.

1. Specification:
   This first phase covers the following (1) the scope and specifications of the ontology and (2) competency questions that the knowledge from the ontology will be able to answer.

2. Conceptualization:
   For this phase, the following tasks are to be carried out:
   - The description of the ontology concepts
   - The description of these concepts' relationships
   - The ontology design using UML classes
   - Rules and constraints expression

At this phase, any decision to reuse existing ontologies is made.

3. Formalization:
   One can easily translate the conceptual model into a formal language from the previous phase. (1) For example, the designed ontology can be translated into OWL, and (2) the rules in Semantic Web Rule Language (SWRL)
   SWRL supports OWL for reasoning from an ontology because it can infer knowledge from rules constructed from OWL individuals, classes, properties, and specific built-ins that OWL does not provide. For example, it is possible to build rules with some arithmetic operators with SWRL, while this is not possible with OWL.
   Tools such as protégé (Musen and Team, 2013) can be utilized at this step.

4. Evaluation:
   Ontology evaluation has two main methods, which are validation and verification. They consist in checking on the one hand if the ontology structure is consistent and was designed correctly and on the other hand if the ontology maps well the real world for which it was designed, respects its characteristics, constraints, and semantics (Amirhosseini and Salim, 2019; Brank, Grobelnik, and Mladenic, 2005).

Figure 29 shows the cycle of the methodology for ontology construction used in this study. The construction process iterates on this cycle to produce the desired ontology. This iterative approach helps to refine the ontology as it is constructed.

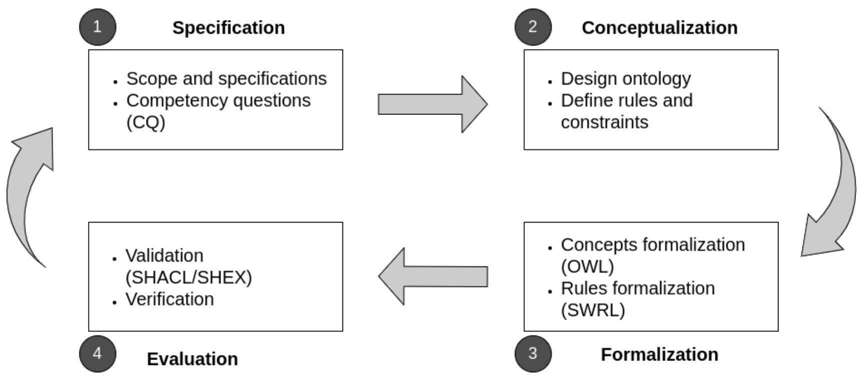


Figure 29: Ontology construction methodology cycle

The sections below illustrate how the BAEO was designed using the abovementioned approach.

5.3.1 *Scope and specification*

This study aims to construct a base ontology to represent accident expertise knowledge. This ontology will be reused in various fields of accident expertise, such as aircraft, automobiles, and railways.
As a result, the proposed ontology must capture high-level knowledge common to accident expertise.

Competency questions for the proposed ontology are the following.

- How did the accident occur?
- When did the accident occur?
- Where did the accident occur?
- Which stakeholders are involved in this accident?
- How many victims were in the accident?
- What were the causes of the accident?
- What is the list of victims of the accident?
- Who are potential witnesses of the accident?
- What are the consequences?
- Which vehicle/equipment/asset/part is involved in the accident?

5.3.2 *Conceptualization*

The following knowledge containers were identified from these competency questions and accident expertise reports, context, activity, cause, and consequence. These knowledge containers cover essential knowledge of accident expertise, without which it will be considered as **incomplete**.
For ontology designing, this work uses the OMG Meta-model Object Facility (MOF), which offers an independent platform framework and facilitates ontology interoperability concerning the MDA (MOF, 2015). UML offers diagrams, extensions, and Object Constraint Language (OCL), that makes it flexible and suitable for modeling ontologies' class/subclass hierarchy, relations, and axioms (Kogut et al., 2002).
The knowledge containers identified earlier are considered the main concepts of accident expertise. They form the building blocks of the proposed base accident expertise ontology (BAEO) taxonomy completed with additional sub-concepts.

- Context
  Context-awareness is an essential aspect of expertise because it describes the surroundings of where the accident took place. It has the same importance as the context in the domain of context-aware computing or services (Cabrera, Franch, and Marco, 2019; Guermah et al., 2014). In other words, a context presents an environment where an accident occurred. For this study, context is designed from concepts found in the work of (Cabrera, Franch, and Marco, 2019).
  This concept includes the sub-classes: agent,resource , ,location and environment.

  - Agent: Corresponds to an actor who actively or passively participates in an accident.
  - Resource: Its instances are any tool involved in an accident. Resource instances are domain-dependent, and specifying them will vary from one accident expertise field to another.

For example, resources such as helicopters or airplanes will belong to aircraft accident expertise, whereas vehicles will be for road accident expertise, and cranes for construction accident expertise.

  – Location: instances of the location concept identify the place of an accident.
  – Environment: Instances of this concept are resources that are not specific to the domain of an accident, such as natural or physical phenomena.

- Activity
  It describes anything that happened in the context of an accident under expertise at a specific length or interval of time. This concept adds dynamic and temporal characteristics to the represented expertise. An activity can have a beginning and end event.

- Consequence
  This concept describes the outcome and damages yielded by an entity in the accident context. In general, these consequences can involve material damage, including human fatality.
  Sub-classes under consequences are disruption, destruction, loss, harm, and failure. These classes of consequence are divided into two groups which are material and living consequences.

- Cause
  These elements are identified as triggers of the consequences of an accident. They are the factors of the accident and are grouped into sub-classes: *humancause* for human causes, *naturalcause* for natural causes like wind, fog, earthquake, or weather conditions, to name a few, and *systemcause* for system causes.

Figure 30 shows the above-named main concepts of the BAEO and the relations that exist between them.
Rule: An expertise is **complete** only if these five knowledge containers are not empty; otherwise, it is considered incomplete.
Rule: The number of victims is equal to the number of agents affected by the accident consequences.
Constraint: An accident expertise must have at least a consequence.
Constraint: An event must have a specific time or period.
Constraint: An activity must have a beginning and ending event or duration.
Constraint: An accident expertise must have a context.
The proposed modeling approach is based on model-driven architecture (MDA) and UML, as shown in Figure 31. This approach uses a UML profile to adapt UML for ontology design.

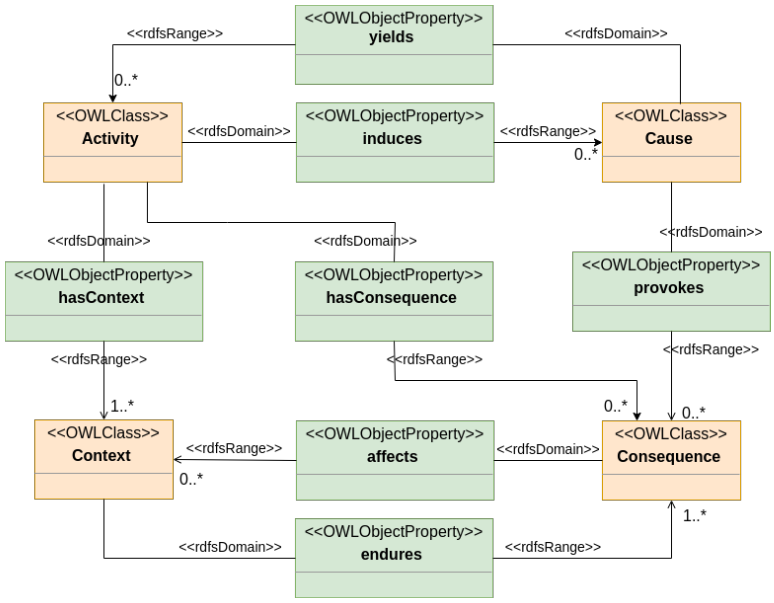


Figure 30: Main concepts and their inter-relations

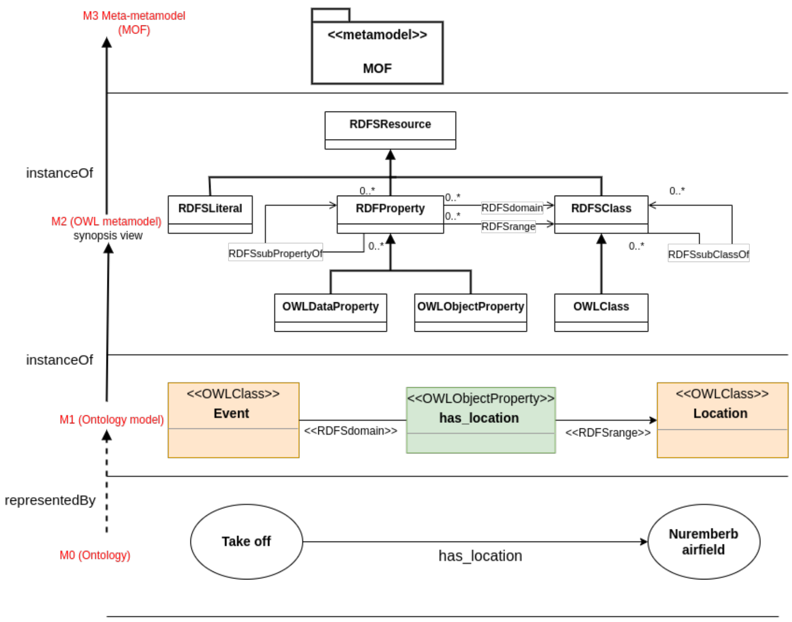


Figure 31: UML profile for modeling ontology with MDA

The first layer (M3) comprises the meta-object facility, a language and self-defining framework derived from the core UML. This language contains basic concepts such as class, association, or data type used for describing other meta-models (Gaševic, Djuric, and Devedžic, 2009).
The second layer (M2) of this modeling architecture is based on the ontology definition meta-model (ODM), in which the OWL meta-model and the RDFS meta-model (ODM, 2007) are found.
The third layer (M1) comprises the ontology model built from components of M2. This layer is followed by the last layer (M0), the real-world representation.
In terms of reuse, as shown in Figure 32 this study uses the time ontology to assign

temporal instances to event or activity entities of the BAEO.

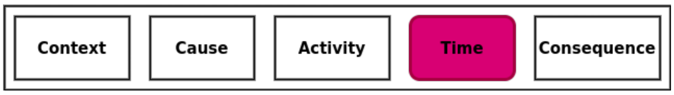


Figure 32: BAEO ontology showing main concepts and reuse of the Time ontology

For further hierarchy levels, Figure 33 presents the consequence class hierarchy,

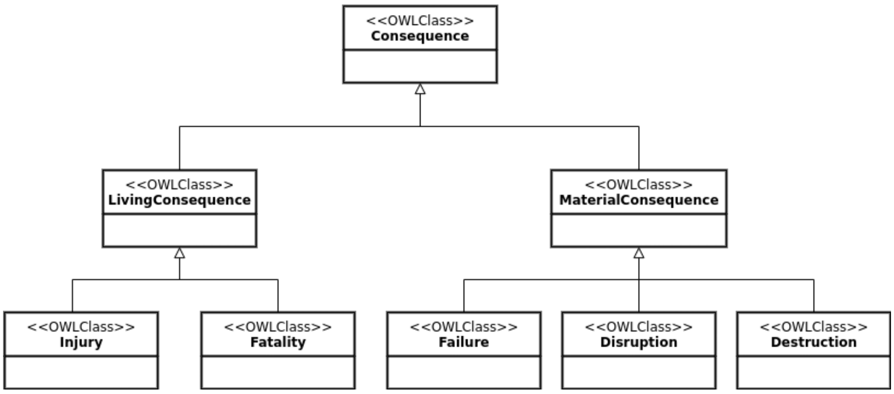


Figure 33: BAEO cause class hierarchy

Figure 34 the context class hierarchy,

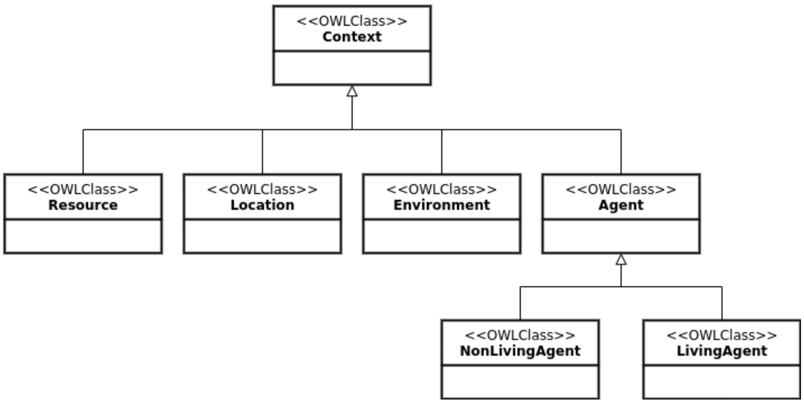


Figure 34: BAEO context class hierarchy

and Figure 35 the cause class hierarchy.

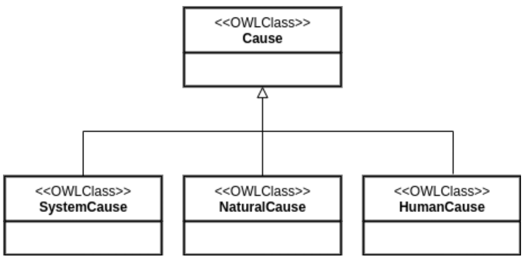


Figure 35: BAEO cause class hierarchy

Code 11: OWL main concepts

```
:Activity rdf:type owl:Class .
:Context rdf:type owl:Class .
:Consequence rdf:type owl:Class .
:Cause rdf:type owl:Class .
:Time rdf:type owl:Class .
```

5.3.3 *BAEO formalization*

Formalization is transforming a model diagram into a machine-readable and reasonable format.
Ontology languages offer constructs to formalize models. However, each language has its level of expressivity and limitations.
Examples of ontology languages are RDF, RDFS, OWL 1, and OWL 2.

Under the proposed methodology, this work adopted an expressive and decidable OWL subset called OWL 2 Description Logic (OWL 2 DL). This language is built from specific DL components, oriented toward expressive ontologies, and offers semantics, interoperability, and reasoning benefits. OWL 2 DL provides constructs for ontology formalization such as *classes* and *subclasses*, *property hierarchies*, *property chain*, *object-properties* for defining relationships between individuals, *data-properties* for their value properties, and *inverse* of properties (Gayo et al., 2017; Horrocks, 2005). The advantage offered by OWL 2 DL cannot be achieved by its counterparts because it provides other vocabularies they do not possess.
For reasoning, this work relies on *Pellet* reasoner because it is free and open-source software that supports the $\mathcal{SROIQ}$ DL as OWL 2 DL. In addition, it is optimized for standard DL reasoning (Bock et al., 2008).
The design presented in the previous sections produces the following formalization in OWL 2 DL.
Code 11 encodes the main concepts of the BAEO
Code 12 encodes the consequence class hierarchy of the BAEO
Code 13 encodes the cause class hierarchy of the BAEO
Code 14 encodes the context class hierarchy of the BAEO

The following rule was defined as presented in Code 15 from which one can infer the number of victims.

5.3.3.1 *Property Chains*

Property chains are mechanisms provided by OWL 2 DL to infer new knowledge from chains of properties available in the ontology.
From the given design, the following chains were uncovered.
Figure 36 shows the agent-activity-cause chain.

Code 12: OWL Consequence sub-classes

```
:LivingConsequence rdfs:subClassOf
:Consequence .

:MaterialConsequence rdfs:subClassOf
:Consequence .

:Fatality rdfs:subClassOf :LivingConsequence .
:Injury rdfs:subClassOf :LivingConsequence .
:Failure rdfs:subClassOf :MaterialConsequence .

:Disruption rdfs:subClassOf
:MaterialConsequence .

:Destruction rdfs:subClassOf
:MaterialConsequence .
```

Code 13: OWL Cause sub-classes

```
:SystemCause rdfs:subClassOf :Cause .
:HumanCause rdfs:subClassOf :Cause .
:NaturalCause rdfs:subClassOf :Cause .
```

Code 14: OWL Context sub-classes

```
:Agent rdfs:subClassOf :Context .
:Location rdfs:subClassOf :Context .
:Resource rdfs:subClassOf :Context .
:Environment rdfs:subClassOf :Context .
```

Code 15: Count victims semantic web rules

```
baeo:Consequence(?C) ^ baeo:isVictimOf(?V, ?C)  .  sqwrl:makeSet(?set, ?V) ->
sqwrl:select(?V) ^ sqwrl:count(?V)
```

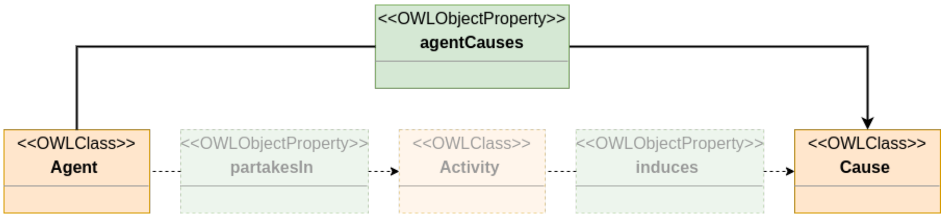


Figure 36: BAEO cause object property chain: Agent-Activity-Cause

This chain identifies the agent at the root of a cause.
Figure 37 shows the activity-cause-consequence chain.

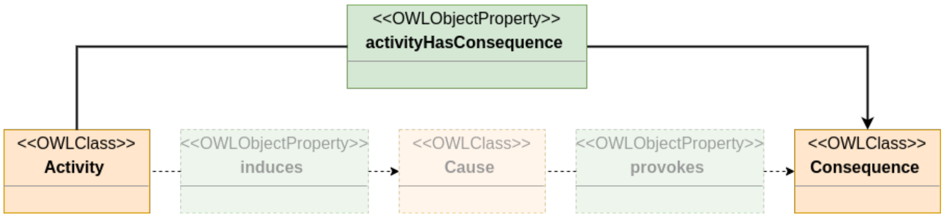


Figure 37: BAEO consequence object property chain: Activity-Cause-Consequence

This chain links a consequence to the activity that produces it.
Figure 38 shows the agent-cause-consequence chain.

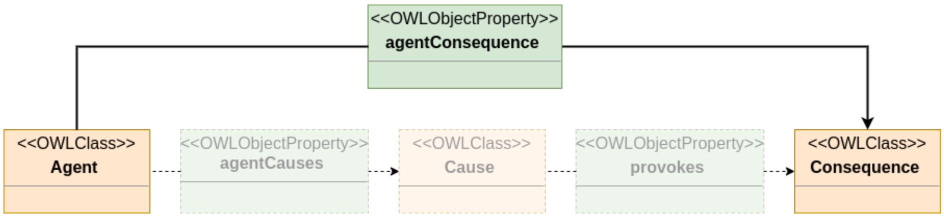


Figure 38: BAEO provocation object property chain: Agent-Cause-Consequence

This chain links an agent to the consequence it causes.
Figure 39 shows the cause-activity-consequence chain.

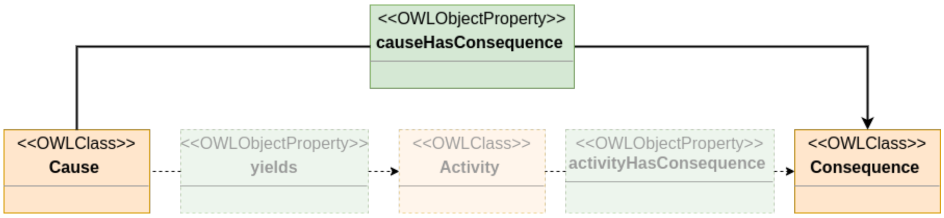


Figure 39: BAEO provocation object property chain: cause-activity-consequence

This chain links a cause to a consequence it provokes through an activity.

5.3.4 *BAEO evaluation*

For this study, the following evaluation and validation were carried out:

- The external evaluation consists in querying the knowledge stored by the ontology regarding the competency questions.
- The internal evaluation consists in checking the ontology consistency with a reasoner. Pellet reasoning was used in this study (Jain, 2021).
- The validation was done with the SHACL.

5.3.4.1 *Validation*

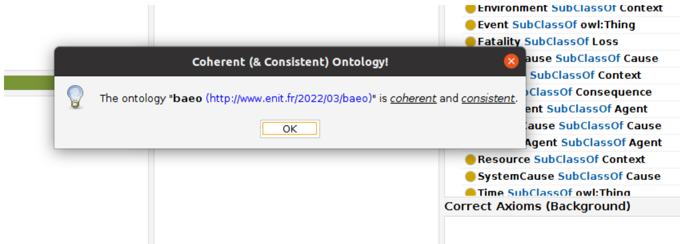


Figure 40: BAEO debugging result shows it is consistent and coherent

The ontology hierarchical structure and relationships were verified using the PROTEGE[2] built-in interaction debugging tool DEBUGGER, which checks the coherence and consistency of ontologies using PELLET reasoner in the case of this study. Figure 40 shows that the built ontology is consistent and coherent.

5.3.4.2 *Constrains verification*

Although OWL 2 DL is suitable for domain description because of its high expressivity, it lacks the means for auto-validation. For this purpose, the W3C community developed constraints languages such as SHACL or the Shape Expression (ShEx), which allows RDF data validations. Significantly, these languages and SHACL, in particular, do not only enhance RDF graph understanding, but they also help to detect problems on data graphs and therefore provide guarantees for better interoperability (Pareti and Konstantinidis, 2021).
In this study, SHACL was used as a language to build validation graphs since it offers basic inference and the possibility of supporting open-world assumptions (OWA) over its counterpart ShEx (Martínez-Costa and Schulz, 2017). SHACL provides a way to define the data model and value restrictions called shape, which an RDF knowledge graph most respect (Cimmino, Fernández-Izquierdo, and García-Castro, 2020; Das and Hussey, 2021).
Code 16 describes the validation of *Event* instances specifying that an event must have a temporal property in the RDF graph.
Code 17 describes the validation of *Activity* instances specifying that an activity has at least a beginning or ending event.

5.3.4.3 *Structural verification*

This study used the ontology pitfall scanner (OOPS!) to verify the BAEO structure. This tool is used for ontology diagnosis and repair because it identifies common mistakes

2 http://protege.stanford.edu/

Code 16: Event validation shape (SHACL)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix baeo: <http://www.enit.fr/2022/03/baeo#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix time: <http://www.w3.org/2006/time#> .
baeo:EventShape a sh:NodeShape;
sh:targetClass baeo:Event;
sh:NodeKind sh:IRI;
sh:xone(
        [
                a sh:NodeShape;
                sh:property
                [
                        sh:path  time:hasTime;
                        sh:class baeo:Time;
                        sh:nodeKind sh:IRI ;
                        sh:minCount 1;
                        sh:maxCount 1
                ]
        ]
        [
                a sh:NodeShape;
                sh:property
                [
                        sh:path  time:hasTemporalDuration;
                        sh:class baeo:Time;
                        sh:nodeKind sh:IRI ;
                        sh:minCount 1;
                        sh:maxCount 1
                ]
        ]
) .
```

Code 17: Activity validation shape (SHACL)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix baeo: <http://www.enit.fr/2022/03/baeo#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix time: <http://www.w3.org/2006/time#> .

baeo:ActivityShape a sh:NodeShape;
sh:targetClass baeo:Activity;
sh:NodeKind sh:IRI;
sh:xone(
        [
                sh:property baeo:EventShape
        ]
        [
                sh:or(
                        [
                                a sh:NodeShape;
                                sh:property
                                [
                                        sh:path baeo:hasEvent ;
                                        sh:class baeo:Event;
                                        sh:nodeKind sh:IRI ;
                                        sh:minCount 1;
                                ]
                        ]
                        [
                                a sh:NodeShape;
                                sh:property
                                [
                                        sh:path  [ sh:inversePath  baeo:beginsActivity ];
                                        sh:class baeo:Event;
                                        sh:nodeKind sh:IRI ;
                                        sh:minCount 1;
                                        sh:maxCount 1;
                                ]
                        ]
                        [
                                a sh:NodeShape;
                                sh:property
                                [
                                        sh:path  baeo:endsActivity;
                                        sh:class baeo:Event;
                                        sh:nodeKind sh:IRI ;
                                        sh:minCount 1;
                                        sh:maxCount 1;
                                ]
                        ]
                )
        ]
) .
```

such as domain and range classes intersections, naming conventions issues, and cycles in taxonomy hierarchy (Poveda-Villalón, Gómez-Pérez, and Suárez-Figueroa, 2014). The OOPS tool validation revealed the BAEO has 01 critical alerts in multiple ranges in its properties. This alert on *BAEO:operatesIn* object property is because an agent can operate in an event or activity as well as a particular resource.
Figure 44 describes the structure of the BAEO obtained after the design process. Figure 41 shows the class hierarchy and Figure 42 the property hierarchy.

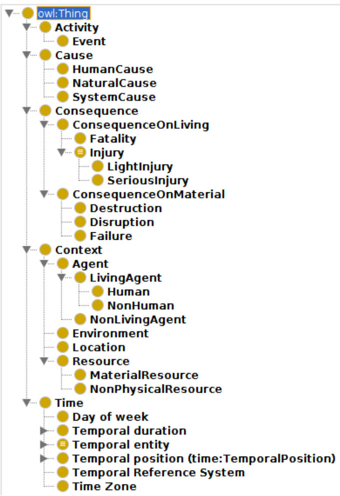


Figure 41: BAEO taxonomy

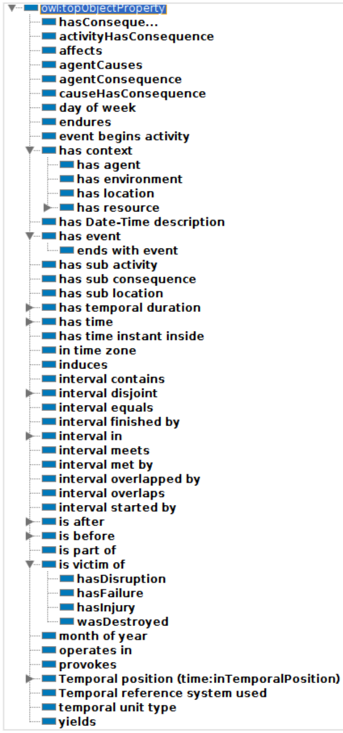


Figure 42: BAEO object properties hierarchy

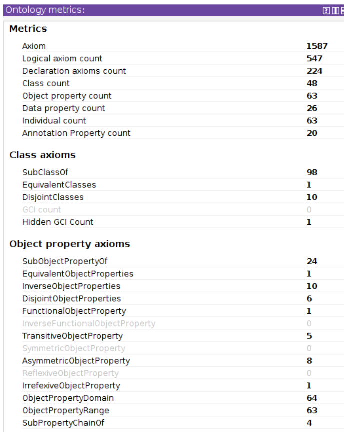


Figure 43: BAEO metrics

Figure 44: BAEO ontology

Figure 43 shows the metrics of BAEO structure in terms of classes, axions, and properties.

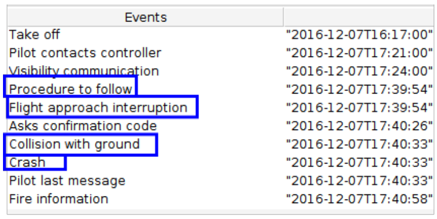
5.4 ILLUSTRATION

An aircraft accident expertise was addressed to apply the constructed BAEO. The accident expertise report of a Piper PA34-200T aircraft that occurred on the 7th December 2016 at Bale-Mulhouse was used for this illustration.

- How did the accident occur?
  This Code 18 is the query that describes the events of the accident and when these events took place. Figure 45a shows the result of this query, and it is compared with the result from the report as shown in Figure 45 . Figure 45b

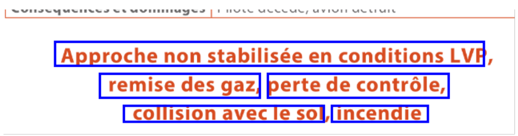
Code 18: Query out how the accident occurred

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX baeo: <http://www.enit.fr/2022/03/baeo#>
PREFIX time: <http://www.w3.org/2006/time#>
SELECT DISTINCT ?Events  ?Time
WHERE {
        {
                ?Events time:hasTime  ?T1.
                ?T1 time:after ?T2.
                OPTIONAL {
                        ?T1 time:before ?T2.
                }
                ?T2 time:inXSDDateTimeStamp ?Time.
        }
        UNION
        {
                ?Events time:hasTime  ?T1.
                OPTIONAL {
                        ?Events  time:hasBeginning ?T1;
                        time:hasEnd  ?T1
                }
                ?T1 time:inXSDDateTimeStamp ?Time.
        }
}
ORDER BY  (?Time)
```



Events	
Take off	"2016-12-07T16:17:00"
Pilot contacts controller	"2016-12-07T17:21:00"
Visibility communication	"2016-12-07T17:24:00"
Procedure to follow	"2016-12-07T17:39:54"
Flight approach interruption	"2016-12-07T17:39:54"
Asks confirmation code	"2016-12-07T17:40:26"
Collision with ground	"2016-12-07T17:40:33"
Crash	"2016-12-07T17:40:33"
Pilot last message	"2016-12-07T17:40:33"
Fire information	"2016-12-07T17:40:58"

(a) Chronological order of events concerning the Piper PA34-200T accident



(b) Summarized chronological order of events from report

Figure 45: Chronological order of events concerning the Piper PA34200T accident from the ontology and the report

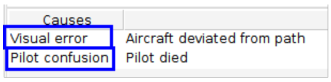
shows the summary of the main events of the accident from the report. This result is included in the output of the events from the ontology as shown in Figure 45a

- What are the causes of the problem?
  This query in Code 19 shows the accident's main causes and consequences. Figure 46a is the output of this query. Technical causes cannot be captured, as

Code 19: All causes query

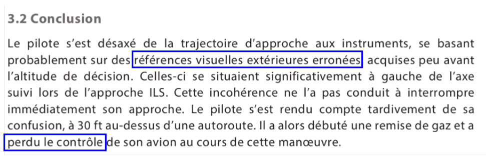
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX baeo:    <http://www.enit.fr/2022/03/baeo#>
PREFIX time:  <http://www.w3.org/2006/time#>
SELECT DISTINCT ?Causes ?Consequences ?Activity
WHERE {
        ?Causes a ?C.
        ?C rdfs:subClassOf  baeo:Cause.
        OPTIONAL {
                ?Causes baeo:provokes   ?Consequences.
        }
}
```

shown in Figure 46b because of the lack of specific domain knowledge. This is normal since BAEO is designed to be a high-level ontology that will be extendable to various domains.



Causes	
Visual error	Aircraft deviated from path
Pilot confusion	Pilot died

(a) The main causes of the accident from the ontology



3.2 Conclusion

Le pilote s'est désaxé de la trajectoire d'approche aux instruments, se basant probablement sur des références visuelles extérieures erronées acquises peu avant l'altitude de décision. Celles-ci se situaient significativement à gauche de l'axe suivi lors de l'approche ILS. Cette incohérence ne l'a pas conduit à interrompre immédiatement son approche. Le pilote s'est rendu compte tardivement de sa confusion, à 30 ft au-dessus d'une autoroute. Il a alors débuté une remise de gaz et a perdu le contrôle de son avion au cours de cette manœuvre.

(b) The main causes of the accident from the report

Figure 46: Causes from ontology and the report

- Who are the victims of this problem?

  This query in Code 20 lists the victims of the accident and the consequences which they suffered.
  As in the report, the victim is the *Pilot* who died in the accident and whose aircraft was destroyed. Figure 47a shows the result of this query and its comparison with the one in the report. Figure 47 shows the similarity between these results. The result from the ontology is consistent with the report but lacks some detail about the domain as the report because it is still a high-level representation.

The base accident expertise ontology is hosted on git for easy download and reuse. The BAEO is available at ENIT[3].

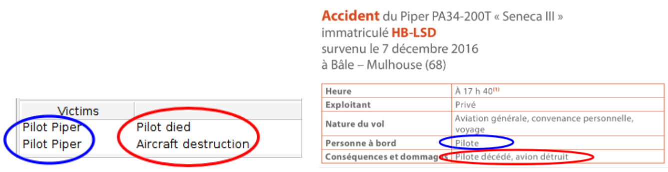
3 https://git.enit.fr/ssonfack/baeo

Code 20: Shows all victim of the accident

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX baeo: <http://www.enit.fr/2022/03/baeo#>
PREFIX time:  <http://www.w3.org/2006/time#>


SELECT DISTINCT ?Victims ?Consequences
WHERE {
        ?Victims a ?X .
        ?X rdfs:subClassOf*  baeo:Agent.
        ?Victims baeo:isVictimOf  ?Consequences.

}
```



(a) The victim and the consequences he suffered from **(b)** The victims from the report

Figure 47: The victim from the ontology and the report

5.5 DISCUSSION

To our knowledge, the task of accident expertise representation using ontology has not been investigated. However, some research has been done in domains similar to accident expertise, such as accident, risk, or safety ontology.

The proposed study is dedicated to the field of accident expertise. It describes the construction of a high-level ontology (BAEO) to represent knowledge of accident expertise.

This study bases its methodology on an iterative, cyclic, and MDA approach that complements the approaches proposed by (Hassan and Mokhtar, 2021) and (De Nicola, Missikoff, and Navigli, 2009) for building autism ontology and a software engineering approach for ontology construction.

These approaches are based on well-known methods such as METHODOLOGY (Uschold and King, 1995) and ontology development 101 (Noy, McGuinness, et al., 2001) which define essential steps to consider when designing ontology.

Because the proposed ontology is domain-independent and dedicated to accident expertise knowledge representation, there is a need to have an understandable design. This approach based on model-driven development provides not only a systematic engineering process but also interoperability and reusability (Araújo Silva et al., 2021). These characteristics will allow developers and knowledge engineers to extend this foundation to their domain of concern more easily.

Furthermore, the proposed study distinguishes itself from others by its ability to capture essential features of accident expertise such as cause, consequence, event, context, and domain. In contrast, existing ontologies in fields closed to accident capture only some of these essential concepts. Table 16 shows a benchmark of this study compared to existing projects.

	Domain	Expertise knowledge			
	General purpose	Cause	Consequence	Events	Context
ECCAIR Aviation Ontology (Křemen et al., 2017)	No	No	No	**Yes**	**Yes**
Ontology Modeling accident scenarii (Maalel et al., 2012)	No	**Yes**	No	**Yes**	**Yes**
Ontology safety risk concepts (Kaindl et al., 2016)	No	**Yes**	No	**Yes**	**Yes**
Vehicular accident Ontology (Barrachina et al., 2012b) , Car accident Ontology for VANETs (Barrachina et al., 2012a)	No	No	No	No	**Yes**
Ontology metro accident (Wu et al., 2020)	No	**Yes**	No	No	**Yes**
BAEO	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Table 16: Benchmark of BAEO and other ontologies

BAEO is a base ontology that was designed with reusability in mind. Its generalized concepts facilitate its reuse in various fields of accident expertise. Bringing more specializations to these concepts makes it possible to extend BAEO to specific domains. For example, a road accident expertise ontology can be obtained by (1) extending the **baeo:Human** concept with the *rao:Person* concept from the road accident ontology as shown in Figure 48, (2) extending the *baeo:MaterialResource* concept with the *rao:Vehicle* concept. *rao* stands for road accident ontology[4] This shows that BAEO is domain-independent and captures causes, consequences, events, and context while other ontologies do not.

5.6 CONCLUSION

Accident expertise is an expensive and knowledge-intensive activity used by human experts to understand problems. In this study, a base ontology was designed to represent accident expertise knowledge using a model-driven methodology and OWL.
The proposed ontology was evaluated, and a use case was illustrated from the BEA's aircraft accident expertise report, and satisfactory results were obtained compared to

4 https://www.w3.org/2012/06/rao.html

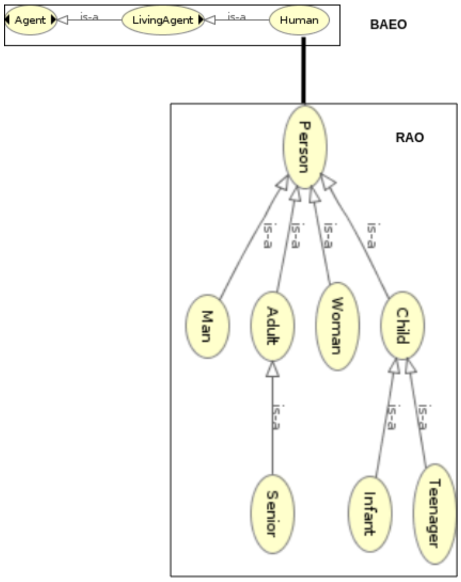


Figure 48: Extending BAEO for road accident expertise

conclusions presented in the accident expertise report of the illustration.
Furthermore, this study showed that this ontology could be reused in other accident expertise fields in the example of the road accident.
However, this ontology suffers from its strengths (high-level ontology) because it cannot represent details about specific domains; therefore, its extension with domain knowledge is recommended if one is required to capture a certain level of detail of accident expertise.

6 CONCLUSION AND PERSPECTIVES

This thesis proposes models to alleviate humans' tasks in Collaborative Expertise Processes that made difficult due to their doubtful context and lack of knowledge. The proposed approaches support traceability, collaborative reasoning, uncertainty management, and communication among experts as recommended by the *NF X50-110* standard. Furthermore, lessons can be learned from the proposed knowledge representation.

To accomplish the proposed contributions, this thesis started by investigating technological foundations of knowledge representation, reasoning, and concepts close to Expertise Process and collaborative intelligence, including the main principles of uncertainty management theories to propose meaningful solutions. Afterward, three points were selected to contribute to this topic of Collaborative Expertise Process Modeling and Reasoning. These contributions include a design of Expertise Process and knowledge representation based on human-machine collaborative reasoning that relies on hypotheses and possibility theory, human belief integration with logic programming using DST theory of uncertainty and an ontological description of accident expertise reports.

The subsequent sections provide these contributions, followed by an outlook of this study.

6.1 CONTRIBUTIONS

The following sub-sections depict the contributions of this thesis to collaborative Expertise Processes. The first contribution is the combination of default reasoning and belief function theory, the second is the design of a Expertise Process, including reasoning mechanisms, and the third is an ontological representation of accident expertise.

6.1.1 *Contribution 1: Expertise Process modeling and reuse*

An Expertise Process widely uses hypothesis-driven approach to solve problems under conditions of limited knowledge and understanding conditions. Although this approach has been standardized, it lacks tools to assist experts in exploring and tracking all possible explanations of a problem. It also lacks mechanisms to capture experts' reasoning methodology and knowledge produced during the process. In order to acquire, learn, share and reuse experts' knowledge applied during this process while assisting experts in bringing understanding to problems, this study introduces a framework that formalizes experts' knowledge and methodology for solving problems collectively using hypotheses. The framework is based on Hypothesis Theory extended with qualitative doubt and a systematic reasoning process using hypotheses exploratory graphs. The proposed approach allows experts to collaborate and interact with machines through a simple procedure, while the knowledge and steps followed during the expertise are stored. In addition, it is possible to learn the causal influence of knowledge over hypotheses and a semantic causal graph of the problem being solved. Furthermore, an experiment conducted on an actual case verified the feasibility and effectiveness of the approach. The knowledge derived from HEG is better synthesized and can stand as experience for similar problems.

6.1.2 *Contribution 2: Default reasoning and uncertainty*

For this contribution, this study uses ASP, which is a declarative knowledge representation language that uses non-monotonic reasoning to search for all answer sets or models of a specific problem. This reasoning mechanism makes ASP suitable for problem-solving activities, such as expertise, where there is a lack of knowledge, and where defeasible reasoning is required. However, this language is not equipped with a convenient means to select a preferred model among its answer sets as experts proceed in Expertise Processes. Clearly, in Expertise Processes, experts who have acquired knowledge from their experience will express possible explanations and, based on their beliefs and reasoning, will select the most appropriate ones for the problem.
To have the best of ASP logic programming language and human experience, this study proposes and illustrates a general and domain-independent framework that extends ASP using experts' knowledge. The framework applies belief to answer sets to systematically find explanations for expertise activities.
This extension provides a means to evaluate ASP models' beliefs using experts' evidence distributions while reducing the knowledge-intensive load of the Expertise Process.

6.1.3 *Contribution 3: Expertise ontological representation*

Expertise contributes to the development of society in general, as it helps to elucidate unknown situations and facilitates problem-solving. For example, it improves accident understanding by describing how it happened and identifying events, causes and consequences. As a result, knowledge from accident expertise is helpful for safety-systems designing, decision-making, lessons learned, and problem-understanding. However, existing representations of accident knowledge, such as documents, relational databases, or accident ontologies do not fulfill accident expertise expectations. Moreover, these representations are unlikely to provide the appropriate use of accident expertise knowledge.
This study presents a base ontology for accident expertise knowledge representation designed with model-driven methodology and semantic web tools. This ontology obtained satisfactory results on its competency questions evaluation. When extended and reused on an aircraft accident expertise taken from the French Bureau of Inquiries and Analysis (BEA) for civil aviation safety, its structure and constraints gave positive outcomes.

6.2 PERSPECTIVES

This section describes future studies that can be carried out following this thesis. After contributions made to the Expertise Process, much is still to be done in this field. This section presents some important gaps in this study from a short- and long-term perspective.

6.2.1 *Short-term perspective*

In the short term, it will be necessary to build a digital platform that domain experts can use to carry out expertise driven by hypotheses as described in this thesis. On the one hand, this tool will facilitate the hypotheses reasoning-based approach and popularize the proposed technique. On the other hand, the tool will assist in storing and sharing experts' synthesized knowledge (experience) to ease future problem understanding.

6.2.1.1 *Functional requirements*

An overview of such a digital platform is shown in Figure 49. Its main functional components and operating mode are formally described in Figure 50 and explained in subsequent paragraphs.
The main components of the system are as follows:

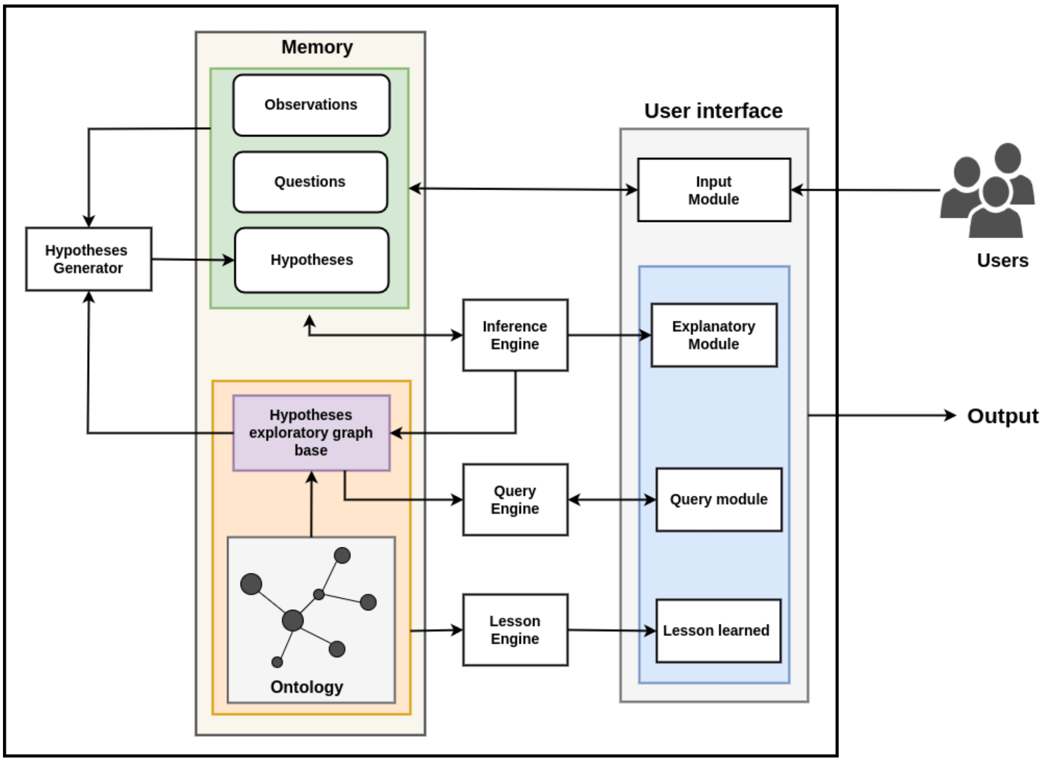


Figure 49: Overview architecture of the hypotheses reasoning system

- Memory module
  This is the central module of the system and comprises a sub-module to manage

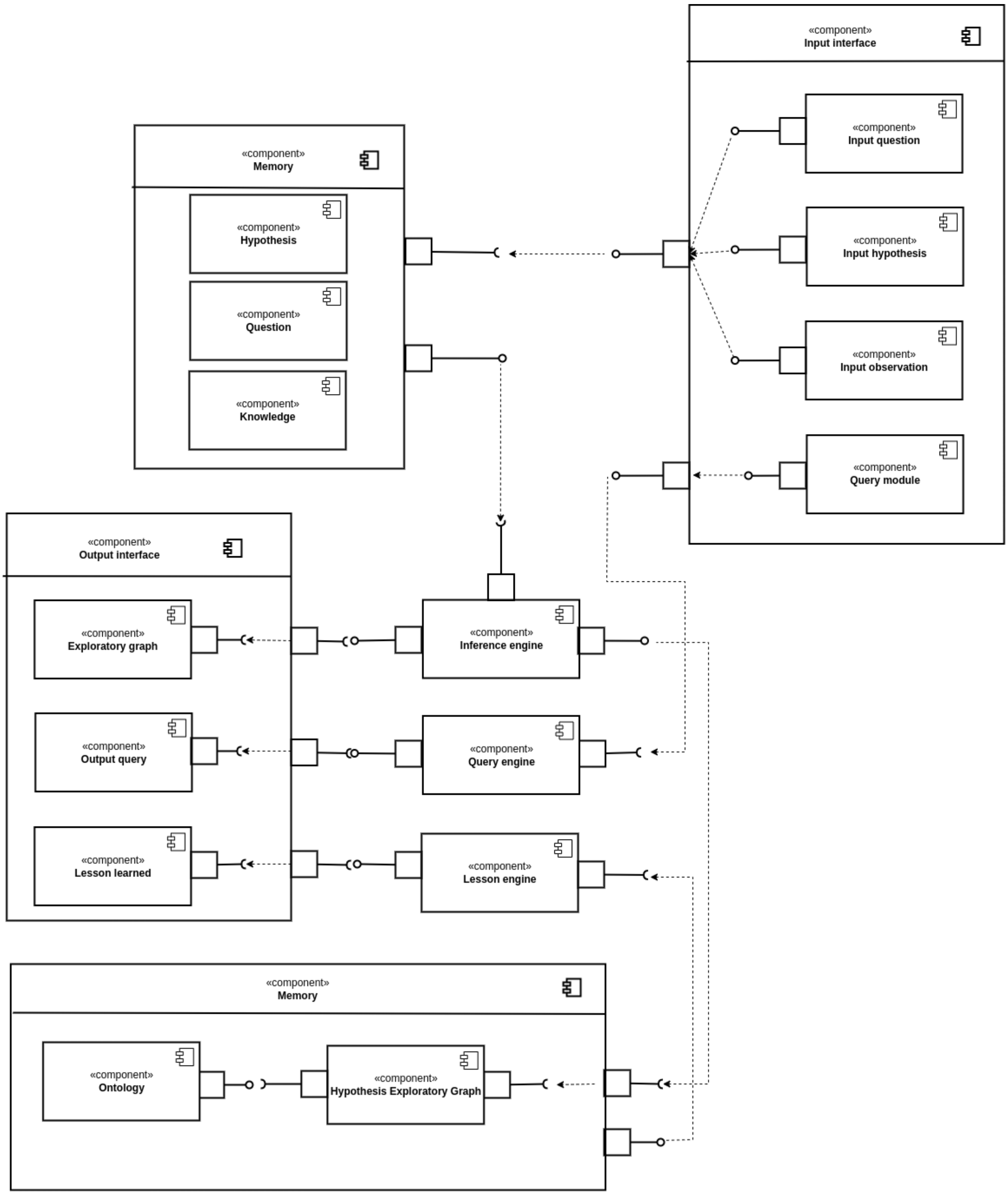


Figure 50: UML component diagram of the proposed Expertise Process system

hypotheses, observation or additional knowledge, and hypotheses exploratory base.

- Observations
  Correspond to the available knowledge about the problem being solved.

  - Hypotheses
    These are formalized generated and proposed hypotheses by the system or expressed by domain experts.
  - Hypotheses exploratory graph base
    This is where the final HEGs are stored after their construction.

- Hypotheses generator
  This component's role is to generate user hypotheses using the existing hypotheses' exploratory base and observations provided during the exploration process.

- Inference engine or reasoner
  This component is the non-monotonic reasoner base on the Hypothesis Theory which uses additional knowledge to extend hypotheses proposed by experts.

- Lesson engine
  The role of this component is to generate lesson learned from HEG and a domain ontology.

- User interface
  This component contains modules necessary for entering or outputting information to the system. It stands as the communication interface between the system and users.
  - Input module
    This module requires a semi-structured template that will be easier for experts to use and simple for machines to process.
    For this purpose, techniques such as Text Reasoning Network (TRN) (Sizov and Öztürk, 2013) can be revisited with more reasoning mechanisms and semantics to facilitate hypotheses formalization. Another supportive tool that can be exploited to ease interaction with the proposed system is the *Jigsaw notation* which has been used for knowledge acquisition in the field of ontology (Sanctorum et al., 2022). This notation is simple, human-friendly and can be an ideal tool for human-machine communication.
  - Query module
    This is responsible for retrieving part of the queried graph to users.
  - Visualization module
    This will help to visualize the hypotheses exploratory graph.
  - Explanatory module
    This sub-component will output how the graph was constructed. This explanation will also stand as a criterion for users' acceptability of the system. Furthermore, it can be used for debugging in case of unexpected results.

Figure 51 shows how information flows in the proposed system from the user input to the result.

- Step 1
  This first step is the system's entry point, where the experts carry out their

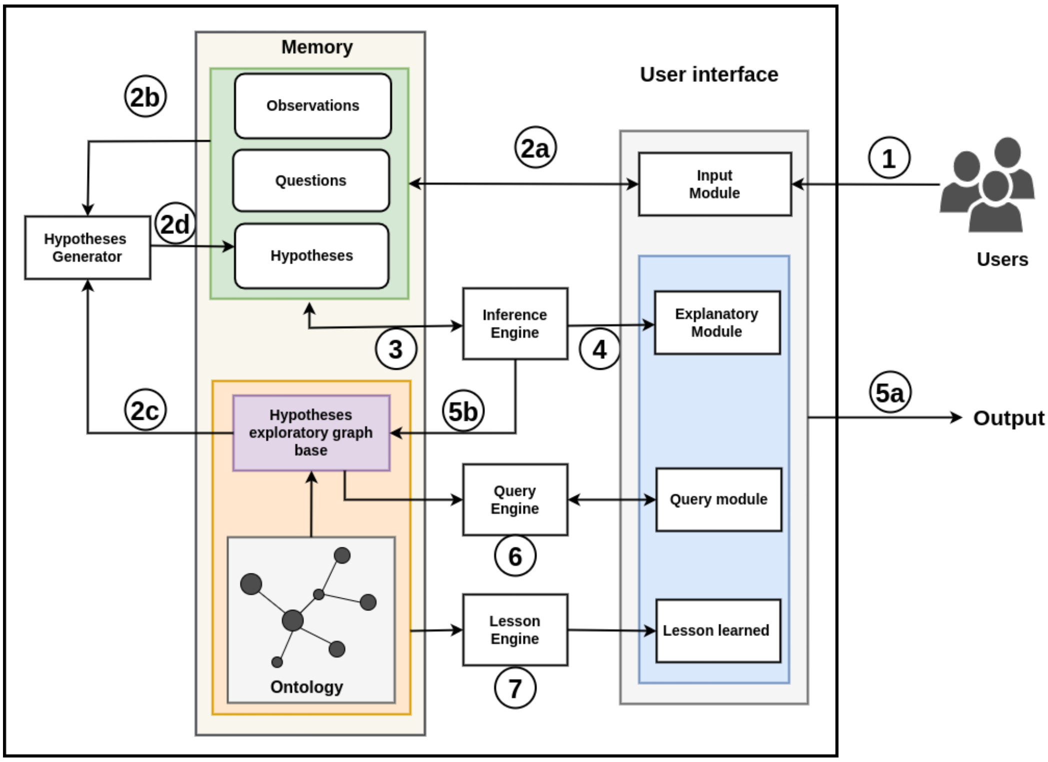


Figure 51: Overview workflow

expertise and enter hypotheses and observations related to the problem their wish to understand.

- Step 2
  Formalized inputs (hypotheses and observations) are given to the next module for processing. This interactive step gives experts the means to enter their own hypotheses or accept those proposed by the system.
  (a) Domain experts' inputs are formalized (observations and hypotheses of the problem).
  (b) The hypotheses generator module assesses the current problem observations and hypotheses.
  (c) The hypotheses generator module assesses past hypotheses exploratory graphs.
  (d) The hypotheses generator module uses available observations, hypotheses, and past hypotheses exploratory graphs to suggest hypotheses to domain experts.

- Step 3
  The inference engine will use at each iteration of the Expertise Process, available observations, and hypotheses to construct the HEG while evaluating these hypotheses.

- Step 4
  The inference engine produces intermediate results during the Expertise Process

and the final result at the end of the activity. This result includes the HEG and possible explanations.

- Step 5
  (a) The result that includes the HEG and its possible explanations can be accessed by end users through an intuitive interface
  (b) The hypotheses exploratory graph is stored so the hypotheses generator module can use it in future Expertise Processes.

- Step 6
  The query engine's role is to access portions and components of the HEG. For example, it can be used to access hypotheses, knowledge, and questions of given expertise. Moreover, querying components of expertise can be used by experts for learning purposes.

- Step 7
  This step is used to learn from existing expertise. Three lessons can be learned from this module: (1) lesson from graph analysis, (2) lesson from knowledge-hypothesis causal structure and (3) lesson from semantic mapping.

Figure 52 is the use case diagram that presents actors of the proposed system and how they will interact.

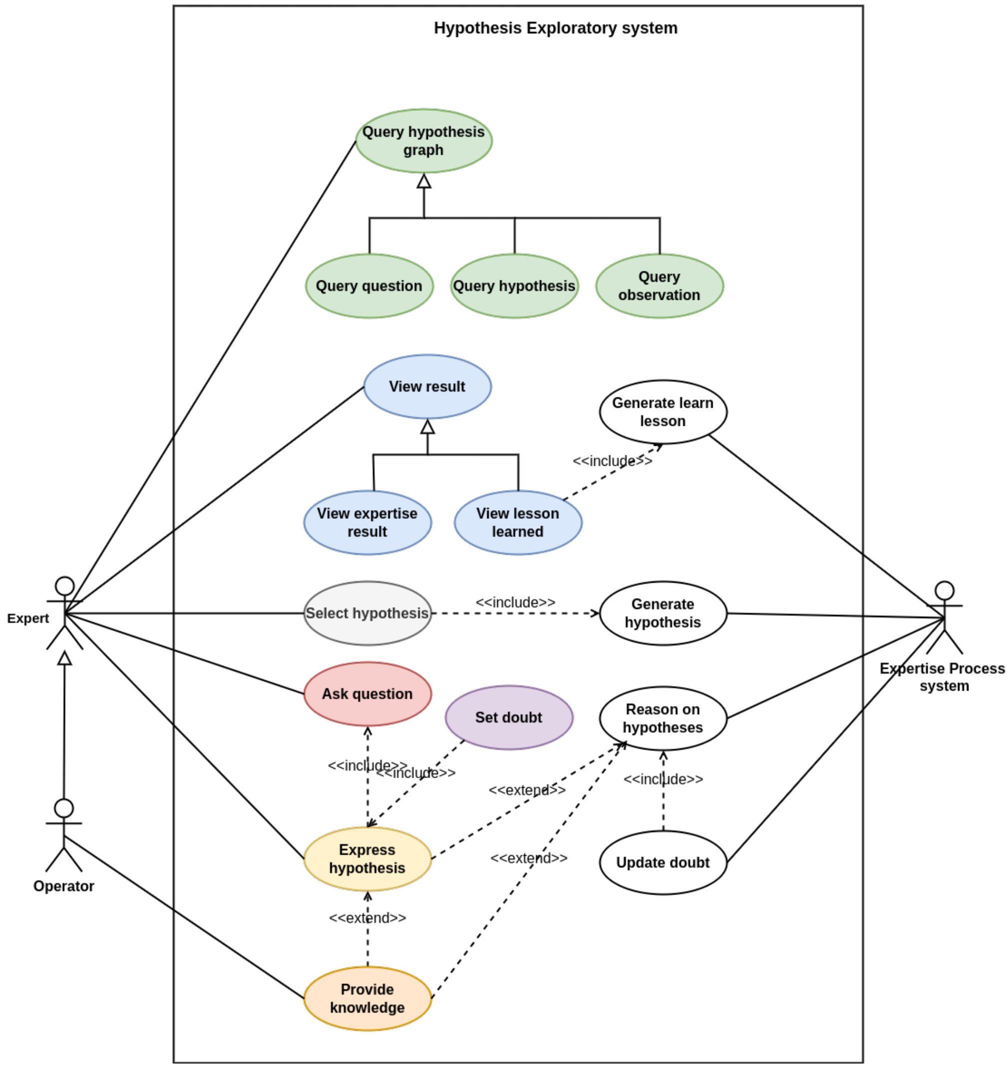


Figure 52: Expertise Process system use case diagram

6.2.1.2 *Non Functional Requirements (NFR)*

NFRs are not parts of the core requirements of the desired system but are essential for its acceptance. This study identified the following requirements to complement the core ones.

- Accessibility: This system's design should consider devices such as personal computers and experts' handsets. For this reason, using Web technologies may avoid building multiple views for multiple devices.
- Response time: the proposed system should have a reasonable response time so that the users will not be kept waiting for the results. The interaction between the users and the system should be similar to those of a chat-bot.
- Storage requirements and data availability: Appropriate storage in terms of data structure and technology, such as centralized or distributed, will improve the access time to the knowledge used by the system, therefore reducing the latency.

- Security: How can the knowledge produced and stored by the system be accessed only by authorized users? This is a central question that the security requirement will answer.
- Globalization/localization: This requirement will allow users with different languages to work together on the same problem.

6.2.2 *Medium-term perspective*

This thesis's medium-term perspective focuses on reusing HEG to solve new problems and assist experts in their tasks. For this purpose, Case Based Reasoning (CBR) is an ideal choice that has been used in this type of situation.

CBR is a problem-solving mechanism similar to human reasoning, in which experiences called cases are used to solve new problems (El-Sappagh and Elmogy, 2015). In its operation mode, CBR relies on four steps as shown in Figure 53 which are: (1) *retrieve* consists in looking for the most similar case to the entry problem (2) *reuse* of a selected case to solve the current problem (3) *revise* the solution obtained from the reuse after an evaluation (4) *retain* the solution as a new case in the base if it is satisfactory.

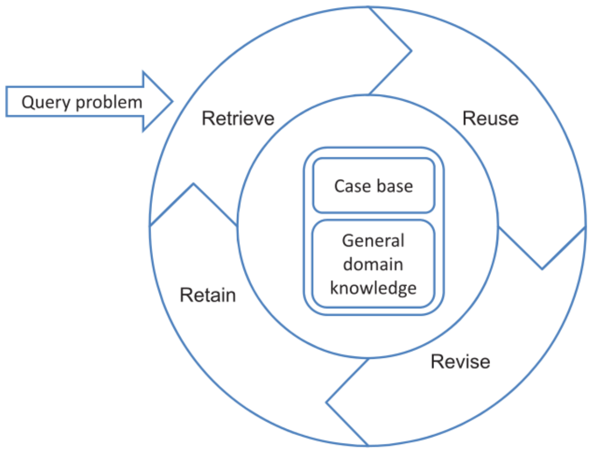


Figure 53: CBR methodology from (López, 2013)

CBR has advantages over some reasoning systems. Firstly, the engineering effort needed to build them is lesser than those of other knowledge base systems, making it a solution for knowledge acquisition bottleneck (Cordier, Fuchs, and Mille, 2006). Secondly, it does not require a massive quantity of cases to be operational compared to methods like neural networks of statistical machine learning. Thirdly, its interpretability encourages users to accept its solutions since it is based on human-like reasoning (Shepperd, 2003). Finally, CBR achieves transparency and justification without particular effort, making it interpretable and explainable (Paraschiv and Sermpinis, 2021).

To apply CBR to Expertise Process, it is essential to define the case structure carefully. Figure 54 shows how cases proposed in this thesis can be structured. A case

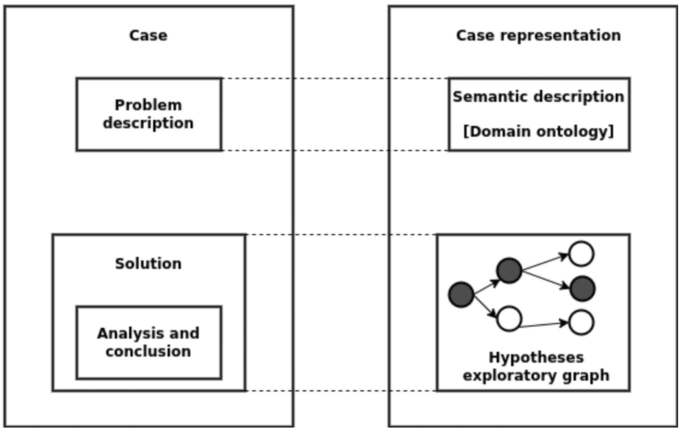


Figure 54: Case structure and representation

representation, which corresponds to an Expertise Process, is described by two components: (1) a domain knowledge of the problem that represents its context. (2) the HEG corresponding to the solution to the case. These components will be used to search for similar cases from an Expertise Process base.

Reusing past Expertise Processes can help reduce the cognitive task of asking questions and expressing hypotheses carried out by human experts during Expertise Processes and consequently speed up the whole process.

6.2.3 *Long-term perspective*

From a long-term perspective, there is a need to align expertise activities with current technological standards and AI. This alignment implies, on the one hand, efficiently carrying out expertise within Industry 4.0 (I4.0) and Industry 5.0 (I5.0) environments, on the other hand, using available technologies to improve Expertise Process both before, during and after.

6.2.3.1 *Expertise system for I4.0 and I5.0*

I4.0 and I5.0 are collections of technologies created for the mass personalization of manufacturing products that affect several industry areas, such as production efficiency, order management, research and development, user experience, and product life cycle. In other words, they offer smart manufacturing in industries by driving them with data and knowledge (Souza, Ferenhof, and Forcellini, 2022; Tao et al., 2019).
These two Industrial Revolutions take advantage of advances in information technology and particularly Internet of Things (IoT), cloud computing, Cyber-Physical System (CPS), Digital Twin (DT)s, virtual and augmented reality, additive manufacturing, data sciences, and simulation to name a few (Sony, 2020). However, I4.0 refers to the integration of machines, processes, and systems to maximize industries' performance and optimization in industry. On the other hand, I5.0 focuses on the collaboration and interaction between humans and machines. In fact, I5.0 recognizes the power of industry to achieve societal goals.

For these two industrial revolutions, the concepts of CPS and DTs are essential to their implementation and will be explored and used for the vision of Expertise Process system proposed in this thesis.
CPS is a transformative technology that integrates the physical and the cyber world to realize the I4.0 vision (Raisin et al., 2020; Suhail et al., 2022). It can be implemented with 5C architecture as shown in Figure 55 and describe as follows. *Connection* for sensor networks and data collection, *convertion* which converts the data from sensors into information, *cyber* that serves as a hub of information coming from different sensors, *cognitive* concerns knowledge and decision-making, *configuration* regards configurations that will affect the physical world. A DT is a virtual replicate of a physical world

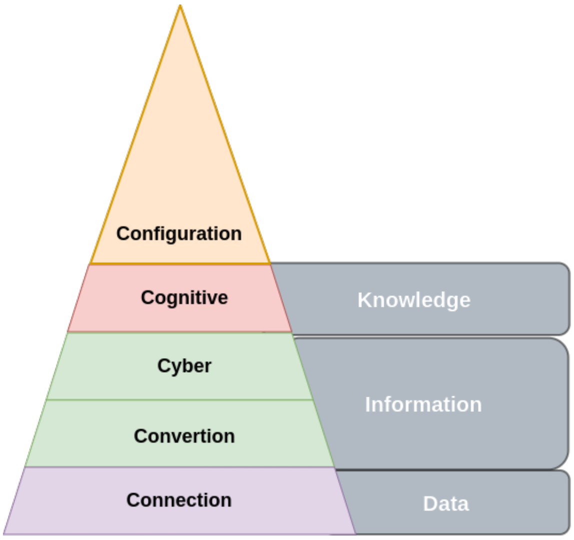


Figure 55: 5C architecture for implementing CPS

capable of simulating its characteristics, attributes, and behavior and providing feedback. DTs like CPS use data from sensors to provide a complete bi-directional dynamic mapping of the physical world and a digital footprint of this world. They are based on three main components: (1) *data* to acquire data from the physical world required for its structure and behavior representation, (2) *model* is the computational description and understanding of the physical world, capable of mimicking this world's behavior or prescribing action from business logics, and (3) *service interfaces* open a DT to third parties services or other DTs to access its data and invoke its capabilities (Schalkwyk, Lin, and Malakuti, 2019).
From DTs, companies can predict and detect physical issues more accurately.

This section presents an overall architecture for Expertise Process systems with respect to I4.0 and I5.0 as shown in Figure 56 and describes how it can be used in the context of these new Industrial Revolutions. This architecture shows how third-party systems can be used to enhance the proposed Expertise Process System. The architecture is made up of three layers. The first layer is the *access layer* which offers different types of interaction with the system.
The communication with CPS, DTs, other Information Systems, and operators, intends to provide additional knowledge to the Expertise Process system and make it context-aware as they provide real-time knowledge of the domain in which expertise

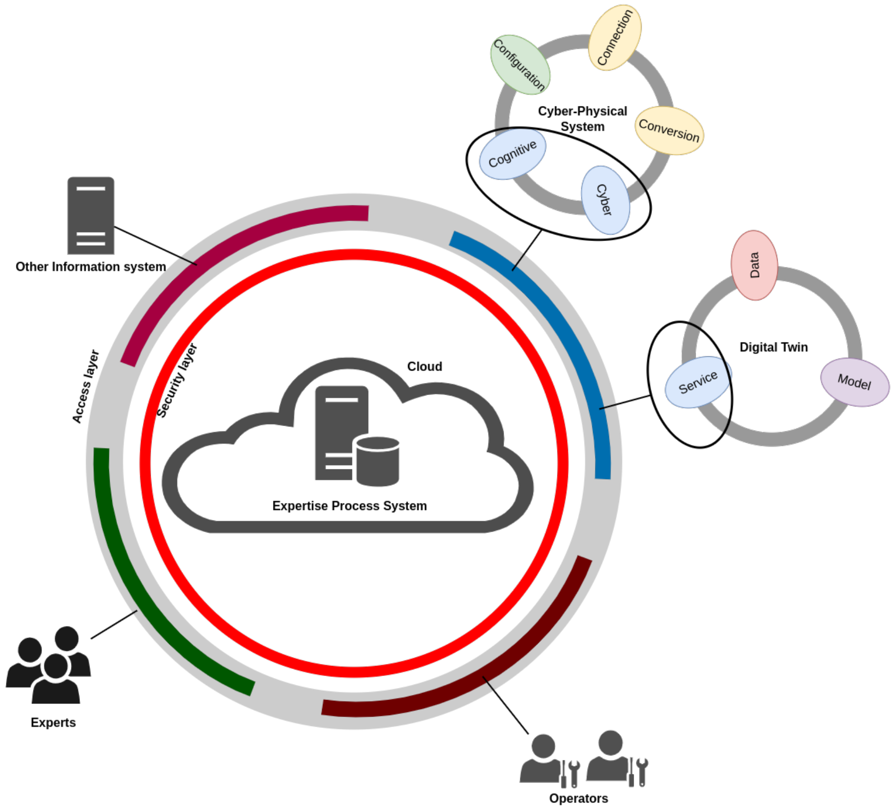


Figure 56: Overall architecture for Expertise Process system

is being carried out. To be more precise, the communication with the CPS is done from its *cyber* and *cognitive* components, which respectively collect information and provide knowledge and decisions about the physical world. In addition, the DT can be used for simulation and verification of hypotheses expressed in the physical world. Furthermore, the *operators* working with machines for the case of I5.0 can also provide pertinent knowledge from their collaboration with machines through this layer.
Other *information Systems* refer to classical information systems that should not be left out when carrying out an expertise in the physical worlds of these systems. They are used to collect knowledge for the Expertise Process system.
Finally, there is access reserved for *experts* who are in charge of the Expertise Process.

The second layer is the *security layer* which function is to protect the Expertise Process system, including its knowledge base, from unapproved access, maintain its confidentiality, and integrity. In fact, cloud computing is sometimes subject to security attacks such as denial of service (DOS), which may cause the cloud resource to be unavailable and thus affect the confidentiality, integrity, and availability of the cloud resources and hosted software.
Even though cloud computing saves internal information technology resources, it is exposed to security risks since there may be a difference in the implementation of configurations and settings between the cloud provider and the cloud consumer services (Alqahtani and Gull, 2018).

The third layer is the *cloud* environment in which the Expertise Process system, as presented in this thesis, can be hosted. Cloud computing services exist in three types: (1) Private cloud owned by an organization to provide services to its users, (2) Public cloud owned by a third party to provide services to other organizations, and (3) Hybrid cloud is a combination of both private and public cloud services.
The primary services offered by cloud providers are (1) Infrastructure as a service (IaaS) for virtual infrastructures such as virtual servers, operating systems, and virtual memories, (2) Platform as a service (PaaS) for virtual platforms such as databases and middlewares, and Software as a service (SaaS) for software and application services that cloud consumers can interact with. The use of cloud computing presents two main advantages. The first is that hosting on a service-oriented architecture will offer flexibility to a virtual network for an on-demand service such as infrastructure, platform, or software anytime and anywhere. It will help fulfill non-functional requirements regarding response time, accessibility, and storage because it is possible to adjust the infrastructure requirements on demand.
The second is that the Expertise Process system users will be able to use the system online (Omri et al., 2018; Upadhyay, 2017) and to work collaboratively from different geographical areas.

In addition, CPS, DT can also benefit from an Expertise Process system. In some cases, the expertise result or lesson learned from expertise can be used to set specific configurations and serve as feedback for the physical world. For this case, the Expertise Process system will be connected to the CPS *configuration* layer.
For DT, this exchange of knowledge is done through the service interface of the DT.

6.2.3.2 *The use of Artificial Intelligence in Expertise*

Globally, expertise can be divided into three main phases as listed in Figure 57. First is the phase before experts carry out the expertise (*Before expertise*). This phase comprises activities such as problem classification that will identify and group types of problems, experts' classification, grading the human experts, and aligning them with their respective areas of expertise. Outcomes of this phase can contribute to building human experts' recommendation systems, which are highly important for quality results based on experts' characteristics, expertise problems, and past activities.
In the second phase, experts carry out their activities that can assist in solving the problem at hand. In this phase, in general, AI can assist in defining semi-automated procedures for expertise. These approaches have to consider human doubt, tacit and past knowledge and should be able to satisfactorily use AI learning techniques with a priority on explainability for expertise.
HEG in particular can be complemented with other machine learning techniques to provide robust models that will ease and enhance human-human and human-machine collaboration with suitable communication protocols between entities collaborating within an expertise.

In addition, the proposed methodology of human-machine collaboration that produced the HEG representation can be exploited in complex problems faced in current Industry 4.0 and 5.0. These technologies face challenges such as modularity, inter-

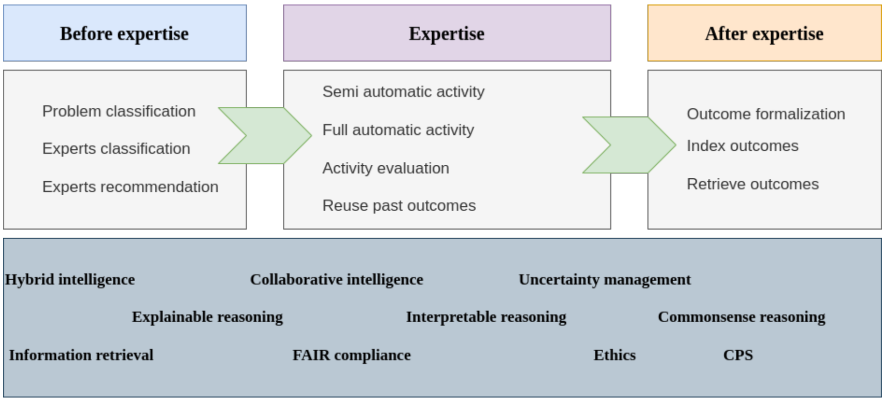


Figure 57: Expertise and AI

operability between humans and objects, virtualization with CPS and digital twins, decentralization of processes and activities, and real-time data storage and analysis. As a result, I4.0 and I5.0 require skills such as robotics to IoT passing through big data for mass storage, AI for learning, and virtualization (Crnjac, Veža, and Banduka, 2017) in order to overcome challenges that will occur.
The use of multiple technical fields will make it challenging to understand problems that can occur in industries aligned with the new industrial revolution. For this reason, the proposed hypotheses reasoning approach which combines human and machine capabilities is suitable for unlocking blockages faced by companies that have adopted Industry 4.0 and 5.0. Human experts' collaboration is an additional asset since it will allow various domain experts to collaborate efficiently and thereby increase the chances of understanding what has gone wrong.
The main objective of the contribution of AI to expertise is to assist humans in this knowledge-intensive task and to acquire and store the knowledge and process in both a human and machine computable format. These will be of benefit in a versatile environment such as Industry 5.0 and even more when reusing the knowledge acquired.

Finally, the *After expertise* phase activities will consist of formatting the outcome of expertise so that it can be human-readable as well as machine-readable, making sure that the expertise outcome can be searched, retrieved, and reused easily. Furthermore, expertise outcomes should be Findable Accessible Interoperable Reusable (FAIR) with high semantics to guarantee transparency and traceability. Hence expertise outcome will be trustworthy.

This thesis stands as new direction for AI in expertise to reduce human cognitive labor while fostering efficiency. Applying AI to this field will help save time and make expertise outcomes more accessible.

Abraham, Ajith and Baikunth Nath (2000). "Hybrid intelligent systems design: A review of a decade of research." In: *IEEE Transactions on Systems, Man & Cybernetics (Part-C) August*.

Aggarwal, Charu C (2021). "Knowledge Graphs." In: *Artificial Intelligence*. Springer, pp. 409–435.

Aikins, Janice S (1979). "Prototypes and Production Rules: An Approach to Knowledge Representation for Hypothesis Formation." In: *IJCAI*. Vol. 6. Citeseer, pp. 1–3.

Akata, Zeynep et al. (2020). "A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect With Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence." In: *Computer* 53.8, pp. 18–28.

Al Machot, Fadi, Heinrich C Mayr, and Suneth Ranasinghe (2018). "A Hybrid reasoning approach for activity recognition based on answer set programming and Dempster–Shafer theory." In: *Recent Advances in Nonlinear Dynamics and Synchronization*. Springer, pp. 303–318.

Alola, Andrew Adewale (2012). "Analysis of possibility theory for reasoning under uncertainty." PhD thesis. Eastern Mediterranean University (EMU).

Alqahtani, Abdullah and Hina Gull (2018). "Cloud computing and security issues—a review of amazon web services." In: *Int J Appl Eng Res* 13.22, pp. 16077–16084.

Amgoud, Leila and Jonathan Ben-Naim (2018). "Evaluation of arguments in weighted bipolar graphs." In: *International Journal of Approximate Reasoning* 99, pp. 39–55.

Amirhosseini, Maziar and Juhana Salim (2019). "A Synthesis Survey of Ontology Evaluation Tools, Applications and Methods to Propose a Novel Branch in Evaluating the Structure of Ontologies: Graph-Independent Approach." In: *International Journal of Computer* 33.1, pp. 46–68.

Araújo Silva, Edson de et al. (2021). "A survey of Model Driven Engineering in robotics." In: *Journal of Computer Languages*, p. 101021.

Arnett, Dennis B, C Michael Wittmann, and John D Hansen (2021). "A process model of tacit knowledge transfer between sales and marketing." In: *Industrial Marketing Management* 93, pp. 259–269.

Ashley, Kevin D (2007). "Interpretive Reasoning with Hypothetical Cases." In: *FLAIRS Conference*, pp. 387–392.

Astorga-Vargas, Maria Angelica et al. (2017). "Explicit and tacit knowledge conversion effects, in software engineering undergraduate students." In: *Knowledge Management Research & Practice* 15.3, pp. 336–345.

Aubert, Jean-Pierre, J-F Baget, and Michel Chein (2006). "Simple conceptual graphs and simple concept graphs." In: *International Conference on Conceptual Structures*. Springer, pp. 87–101.

Baker, Michael et al. (1999). "The role of grounding in collaborative learning tasks." In: *Collaborative learning: Cognitive and computational approaches* 31, p. 63.

Baporikar, Neeta (2020). "Learning Link in Organizational Tacit Knowledge Creation and Dissemination." In: *International Journal of Sociotechnology and Knowledge Development (IJSKD)* 12.4, pp. 70–88.

Barley, William C, Jeffrey W Treem, and Paul M Leonardi (2020). "Experts at coordination: Examining the performance, production, and value of process expertise." In: *Journal of Communication* 70.1, pp. 60–89.

Baroroh, Dawi Karomati, Chih-Hsing Chu, and Lihui Wang (2020). "Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence." In: *Journal of Manufacturing Systems*. ISSN: 0278-6125.

Barrachina, Javier et al. (2012a). "Caova: A car accident ontology for vanets." In: *2012 IEEE wireless communications and networking conference (WCNC)*. Ieee, pp. 1864–1869.

– (2012b). "VEACON: A Vehicular Accident Ontology designed to improve safety on the roads." In: *Journal of Network and Computer Applications* 35.6, pp. 1891–1900.

Baskara, SN et al. (2019). "Influence of pavement condition towards accident number on Malaysian highway." In: *IOP Conference Series: Earth and Environmental Science*. Vol. 220. 1. IOP Publishing, p. 012008.

Bauters, Kim et al. (2012a). "Possibilistic answer set programming revisited." In: *arXiv preprint arXiv:1203.3466*.

– (2012b). "Possible and necessary answer sets of possibilistic answer set programs." In: *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. Vol. 1. IEEE, pp. 836–843.

Bergmann, R (2002). *Experience management : foundations, development methodology, and Internet-based applications*. Berlin New York: Springer. ISBN: 3-540-44191-3.

Besnard, Philippe and Anthony Hunter (2008). *Elements of argumentation*. Vol. 47. MIT press Cambridge.

Bettoni, Andrea et al. (2020). "Mutualistic and Adaptive Human-Machine Collaboration Based on Machine Learning in an Injection Moulding Manufacturing Line." In: *Procedia CIRP* 93, pp. 395–400.

Bock, Jürgen et al. (2008). "Benchmarking OWL reasoners." In: *ARea2008-Workshop on Advancing Reasoning on the Web: Scalability and Commonsense*. Tenerife Spain.

Brank, Janez, Marko Grobelnik, and Dunja Mladenic (2005). "A survey of ontology evaluation techniques." In: *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*. Citeseer Ljubljana Slovenia, pp. 166–170.

Bravo, G et al. (2022). "The influence of age on fatal work accidents and lost days in Chile between 2015 and 2019." In: *Safety Science* 147, p. 105599.

Brewka, Gerhard (1991). *Nonmonotonic reasoning: logical foundations of commonsense*. Vol. 12. Cambridge University Press.

Bromme, Rainer, Riklef Rambow, and Matthias Nückles (2001). "Expertise and estimating what other people know: the influence of professional experience and type of knowledge." In: *Journal of experimental psychology: Applied* 7.4, p. 317.

Buche, Patrice et al. (2019). "Expertise-based decision support for managing food quality in agri-food companies." In: *Computers and Electronics in Agriculture* 163, p. 104843.

Cabrera, Oscar, Xavier Franch, and Jordi Marco (2019). "3LConOnt: a three-level ontology for context modelling in context-aware computing." In: *Software & Systems Modeling* 18.2, pp. 1345–1378.

Calegari, Roberta et al. (2020). "Logic-Based Technologies for Intelligent Systems: State of the Art and Perspectives." In: *Information* 11.3, p. 167.

Chai, Joyce Y et al. (2016). "Collaborative language grounding toward situated human-robot dialogue." In: *AI Magazine* 37.4, pp. 32–45.

Chen, Xiaojun, Shengbin Jia, and Yang Xiang (2020). "A review: Knowledge reasoning over knowledge graph." In: *Expert Systems with Applications* 141, p. 112948.

Cheng, Justin et al. (2015). "Break it down: A comparison of macro-and microtasks." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 4061–4064.

Chuvikov, DA et al. (2019). "Mivar models of reconstruction and expertise of emergency events of road accidents." In: *IOP Conference Series: Materials Science and Engineering*. Vol. 534. 1. IOP Publishing, p. 012007.

Cimmino, Andrea, Alba Fernández-Izquierdo, and Raúl García-Castro (2020). "Astrea: automatic generation of SHACL shapes from ontologies." In: *European Semantic Web Conference*. Springer, pp. 497–513.

Cordier, Amélie, Béatrice Fuchs, and Alain Mille (2006). "Engineering and learning of adaptation knowledge in case-based reasoning." In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer, pp. 303–317.

Crnjac, Marina, Ivica Veža, and Nikola Banduka (2017). "From concept to the introduction of industry 4.0." In: *International Journal of Industrial Engineering and Management* 8.1, p. 21.

Croitoru, Madalina and Ernesto Compatangelo (2004). *On conceptual graph projection*. Tech. rep. Technical Report AUCS/TR0403, University of Aberdeen, UK, Department of . . .

Das, Subhashis and Pamela Hussey (2021). "ContSOnto: A Formal Ontology for Continuity of Care." In: *pHealth 2021*. IOS Press, pp. 82–87.

De Lope, Rafael P et al. (2021). "A novel UML-based methodology for modeling adventure-based educational games." In: *Entertainment Computing* 38, p. 100429.

De Nicola, Antonio, Michele Missikoff, and Roberto Navigli (2009). "A software engineering approach to ontology building." In: *Information systems* 34.2, pp. 258–275.

Delgrande, James P and Torsten Schaub (2000). "The role of default logic in knowledge representation." In: *Logic-based artificial intelligence*. Springer, pp. 107–126.

Dellermann, Dominik et al. (2019). "Hybrid intelligence." In: *Business & Information Systems Engineering* 61.5, pp. 637–643.

Denœux, Thierry, Didier Dubois, and Henri Prade (2020). "Representations of uncertainty in artificial intelligence: Probability and possibility." In: *A Guided Tour of Artificial Intelligence Research*. Springer, pp. 69–117.

Desouki, Abdelmoneim Amer, Michael Röder, and Axel-Cyrille Ngonga Ngomo (2019). "Ranking on Very Large Knowledge Graphs." In: *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, pp. 163–171.

Djurić, Dragan et al. (2004). "A UML profile for OWL ontologies." In: *Model driven architecture*. Springer, pp. 204–219.

Dodaro, Carmine and Marco Maratea (2017). "Nurse scheduling via answer set programming." In: *International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer, pp. 301–307.

Dörner, Dietrich and Joachim Funke (2017). "Complex problem solving: What it is and what it is not." In: *Frontiers in psychology* 8, p. 1153.

Doumbouya, Mamadou Bilo et al. (2018). "Argumentation graphs with constraint-based reasoning for collaborative expertise." In: *Future Generation Computer Systems* 81, pp. 16–29.
Dubois, Didier and Henry Prade (2015). "Possibility theory and its applications: Where do we stand?" In: *Springer handbook of computational intelligence*. Springer, pp. 31–60.
Dudkina, Ekaterina et al. (2021). "On node ranking in graphs." In: *arXiv preprint arXiv:2107.09487*.
Dulgheru, Valeriu (2012). "Technological evolution and creativity." In.
Dung, Phan Minh (1995). "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games." In: *Artificial intelligence* 77.2, pp. 321–357.
Durmaz, Ahmet, Hakan Demir, and Bülent Sezen (2021). "The role of negative entropy within supply chain sustainability." In: *Sustainable Production and Consumption* 28, pp. 218–230.
Economic Co-operation, Organisation for and Development (OECD) (2017). "PISA 2015 results (volume V): Collaborative problem solving, PISA." In.
El-Sappagh, Shaker H and Mohammed Elmogy (2015). "Case based reasoning: Case representation methodologies." In: *International Journal of Advanced Computer Science and Applications* 6.11, pp. 192–208.
Epstein, Susan L (2015). "Wanted: collaborative intelligence." In: *Artificial Intelligence* 221, pp. 36–45.
Ericsson, K Anders, Robert R Hoffman, and Aaron Kozbelt (2018). *The Cambridge handbook of expertise and expert performance*. Cambridge University Press.
Faci, Adam, Marie-Jeanne Lesot, and Claire Laudy (2021). "CG2A: Conceptual Graphs Generation Algorithm." In: *Joint Proceedings of the 19th World Congress of the International Fuzzy Systems Association (IFSA), the 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and the 11th International Summer School on Aggregation Operators (AGOP)*. Atlantis Press, pp. 63–70. ISBN: 978-94-6239-423-0. DOI: https://doi.org/10.2991/asum.k.210827.009. URL: https://doi.org/10.2991/asum.k.210827.009.
Fandinno, Jorge and Claudia Schulz (2019). "Answering the "why" in answer set programming–a survey of explanation approaches." In: *Theory and Practice of Logic Programming* 19.2, pp. 114–203.
Farrington-Darby, Trudi and John R Wilson (2006). "The nature of expertise: A review." In: *Applied ergonomics* 37.1, pp. 17–32.
Fionda, Valeria and Giuseppe Pirrò (2019). "Ontology: Definition Languages." In.
Frankish, Keith (2005). "Non-monotonic inference." In: *The Encyclopedia of Language and Linguistics* 2.
Fussell, Susan R et al. (2008). "How people anthropomorphize robots." In: *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, pp. 145–152.
Gašević, Dragan, Dragan Djurić, and Vladan Devedžić (2006). *Model driven architecture and ontology development*. Springer.
Gaševic, Dragan, Dragan Djuric, and Vladan Devedžic (2009). *Model driven engineering and ontology development*. Springer Science & Business Media. Chap. 4.
Gayo, Jose Emilio Labra et al. (2017). "Validating RDF data." In: *Synthesis Lectures on Semantic Web: Theory and Technology* 7.1, pp. 1–328.

Gebser, Martin et al. (2011). "Potassco: The Potsdam answer set solving collection." In: *Ai Communications* 24.2, pp. 107–124.
Gebser, Martin et al. (2012). "Answer set solving in practice." In: *Synthesis lectures on artificial intelligence and machine learning* 6.3, pp. 1–238.
Genesereth, Michael and Vinay K Chaudhri (2020). "Introduction to logic programming." In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14.1, pp. 1–219.
Giannoulis, Michail, Haridimos Kondylakis, and Emmanouil Marakakis (2019). "Designing and implementing a collaborative health knowledge system." In: *Expert Systems with Applications* 126, pp. 277–294.
Grigsby, Scott S (2018). "Artificial Intelligence for Advanced Human-Machine Symbiosis." In: *International Conference on Augmented Cognition*. Springer, pp. 255–266.
Grimm, Stephan and Boris Motik (2005). "Closed World Reasoning in the Semantic Web through Epistemic Operators." In: *OWLED*.
Gruber, Thomas R (1993). "A translation approach to portable ontology specifications." In: *Knowledge acquisition* 5.2, pp. 199–220.
Guergour, Habib-Ellah, Razika Driouche, and Zizette Boufaïda (2006). "An Approach for Application Ontology Building and Integration Enactment." In: *SWAP*.
Guermah, Hatim et al. (2014). "An ontology oriented architecture for context aware services adaptation." In: *arXiv preprint arXiv:1404.3280*.
Guizzardi, Giancarlo et al. (2021). "UFO: Unified Foundational Ontology." In: *Applied Ontology* Preprint, pp. 1–44.
Hassan, Mariam M and Hoda MO Mokhtar (2021). "AutismOnt: An Ontology-Driven Decision Support For Autism Diagnosis and Treatment." In: *Egyptian Informatics Journal*.
Hoffman, Kerry (2007). "A comparison of decision-making by "expert" and "novice" nurses in the clinical setting, monitoring patient haemodynamic status post abdominal aortic aneurysm surgery." PhD thesis.
Horrocks, Ian (2005). "Owl: A description logic based ontology language." In: *International conference on principles and practice of constraint programming*. Springer, pp. 5–8.
Huang, Chun-Che and Chia-Ming Kuo (2003). "The transformation and search of semi-structured knowledge in organizations." In: *Journal of Knowledge management*.
Hur, Ali, Naeem Janjua, and Mohiuddin Ahmed (2021). "A Survey on State-of-the-art Techniques for Knowledge Graphs Construction and Challenges ahead." In: *arXiv preprint arXiv:2110.08012*.
Hurley, Patrick J (2014). *A concise introduction to logic*. Nelson Education, pp. 609–631.
Ikujiro Nonaka, HT (1995). *The Knowledge-Creating Company-Ikujiro Nonaka*.
Jain, Sarika (2021). *Understanding semantics-based decision support*. CRC Press. Chap. 4, pp. 67–84.
Janssen, Jeroen et al. (2012). *Answer Set Programming for Continuous Domains: A Fuzzy Logic Approach*. Vol. 5. Springer Science & Business Media.
Jean, Stéphane, Guy Pierra, and Yamine Ait-Ameur (2007). "Domain ontologies: A database-oriented analysis." In: *Web Information Systems and Technologies*. Springer, pp. 238–254.

Jetlund, Knut, Erling Onstein, and Lizhen Huang (2019). "Adapted rules for UML modelling of geospatial information for model-driven implementation as OWL ontologies." In: *ISPRS International Journal of Geo-Information* 8.9, p. 365.

Johnson, Marina et al. (2022). "Integrating human knowledge into artificial intelligence for complex and ill-structured problems: Informed artificial intelligence." In: *International Journal of Information Management* 64, p. 102479.

Jonassen, David H (1997). "Instructional design models for well-structured and III-structured problem-solving learning outcomes." In: *Educational technology research and development* 45.1, pp. 65–94.

Kaindl, Hermann et al. (2016). "A core ontology of safety risk concepts." In: *Human-Centered and Error-Resilient Systems Development*. Springer, pp. 165–180.

Kakas, Antonis C (1994). "Default reasoning via negation as failure." In: *Foundations of Knowledge Representation and Reasoning*. Springer, pp. 160–178.

Karhu, Katja (2002). "Expertise cycle–an advanced method for sharing expertise." In: *Journal of Intellectual Capital*.

Kell, Douglas B and Stephen G Oliver (2004). "Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era." In: *Bioessays* 26.1, pp. 99–105.

Kim, Joo Yeun and Kyu Yon Lim (2019). "Promoting learning in online, ill-structured problem solving: The effects of scaffolding type and metacognition level." In: *Computers & Education* 138, pp. 116–129.

Koch, Janin and Antti Oulasvirta (2018). "Group cognition and collaborative ai." In: *Human and Machine Learning*. Springer, pp. 293–312.

Kogut, Paul et al. (2002). "UML for ontology development." In: *The Knowledge Engineering Review* 17.1, pp. 61–64.

Kolodner, Janet L (1983). "Towards an understanding of the role of experience in the evolution from novice to expert." In: *International Journal of Man-Machine Studies* 19.5, pp. 497–518.

Kowalski, Robert (1974). "Predicate logic as programming language." In: *IFIP congress*. Vol. 74, pp. 569–544.

Křemen, Petr et al. (2017). "Ontological foundations of european coordination centre for accident and incident reporting systems." In: *Journal of Aerospace Information Systems* 14.5, pp. 279–292.

Lan, Jibin et al. (2015). "A new linguistic aggregation operator and its application to multiple attribute decision making." In: *Operations Research Perspectives* 2, pp. 156–164.

Ławrynowicz, Agnieszka (2020). "Creative AI: A new avenue for the Semantic Web?" In: *Semantic Web* 11.1, pp. 69–78.

Liao, Huchang et al. (2018). "Hesitant fuzzy linguistic term set and its application in decision making: a state-of-the-art survey." In: *International Journal of Fuzzy Systems* 20.7, pp. 2084–2110.

Liu, Liping and Ronald R Yager (2008). "Classic works of the Dempster-Shafer theory of belief functions: An introduction." In: *Classic works of the Dempster-Shafer theory of belief functions*. Springer, pp. 1–34.

Lloyd, John W (2012). *Foundations of logic programming*. Springer Science & Business Media.

Longard, Lukas, Lara Schiborr, and Joachim Metternich (2022). "Potentials and obstacles of the use of data mining in problem-solving processes." In: *Procedia CIRP* 107, pp. 252–257.

López, Beatriz (2013). "Case-based reasoning: a concise introduction." In: *Synthesis lectures on artificial intelligence and machine learning* 7.1, pp. 1–103.

Lu, Ying-Jin and Jun He (2017). *Dempster-shafer evidence theory and study of some key problems*. Infinite Study.

MOF, OMG (2015). *OMG Meta Object Facility (MOF) Core Specification. Version 2.4. 2. April 2014*.

Maalel, Ahmed et al. (2012). "Towards an ontology of help to the modeling of accident scenarii: Application on railroad transport." In: *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*. IEEE, pp. 1–6.

Malgouyres, Hugues and Gilles Motet (2006). "A UML model consistency verification approach based on meta-modeling formalization." In: *Proceedings of the 2006 ACM symposium on Applied computing*, pp. 1804–1809.

Malo, Andy et al. (2013). "Consolidation des données statistiques par expertise et similarité pour la prévision des ventes." In: *QUALITA2013*.

Mariano-Hernández, D et al. (2021). "A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis." In: *Journal of Building Engineering* 33, p. 101692.

Mario Raich Simon L. Dolan, Claudiio Cillulo and Bonnie A. Richley (2019). *Beyond Collaborative Intelligence we can see a Meta-Mind Society Surfacing and we can Dream of a Ω-Mind?* [Online; accessed 27-May-2020].

Marquis, Pierre, Odile Papini, and Henri Prade (2014a). "Panorama de l'intelligence artificielle." In: *Ses Bases Méthodologiques, ses Développements* 1, pp. 555–590.

– (2014b). "Panorama de l'intelligence artificielle." In: *Ses Bases Méthodologiques, ses Développements* 1.

Martínez-Costa, Catalina and Stefan Schulz (2017). "Validating EHR clinical models using ontology patterns." In: *Journal of biomedical informatics* 76, pp. 124–137.

Martínez-García, Jose R et al. (2020). "Ontology for knowledge condensation to support expertise location in the code phase during software development process." In: *IET Software* 14.3, pp. 234–241.

Mascardi, Viviana, Valentina Cordì, and Paolo Rosso (2007). "A Comparison of Upper Ontologies." In: *Woa*. Vol. 2007. Citeseer, pp. 55–64.

Meister, Maximilian et al. (2018). "Problem-solving process design in production: Current progress and action required." In: *Procedia CIRP* 78, pp. 376–381.

Meister, Maximilian et al. (2019). "Manufacturing Analytics for problem-solving processes in production." In: *Procedia CIRP* 81, pp. 1–6.

Meléndez, Diana et al. (2018). "Formalization and reuse of collaboration experiences in industrial processes." In: *International Conference on Complex Systems Design & Management*. Springer, pp. 157–167.

Minutolo, Aniello, Massimo Esposito, and Giuseppe De Pietro (2016). "A Hypothetical Reasoning System for Mobile Health and Wellness Applications." In: *International Conference on Wireless Mobile Communication and Healthcare*. Springer, pp. 278–286.

Moll, Ian (2021). "The Myth of the Fourth Industrial Revolution." In: *Theoria* 68.167, pp. 1–38.

Molnár, Gyöngyvér, Saleh Ahmad Alrababah, and Samuel Greiff (2022). "How we explore, interpret, and solve complex problems: A cross-national study of problem-solving processes." In: *Heliyon* 8.1, e08775.

Morveli-Espinoza, Mariela et al. (2019). "An argumentation-based approach for identifying and dealing with incompatibilities among procedural goals." In: *International Journal of Approximate Reasoning* 105, pp. 1–26.

Mugnier, Marie-Laure (2000). "Knowledge representation and reasonings based on graph homomorphism." In: *International Conference on Conceptual Structures*. Springer, pp. 172–192.

Musen, Mark A. and The Protégé Team (2013). "Protégé Ontology Editor." In: *Encyclopedia of Systems Biology*. Ed. by Werner Dubitzky et al. New York, NY: Springer New York, pp. 1763–1765. ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_1104. URL: https://doi.org/10.1007/978-1-4419-9863-7_1104.

Naubourg, Pierre et al. (2011). "A approach to clinical proteomics data quality control and import." In: *International Conference on Information Technology in Bio-and Medical Informatics*. Springer, pp. 168–182.

Navarro, Charlotte and Nathalie Colbach (2020). *Méthodes d'expertise-Comment les utiliser?*

Nebel, Bernhard (2001). "Logics for knowledge representation." In: *International Encyclopedia of the Social and Behavioral Sciences, Kluwer, Dordrecht*.

Nebel, Bernhard, Charles Rich, and William R Swartout (1992). *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92)*. Morgan Kaufmann Pub.

Nicolas, Pascal et al. (2006). "Possibilistic uncertainty handling for answer set programming." In: *Annals of Mathematics and Artificial Intelligence* 47.1, pp. 139–181.

Niemelä, Ilkka (1999). "Logic programs with stable model semantics as a constraint programming paradigm." In: *Annals of mathematics and Artificial Intelligence* 25.3, pp. 241–273.

Nokes, Timothy J, Christian D Schunn, and Michelene Chi (2010). "Problem solving and human expertise." In: *International encyclopedia of education*. Elsevier Ltd, pp. 265–272.

Noy, Natalya F, Deborah L McGuinness, et al. (2001). *Ontology development 101: A guide to creating your first ontology*.

Núnez, Rafael C et al. (2013). "Modeling uncertainty in first-order logic: a Dempster-Shafer theoretic approach." In: *8th International Symposium on Imprecise Probability: Theories and Applications*.

Núñez, Rafael C et al. (2018). "Uncertain Logic Processing: logic-based inference and reasoning using Dempster–Shafer models." In: *International Journal of Approximate Reasoning* 95, pp. 1–21.

ODM, OMG (2007). "Ontology Definition Metamodel: OMG Adopted Specification." In: *Object Management Group* 26.05, p. 2008.

Omri, Asma et al. (2018). "Towards an understanding of cloud services under uncertainty: a possibilistic approach." In: *International Journal of Approximate Reasoning* 98, pp. 146–162.

Ossmy, Ori et al. (2022). "First "Where" and then "How": Developmental Processes in Exploring Solutions to Problems with Hidden Demands." In.

Panagiotopoulos, Ioannis et al. (2012). "An ontological approach for domain knowledge modeling and management in e-learning systems." In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, pp. 95–104.

Pang, Qi, Hai Wang, and Zeshui Xu (2016). "Probabilistic linguistic term sets in multi-attribute group decision making." In: *Information Sciences* 369, pp. 128–143.

Paolone, Gaetanino et al. (2020). "Automatic Code Generation of MVC Web Applications." In: *Computers* 9.3, p. 56.

Paraschiv, Florentina and Georgios Sermpinis (2021). "A Data-driven Explainable Case-based Reasoning Approach for Financial Risk Detection Wei Li." In: *IRTG 1792 Discussion Paper 2021-010*. International Research Training Group 1792.

Pareti, Paolo and George Konstantinidis (2021). "A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs." In: *arXiv preprint arXiv:2112.01441*.

Parsons, Simon and Geoffrey Parsons (2001). *Qualitative methods for reasoning under uncertainty*. Vol. 13. Mit Press.

Paschen, Jeannette, Matthew Wilson, and João J Ferreira (2020). "Collaborative intelligence: How human and artificial intelligence create value along the B2B sales funnel." In: *Business Horizons*.

Patalas-Maliszewska, Justyna and Sławomir Kłos (2019). "An Application for the Acquisition and Understanding of Expert Knowledge in a Maintenance Department." In: *IFAC-PapersOnLine* 52.13, pp. 832–837.

Peyrouty, Paul (2010). "La norme NF X 50-110 «qualité en expertise»: situation actuelle et perspectives." In: *Revue Experts* 90, pp. 36–39.

Pierre Peyrouty Chanay, Fourniguet Huver Loisel Peyrouty-Pineau-Tuffery (2011). *Recommandations pour l'application de la norme NF X 50-110:2003*. Tech. rep. Association Française de Normalisation.

Poveda-Villalón, María, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa (2014). "OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation." In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 10.2, pp. 7–34.

Prasad, TV et al. (2012). "Hybrid systems for knowledge representation in artificial intelligence." In: *arXiv preprint arXiv:1211.2736*.

Pupkov, KA (2019). "Intelligent systems and human being." In: *Procedia Computer Science* 150, pp. 540–543.

Qin, Hao, Hongwei Wang, and Aylmer L Johnson (2017). "A RFBSE model for capturing engineers' useful knowledge and experience during the design process." In: *Robotics and Computer-Integrated Manufacturing* 44, pp. 30–43.

Raisin, Syarfa Najihah et al. (2020). "Cyber-Physical System (CPS) Application-A REVIEW." In: *REKA ELKOMIKA: Jurnal Pengabdian kepada Masyarakat* 1.2, pp. 52–65.

Reineking, Thomas (2014). "Belief functions: theory and algorithms." PhD thesis. Universität Bremen.

Reiter, Raymond (1988). "Nonmonotonic reasoning." In: *Exploring artificial intelligence*, pp. 439–481.

Rescher, Nicholas (1968). "Hypothetical Reasoning: Studies in Logic and the Foundations of Mathematics." In.

Riedl, Mark O (2019). "Human-centered artificial intelligence and machine learning." In: *Human Behavior and Emerging Technologies* 1.1, pp. 33–36.

Riguzzi, Fabrizio (2018). *Foundations of Probabilistic Logic Programming*. River Publishers.

Rittgen, Peter (2007). *Handbook of ontologies for business interaction*. IGI Global. Chap. 1.

Roventa, Eugene and Tiberiu Spircu (2009). *Management of knowledge imperfection in building intelligent systems*. Springer.

Ruiz, Paula Potes, Bernard Kamsu Foguem, and Bernard Grabot (2014). "Generating knowledge in maintenance from Experience Feedback." In: *Knowledge-Based Systems* 68, pp. 4–20.

Russell, Stuart J (2021). *Artificial intelligence a modern approach 4th edition*. Pearson Education, Inc. Chap. 7.

Salas, Eduardo, Michael A Rosen, and Deborah DiazGranados (2010). "Expertise-based intuition and decision making in organizations." In: *Journal of management* 36.4, pp. 941–973.

Salicone, Simona (2007). *Measurement uncertainty: an approach via the mathematical theory of evidence*. Springer science & business media. Chap. 1.

Sanctorum, Audrey et al. (2022). "End-user engineering of ontology-based knowledge bases." In: *Behaviour & Information Technology*, pp. 1–19.

Schalkwyk, Pieter van, S Lin, and Dr Somayeh Malakuti (2019). "A short introduction to digital twins." In: *Ind. Internet Consortium J. Innov.*

Schreiber, Guus (2008). "Knowledge engineering." In: *Foundations of Artificial Intelligence* 3, pp. 929–946.

Schwind, Camilla and Pierre Siegel (1994). "A modal logic for hypothesis theory." In: *Fundamenta Informaticae* 21.1, 2, pp. 89–101.

Shafer, Glenn (1976). *A mathematical theory of evidence*. Vol. 42. Princeton university press.

– (1986). "Probability judgment in artificial intelligence." In: *Machine Intelligence and Pattern Recognition*. Vol. 4. Elsevier, pp. 127–135.

Shafiq, Syed Imran et al. (2014). "Implementing virtual engineering objects (VEO) with the set of experience knowledge structure (SOEKS)." In: *Procedia Computer Science* 35, pp. 644–652.

Shafiq, Syed Imran et al. (2015). "Virtual engineering object (VEO): Toward experience-based design and manufacturing for industry 4.0." In: *Cybernetics and Systems* 46.1-2, pp. 35–50.

Shafiq, Syed Imran et al. (2017). "Towards an experience based collective computational intelligence for manufacturing." In: *Future Generation Computer Systems* 66, pp. 89–99.

Shaw, Mildred LG and Brian R Gaines (2005). "Expertise and expert systems: emulating psychological processes." In: *The Essential Practitioner's Handbook of Personal Construct Psychology. New York: John Wiley & Sons Inc.*

Shepperd, Martin (2003). "Case-based reasoning and software engineering." In: *Managing Software Engineering Knowledge*. Springer, pp. 181–198.

Siegel, Pierre and Camilla Schwind (1993). "Modal logic based theory for non-monotonic reasoning." In: *Journal of Applied Non-Classical Logics* 3.1, pp. 73–92.

Siegel, Pierre et al. (2017). "Vers une représentation des systèmes dynamiques booléens en logique des hypothèses." In: *Journées d'Intelligence Artificielle Fondamentale 2017*, 14p.

Siegel, Pierre et al. (2020). "Logique Modale des Hypothèses, Systèmes Dynamiques Booléens et Réseaux de gênes." In: *JIAF*, p. 43.

Sizov, Gleb and Pinar Öztürk (2013). "Automatic extraction of reasoning chains from textual reports." In: *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, pp. 61–69.

Song, Bo, Zuhua Jiang, and Lijun Liu (2016). "Automated experiential engineering knowledge acquisition through Q&A contextualization and transformation." In: *Advanced Engineering Informatics* 30.3, pp. 467–480.

Sony, Michael (2020). "Design of cyber physical system architecture for industry 4.0 through lean six sigma: Conceptual foundations and research issues." In: *Production & Manufacturing Research* 8.1, pp. 158–181.

Souza, Raphael, Helio Ferenhof, and Fernando Forcellini (2022). "Industry 4.0 and Industry 5.0 from the Lean perspective." In: *International Journal of Management, Knowledge and Learning* 11.

Sowa, Konrad, Aleksandra Przegalinska, and Leon Ciechanowski (2021). "Cobots in knowledge work: Human – AI collaboration in managerial professions." In: *Journal of Business Research* 125, pp. 135–142.

Stone, Peter et al. (2016). "Artificial Intelligence and Life in 2030. One hundred year study on artificial intelligence: Report of the 2015-2016 Study Panel." In: *Stanford University, Stanford, CA, http://ai100. stanford. edu/2016-report. Accessed: September* 6, p. 2016.

Suhail, Sabah et al. (2022). "Towards situational aware cyber-physical systems: A security-enhancing use case of blockchain-based digital twins." In: *Computers in Industry* 141, p. 103699.

Sun, Zhaohao (2004). "A waterfall model for knowledge management and experience management." In: *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*. IEEE, pp. 472–475.

Sun, Zhaohao and Gavin Finnie (2003). "Brain-like architecture and experience-based reasoning." In: *Proc. 7th Joint Conf on Information Sciences (JCIS)*, pp. 26–30.

– (2004). "Experience based reasoning for recognising fraud and deception." In: *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*. IEEE, pp. 80–85.

Suran, Shweta, Vishwajeet Pattanaik, and Dirk Draheim (2020). "Frameworks for Collective Intelligence: A Systematic Literature Review." In: *ACM Computing Surveys (CSUR)* 53.1, pp. 1–36.

Taj, Imran and Noor Zaman (2022). "Towards Industrial Revolution 5.0 and Explainable Artificial Intelligence: Challenges and Opportunities." In: *International Journal of Computing and Digital Systems* 12.1, pp. 295–320.

Tao, Fei et al. (2019). "Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison." In: *Engineering* 5.4, pp. 653–661.

Tian, Ling et al. (2022). "Knowledge graph and knowledge reasoning: A systematic review." In: *Journal of Electronic Science and Technology*, p. 100159.

Trochim, William MK and James P Donnelly (2001). *Research methods knowledge base*. Vol. 2. Atomic Dog Publishing Cincinnati, OH.

Upadhyay, Nitin (2017). "Managing cloud service evaluation and selection." In: *Procedia computer science* 122, pp. 1061–1068.

Uschold, Michael and Martin King (1995). *Towards a methodology for building ontologies*. Citeseer.

Uschold, Mike and Michael Gruninger (1996). "Ontologies: Principles, methods and applications." In: *The knowledge engineering review* 11.2, pp. 93–136.
Van Den Brink, René and Agnieszka Rusinowska (2021). "The degree ratio ranking method for directed graphs." In: *European Journal of Operational Research* 288.2, pp. 563–575.
Vartolomei, VC and S Avasilcai (2019). "Challenges of digitalization process in different industries. Before and after." In: *IOP Conference Series: Materials Science and Engineering*. Vol. 568. 1. IOP Publishing, p. 012086.
Vo, Minh Hoang Lien and Quang Hoang (2020). "Transformation of UML class diagram into OWL Ontology." In: *Journal of Information and Telecommunication* 4.1, pp. 1–16.
Voorbraak, Frans (1991). "A preferential model semantics for default logic." In: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer, pp. 344–351.
Walker, Vern R (2006). "A Default-Logic Framework for Legal Reasoning in Multiagent Systems." In: *AAAI Fall Symposium: Interaction and Emergent Phenomena in Societies of Agents*, pp. 88–95.
Wang, Jing and Xin Wang (2011). "An ontology-based traffic accident risk mapping framework." In: *International Symposium on Spatial and Temporal Databases*. Springer, pp. 21–38.
Waris, Mohammad Maqbool, Cesar Sanin, and Edward Szczerbicki (2017). "Smart innovation process enhancement using SOEKS and decisional DNA." In: *Journal of Information and Telecommunication* 1.3, pp. 290–303.
Wieten, Sarah (2018). "Expertise in evidence-based medicine: a tale of three models." In: *Philosophy, ethics, and humanities in medicine* 13.1, pp. 1–7.
Wu, Haitao et al. (2020). "An Ontological Metro Accident Case Retrieval Using CBR and NLP." In: *Applied Sciences* 10.15, p. 5298.
Xu, Zeshui (2004). "Uncertain linguistic aggregation operators based approach to multiple attribute group decision making under uncertain linguistic environment." In: *Information sciences* 168.1-4, pp. 171–184.
– (2005). "Deviation measures of linguistic preference relations in group decision making." In: *Omega* 33.3, pp. 249–254.
– (2012). *Linguistic decision making*. Springer. Chap. 1.
Yager, Ronald R (1995). "An approach to ordinal decision making." In: *International Journal of Approximate Reasoning* 12.3-4, pp. 237–261.
Yager, Ronald R and Liping Liu (2008). *Classic works of the Dempster-Shafer theory of belief functions*. Vol. 219. Springer.
Yang, Maolin, Weidong Li, and Pingyu Jiang (2021). "A collective intelligence oriented three-layer framework for socialized and collaborative product design." In: *Expert Systems with Applications* 173, p. 114742.
Zang, Liang-Jun et al. (2013). "A survey of commonsense knowledge acquisition." In: *Journal of Computer Science and Technology* 28.4, pp. 689–719.
Zheng, Xun et al. (2018). "Dags with no tears: Continuous optimization for structure learning." In: *Advances in Neural Information Processing Systems* 31.
Zhong, BT et al. (2015). "An ontological approach for technical plan definition and verification in construction." In: *Automation in Construction* 55, pp. 47–57.

DECLARATION

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

France, Winter 2022

Serge SONFACK
SOUNCHIO