INNOSALE

Innovating Sales and Planning of Complex Industrial Products
Exploiting Artificial Intelligence

Deliverable 3.1
Knowledge Model

Deliverable type:	Document
Deliverable reference number:	ITEA 20054 | D3.1
Related Work Package:	WP 3
Due date:	2024-05-31
Actual submission date:	2024-08-31
Responsible organisation:	Panel
Editor:	Yazmin Andrea Pabón Guerrero
Dissemination level:	Confidential / Public
Revision:	Final

Abstract:	The Knowledge Model employs ontologies to represent concepts within the various application domains of InnoSale. This foundational ontology will enhance semantic search and support the knowledge base in subsequent tasks.
Keywords:	Knowledge model, ontologies, OWL.

Table_head	Name 1 (partner)	Name 2 (partner)	Approval date (1 / 2)
Approval at WP level			
Veto Review by All Partners			

Editor

Yazmin Andrea Pabón Guerrero (Panel)

Sepideh Sobhgol (IFAK)

Contributors

José Cristobal López Zafra (Vector)

Executive Summary

This deliverable is dedicated to T3.1, a specification of an ontology concerning the InnoSale domains. This ontology will be used later on to improve the semantic search in T3.2 and knowledge base in T3.3. This deliverable explains why we need an ontology and two approaches to how we want to create it: semi-automatic approach and upper/domain ontology.

Table of Content

Figures

Tables

1 Introduction

In sales processes, a sales engineer of a manufacturer often wants to find previous offer or project folders, which are similar to the current inquiry of a customer. In InnoSale, a simple word index based search in the project folders shall be replaced by a semantic search, which also takes synonym and abstract terms into account. Knowledge Model (KM) allows to represent the concepts related to the different application domains of InnoSale and to support the development of the semantic search and Knowledge Base components.

A key aspect of the KM is the use of ontologies. In InnoSale, we propose two approaches to represent the knowledge model: semi-automatic approach creating an ontology with regard to the material handling domain and upper/domain approach represented in OWL, a W3C recommendation with regard to the digital products domain. The ontology based semantic search engine will be developed in T3.2.

In this deliverable, it is explained why we need to build an ontology and how we want to create it. Building an ontology requires much manual work. Therefore, we want to reduce that work of domain experts by creating the ontology semi-automatically. This approach is partially based on employing natural language processing (NLP) techniques to help us to generate some layers of the ontology. Our ontology will be in the format of a file which unifies the terminologies of Demag and Konecranes. Besides, an Ontology Editor will be developed, which provides a user interface (UI) for manually editing the ontology.

2 Definitions

2.1 Ontologies

Ontologies are related to different meaning of terms and relations between them [1]. They provide a structure and a common vocabulary for organizing a domain's knowledge and sharing that information. All concepts within an ontology are interconnected through relationships and can be interpreted by a machine. Some of the reasons for using ontologies include:

- Sharing a common understanding of the information structure among people or software agents.
- Enabling the reuse of domain knowledge.
- Making domain assumptions explicit.
- Separating domain knowledge from operational knowledge.
- Analyzing domain knowledge.

Ontologies function as semantic networks representing the concepts of the extralinguistic world. Through relationships between concepts, they help identify the context of a term, which is essential for understanding its meaning in discourse. By representing concepts (not merely terms), ontologies are particularly useful in machine translation systems, where each concept can be associated with the linguistic forms representing it in different languages.

An ontology provides a vocabulary of classes and relationships to describe a domain, emphasizing knowledge sharing and consensus in its representation. For example, an ontology about art might include classes such as Painter, Painting, Style, or Museum, and relationships like the author of a painting, painters belonging to an artistic style, or works located in a museum.

Ontologies have the following components to represent the knowledge of a domain:

- Concepts: These are the basic ideas being formalized. Concepts can be classes of objects, methods, plans, strategies, reasoning processes, etc.
- Relationships: These represent the interaction and linkage between domain concepts. They often form the domain's taxonomy. For example: subclass-of, part-of, exhaustive-part-of, connected-to, etc.
- Functions: A specific type of relationship where an element is identified by calculating a function that considers several elements of the ontology. For instance, functions such as categorize-class, assign-date, etc., may appear.
- Instances: These are used to represent specific objects of a concept.
- Axioms: These are theorems declared about relationships that the elements of the ontology must satisfy. For example: "If A and B are of class C, then A is not a subclass of B," "For every A that meets condition C, A is B," etc.

We can create a knowledge base by defining individual instances of these classes, specifying the specific values of the slots and additional constraints on the slots. In summary, an ontology is a knowledge representation system that results from selecting a domain or area of knowledge and applying a method to obtain a formal representation of the concepts it contains and the relationships that exist between those concepts.

Moreover, an ontology is constructed with respect to a usage context. This means that an ontology specifies a conceptualization or a way of viewing the world, so each ontology incorporates a point of view. Additionally, an ontology contains definitions that provide the vocabulary to refer to a domain, and these definitions depend on the language used to describe them. All conceptualizations (definitions, categorizations, hierarchies, properties, inheritance, etc.) of an ontology can be machine-processable.

Ontologies can be developed using top-down or bottom-up approaches. The bottom-up approach starts with the most specific concepts in a domain of application. Top-down approaches start with high-level concepts that are assumed to be common to many application areas [2].

We can find upper and domain ontologies. "The upper ontologies are domain-independent, and gives a general description of the concepts. Domain ontologies can be constructed with the upper ontologies and are used to state-specific information about their domains, or their situation[3].

2.2 Building an Ontology

Building an ontology usually requires much manual work. No direct procedure is available to offer a full automatic process of modelling an ontology. Figure 1 shows different layers of learning and building an ontology. Taking usage of natural language processing (NLP) techniques will help partially to build an ontology in bottom layers of the ontology learning layered cake model introduced by [4]. In the upper layers, the existence of concepts and relations between concepts are based on the terminology and terms with similar meaning, which is represented in the bottom layers of the model. The idea of InnoSale is to facilitate the process of selling products and reduce the amount of work needed to be done by sales engineers. Hence, it would be more helpful if the terms and their synonyms get generated at least semi-automatically. The necessary term extraction could be partially addressed by NLP.

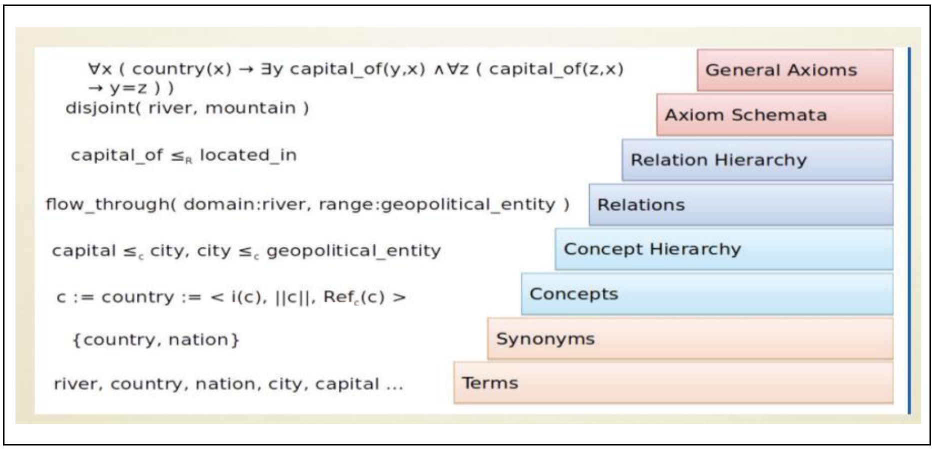

Figure 1: Ontology Learning Layer Cake according to [4]

In the InnoSale project, we are going to build an ontology which includes keywords, synonyms and abstractions by utilizing Demag and Konecranes terminologies and evaluating historical inquiries and offers.
In a first step, our ontology will be a data structure which unifies the terminologies of Demag and Konecranes by relating concepts to their synonyms or finding a concept hierarchy.
Existing project files are additional sources of text, which need to be evaluated for new terms which are added to the initial unified ontology.
Further-on, the unified ontology also needs to be updated by new terms, which are extracted from incoming inquiries and which cannot be found in the original version of the unified ontology.

2.3 Semantic Search

In sales processes, a sales engineer of a manufacturer often wants to find previous offer or project folders, which are similar to the current inquiry of a customer. In InnoSale, a simple word index based search in the project folders shall be replaced by a semantic search, which also takes synonym and abstract terms into account. There are different styles of semantic search possible, for example a word embedding based approach or an ontology based approach.

Semantic search is a document retrieval process that goes beyond simply relying on word occurrences in documents. Instead, it leverages domain knowledge, which can be represented through an ontology -a formal specification of concepts and their relationships [5].

Ontology-based semantic search involves using ontologies, which are formal representations of knowledge that define concepts, relationships, and properties in a specific domain, to enhance the accuracy and relevance of search results. By incorporating ontologies, semantic search aims to grasp the meaning and context of user queries and the content being searched. In this approach, the search engine analyses both the query and the data against the ontology, enabling it to interpret the query's intent and uncover semantic relationships between concepts. This understanding of semantics allows the search engine to provide more precise and contextually relevant search results. The use of ontologies in semantic search offers more advanced search capabilities compared to traditional keyword-based search. It enables the search engine to consider related concepts, synonyms, hierarchical relationships, and other semantic connections, resulting in more accurate and comprehensive search outcomes. Ontology-based semantic search is particularly advantageous in domains with intricate and specialized terminology, where comprehending the semantic context plays a crucial role in retrieving pertinent information[6][7].

The connection between documents and ontologies plays a crucial role in semantic search approaches. There are two main approaches: tight coupling and loose coupling. In tight coupling, documents explicitly refer to concepts in the ontology, making it easier to resolve homonymies. However, it requires significant effort in annotating documents with semantic information. On the other hand, in loose coupling, documents are not bound to a specific ontology, which presents the challenge of selecting the appropriate ontology. While loose coupling provides flexibility, it has limitations in terms of semantic resolution, especially in scenarios like the World Wide Web. Ontology-based semantic search engines rely on the structure of ontologies, which consist of concepts, properties, constraints, and axioms. Standard properties, including synonym_of, hypernym_of, meronym_of, instance_of, and negation_of, are used to capture relationships in semantic search based on common sense. These properties enhance the search capabilities but also introduce dependencies on the structure of ontologies [5]. Regarding the tight coupling approach, document annotation is the process of identifying and marking up specific elements or information within a document for various purposes, such as information retrieval, data extraction, or semantic understanding [8]. The specific types and methods of annotation can vary depending on the context, purpose, and tools used for annotation. Some types of document annotation include: Named Entity Recognition: NER involves identifying and categorizing named entities (e.g., person names, locations, organizations) in a document [9], Sentiment Analysis: Sentiment analysis aims to determine the sentiment or opinion expressed in a document, often categorized as positive, negative, or neutral [10], Topic Modeling: Topic modeling identifies latent topics or themes within a document collection and assigns documents to those topics [11], Text Classification: Text classification involves categorizing documents into predefined classes or categories based on their content [12], Named Entity Linking (NEL): NEL involves linking named entities mentioned in a document to their corresponding entities in a knowledge base or reference source[13], Semantic Role Labelling (SRL): SRL aims to identify and classify the semantic roles of entities and predicates in a sentence or document [14], Coreference Resolution: Coreference resolution deals with identifying and linking expressions that refer to the same entity within a document [15].

Hence, when engaging in ontology-based semantic search, it is imperative to link the entities within documents to those in a knowledge base. Given that we are establishing the concepts within our ontology database, we opt for a different approach by linking the shared tokens or entities between documents and the concepts already present in the ontology. This process constitutes the annotation of documents. In the next section, we explain how we store the connection between concepts within documents and ontology database.

The component to be developed here, will support an ontology-based semantic search approach, especially the creation process for the necessary ontology. The considered project partners already maintained and still maintain different terminologies, which define the vocabulary to be used for naming of products and parts in their projects. The terminologies shall be unified into a single terminology. It will cover a broader set of terms used by the technical experts than one of the existing terminologies. Duplicates need to be removed.

In a second step, the ontology shall constantly be updated. Customers of the manufacturers often use different or sometimes casual terms to describe what they need. Thus, if an inquiry email reaches the sales engineer, it should be analysed for relevant terms, which are then semi-automatically related to other terms of the ontology. Finally, the extended unified ontology will cover the terms of experts, which are related to terms of customers. This builds the foundation for a powerful semantic search engine, which is described in another component specification. For manual editing the ontology, an Ontology Editor will be developed which provides a graphical user interface for this task. We describe here the foundation of that tool.

3 Semi-automatic definition of an ontology for material handling

3.1 Overview

This section represents different steps towards creating ontology. The ontology should be created in following steps:

- Import of existing terminologies: The vocabulary, which should be used by the manufacturers experts is defined in terminologies. Demag especially has an **Excel-file** and an **Acrolinx database** for that purpose, while the Acrolinx database is shared with Konecranes. These terminologies contain translations into different languages. The Acrolinx database additionally may contain synonyms of terms. The existing terminologies should be unified.
- The **unified terminology** should integrate term abstractions in order to make it possible to find product variants too. It is then called an **ontology**.
- Optionally update of the ontology by evaluation of existing project files: existing project files need to be scanned for terms, which are not yet in the ontology. Those terms need to be added and possibly need to be related to existing terms. Even a black list of words should be maintained, which contains words, which frequently occur, but have no influence on the selection of previous project folders.
- Regular update based on incoming inquiries: The manufacturer gets inquiry emails from customers, who use a different vocabulary than used by experts in previous projects. There should be an automatically triggered process to update and extend the ontology whenever an inquiry email of a customer is evaluated. The ontology should be updated by possibly new words, which need to be manually related as synonym terms or term abstractions.

In the following we describe details of this approach.

3.2 Unifying terminologies

There are currently two files available, which have been provided by the project partners Demag and Konecranes. Both files contain terms related to the material handling domain. There is one Excel spreadsheet file (from Demag) and the other file is a .csv file, which is exported from a Web-based terminology management system (Acrolinx). This Acrolinx database is shared between Demag and Konecranes. These terminologies contain translations into different languages. The Acrolinx database additionally may contain synonym relations between terms. Therefore, the first step would be to unify the existing terminologies.

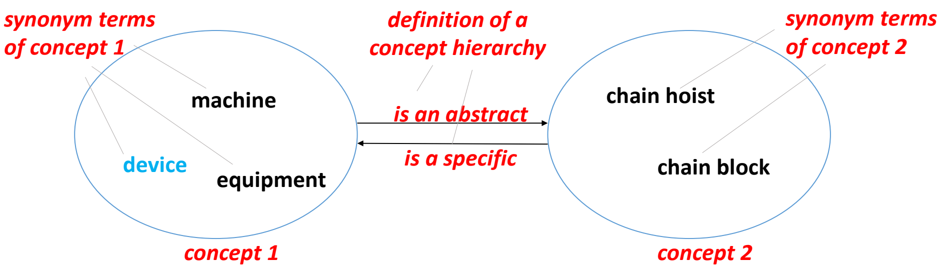

Figure 2: Terms, concepts and synonym / abstraction relations

Tables below show the desired data structure of the unified ontology. Terms and their synonyms could be stored as complex character string based tables but in terms of space efficiency, storing terms in a table consisting of two columns, term (as string) and termID (as integer) is more beneficial, since further relations can be made simply based on the integer IDs. Afterwards, terms can be connected with each other through theirs "IDs" if there is a relation between them. This can be done by using relational or NoSQL databases or simply storing them in files. The termID could be the uniqueID which is already existing in Acrolinx term database. But if this word comes from the Excel file or is a completely new word, then a new termID needs to be generated for this word. Table 1 represents an example of how terms are stored. There are two words such as "machine" and "equipment" which come from the Acrolinx term database. If they have the same conceptID (which equals to entryID in Acrolinx term database) and the same language notification, then these two words can be considered as synonyms and will be related in table 2. In other words, those terms which are synonyms belong to one concept (see also Figure 2). The third example word is "device" which has not been found in the Acrolinx term database but in the Excel file. Therefore, first a termID needs to be automatically generated for this word. Second, the word "device" could be another synonym for "equipment" but they are not related by conceptID (entryID in Acrolinx) from the beginning. In this case, the usage of some NLP techniques is inevitable to evaluate the similarity. There are some techniques for synonym extraction such as calculating a word vector and comparing vectors using cosine similarity, jaccard coefficient or other similarity measurements. Training a model could be another option if we can label pairs of words and define the relation between them manually.

Table 1 term_table

Term (str)	termID(int)	termLang
machine	1	en
equipment	2	en
device	1000	en
chain hoist	1001	en

Table 2 concept_table

termID (int)	conceptID(int)
1	1
2	1
1000	1
1001	2

Table 3 concept_concept_table

conceptID1 (int)	conceptID2 (int)	relationType
1	2	abstract-specific
…	..	…

Figure 3 shows the data model of the ontology. We have defined three entities. The term entity represents a structure of how terms are being stored in database. The concept entity shows that synonyms are assigned the same conceptID and concept_concept entity are representative of the relations between concepts. The data structure is stored in Sqlite, which functions as a relational database.

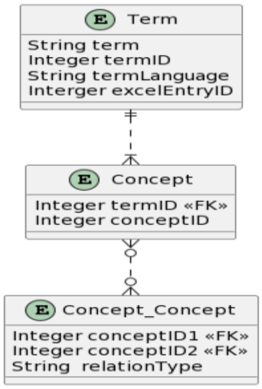

Figure 3: Ontology Data Model

Table 3 shows the relation between concepts. For example, “machine” can be an abstract term for “chain hoist” while “chain hoist” is a specific term of “machine”. For identifying element-of relation, some other ways such as word stemming or lemmatization might be helpful. For example, the word “Rahmen” in German language means “frame”. But in terminologies, this word is often used alongside other words such as “Führungsrahmen” which means “Guide frame”. Therefore, if utilizing word stemming or lemmatization, gives us the root of the word which in this case is “Rahmen”, whatever term follows the term “Rahmen”, could be an indication of a element-of relation. In our ontology editor, first we extract keywords/entities from documents. Sale engineer can choose which terms are important. If sale engineer decides to update the ontology, these selected terms will be automatically processed, new relations between new terms and old term in ontology database will be formed and saved.

In order to integrate the element-of relation, hierarchical relations between terms in a text can be investigated which are essential for organizing concepts and understanding the semantic structure. As it is shown in Figure 4, for instance, consider the term 'cover'. It serves as an abstract term representing a general concept. Within this hierarchy, we can identify specific types of covers, such as 'cover plate' and 'cover surface', which can be considered as elements or instances of the abstract term. We implemented a tree-like algorithm to detect such relation (parent-child) between terms in a given domain. This structure is stored in a table within a SQLite database containing three columns: 'conceptID1', 'conceptID2' and 'relationType'. In this context, 'conceptID1' represents the child, 'conceptID2' represents the parent, and 'relationType' denotes the nature of their connection.

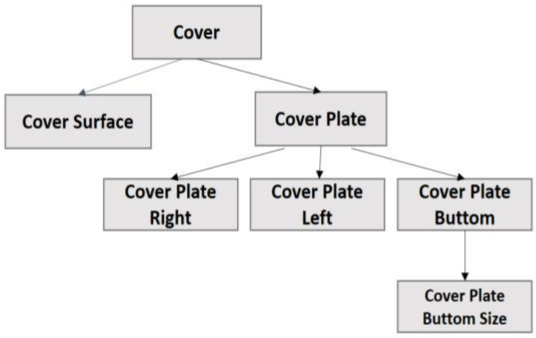

Figure 4: Hierarchical Relation between Terms

To identify abstract-specification relations, we'll utilize a large language model (LLM) known as alloma. The Ollama Python library serves as a robust resource enabling the integration of Python 3.8+ projects with Ollama. This language model is capable of generating and discussing code based on text prompts [1] .

We've also drawn inspiration from a paper on constructing ontologies using LLM [16], although our approaches are slightly different. In their paper, they randomly select various domains and inquire about related terms within each domain. If such terms exist, they further investigate hierarchical relations associated with these terms. We've already established links between keywords in documents and ontology terms in a database, comprising approximately 931 English terms across 70 project files after removing duplicates. Next, employing LLM and careful prompt engineering, we posed questions to LLM regarding abstract-specification relations between terms within the 931-term category. This process took approximately 4323.269 minutes. Below, you'll find how we formulated our prompt. Unfortunately, we are unable to measure the accuracy of this method automatically as there is no ground truth available. However, we can manually investigate if the overall results appear satisfactory.

- Let T_1represent the first term.

- Let T_2represent the second term.

- T_1 might include or be part of T_2 can be denoted as $T_1 \in T_2$, indicating that T_1 is a subset of T_2

- T_1 is not necessarily a T_2 can be denoted as $T_1 \nexists T_2$, indicating that T_1 is not a subset of T_2 .

In mathematical terms:

- For example, if T_1 = 'chain hoist' and T_2 = 'machine', we have 'chain hoist' ∈ 'machine', indicating that 'chain hoist' is-a 'machine'.

- However 'machine' ∄ 'chain hoist' , indicating that 'machine' is not necessarily a 'chain hoist'.

- If T_1 is-a T_2 and T_2 is-a T_1, indicating that T_1 *and* T_2 are synonyms.

Given this information, we need to determine whether $T1$ is-a $T2$, denoted as $T1 \rightarrow T2$ or $T1 \rightarrow \not{T}2$. Please answer with either 'yes' or 'no'. If you cannot decide whether the answer is 'yes' or 'no', it means the answer is 'no'.

---

[1] GitHub - ollama/ollama-python: Ollama Python library

Applying this to the prompt, we can represent the relationships between terms in the material handling domain using set notation and subset relationships.

3.3 Optional update of an ontology using project files

Existing project files need to be scanned for terms, which are not yet in the ontology. Those terms need to be added and possibly need to be related to existing terms. **Error! Reference source not found.** describes different steps for this task. In order to achieve this goal, existing project files need to be converted to raw text first. Afterwards, the text will be processed and some information such as stop words, unnecessary numbers, URLs, email addresses and punctuations which are not required can be removed. In normalization phase, the spelling of the words need to be checked and corrected in case of error detection.

Next step would be keyword extraction. Here, we can take two different approaches. Every email, article or comment has its own specific terms that makes them helpful or useless. Therefore, the first approach would be to extract keywords from text. The keyword extraction process detects only those terms based on different statistical factors [3] such as:

- number of times a term appears in uppercase or as an acronym
- the position of the text within a sentence
- term frequency in a document
- co-occurance of some terms together
- number of times that a term appears in different sentences

There are different algorithms available for finding keywords such as Total Keyword Frequency (TKF), Term Frequency-Inverse Document Frequency (TF-IDF), Rapid Automatic Keyword Extraction (RAKE), Yet Another Keyword Extraction (YAKE), Graph-based Keyword Extraction (GRAPH). The evaluation of results shows that between all the algorithms, YAKE has achieved quite good results as a keyword extractor considering multilingual support [17].

In second approach, if we are not interested only in keywords but all the terms, we can apply tokenization to divide text into smaller pieces called tokens. Tokens could be words, characters or subwords [18]. In any case, extracted terms from the text need to be compared and matched with the terms in the ontology. Possible synonym and abstraction relations must be found and the ontology and term-file relations need to be updated.

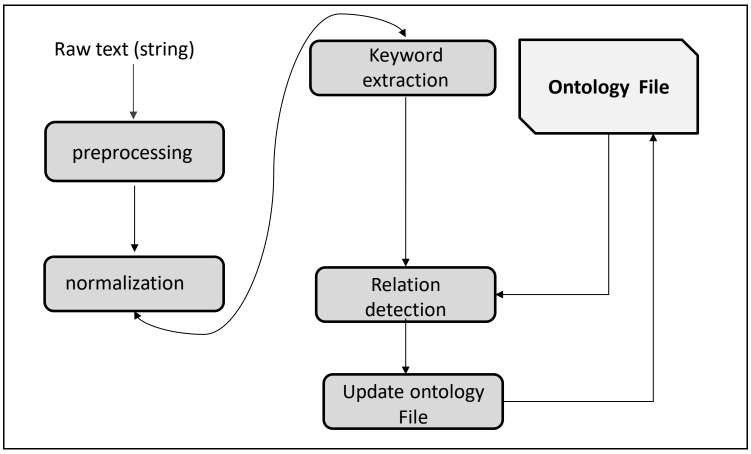

Figure 5: Evaluation of existing project files

3.4 Regular update based on incoming inquiries

The manufacturer gets inquiry emails from customers, who use possibly a different vocabulary than used by experts in previous projects. There should be an automatically triggered process to update and extend the ontology whenever an inquiry email of a customer is evaluated. The ontology and relation files should be updated accordingly by possible new words, which need to be manually related as synonym terms or term abstractions. In order to extract keywords from incoming email, we applied Named Entity Recognition (NER) on some sample data from Demag. The evaluation of results shows that NER can be utilized to do the task. Named Entity Recognition (NER) is a natural language processing (NLP) task that involves identifying and classifying named entities in text. Named entities refer to specific types of entities, such as persons, organizations, locations, dates, time expressions, quantities, monetary values, and more[19]. Figure 6 describes the process.

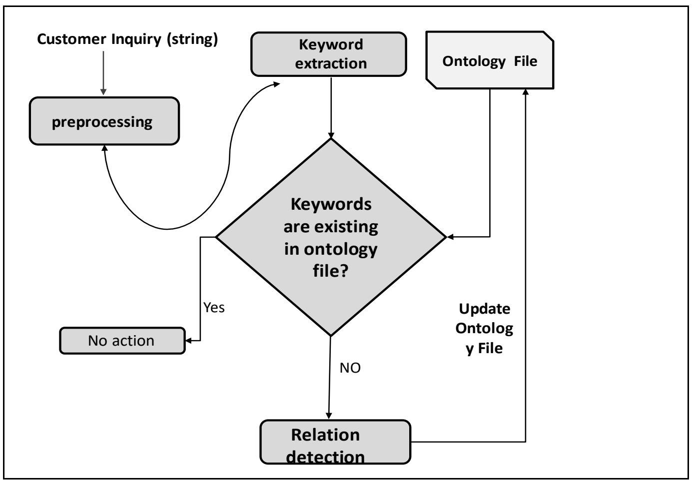

Figure 6: Evaluation of incoming customer inquiries

4 Upper/Domain Ontology approach

4.1 Overview

This section represents different steps towards creating an upper-level ontology for InnoSale. The concepts proposed are in high-level which can be specialized, refined, and instantiated in different domains. We chose Methontology as the methodology to create the ontology, following these steps:

- Specification
- Conceptualization
- Formalization
- Implementation
- Maintenance

This ontology is represented in OWL. OWL is an ontology language for the Web that provides modelling constructs to represent knowledge with a formal semantics[20].

The ontology is encoded in OWL for the following reasons [2]:

- knowledge represented in OWL can be processed by a number of inference software packages.
- support of the creation of reusable libraries.
- a variety of publicly available tools for editing and syntax checking.

In the following we describe details of this approach.

4.2 Specification

In this phase, we identified the domain and scope of the ontology and searched for existing ontologies to consider reusing.

The target domain of our InnoSale project is to innovate today's sales systems and processes for complex and variable industrial equipment, plans, and services. The characteristics of these products make this task fundamentally different from the sales of products that can be easily selected in a shop system. The products considered in InnoSale have a complexity and variance that do not allow for a complete representation in a catalogue. The target users for InnoSale are sales experts, including sales and technical back-office engineers, involved in the manufacturing and sales process. They should be able to query using their vocabulary and receive purchase recommendations that are more accurate based on their requirements.

What is the purpose of the ontology?

Innosale's solution aims to provide a mechanism for customers to express their requirements in natural language, and thus automatically suggest valid product configurations. This ontology is developed to represent concepts related to the industrial products to support the transformation queries described in the vocabulary of the customer into queries using the terms of the product manufacturer. The collection of concepts involved in the ontology will allow to understand customer preferences, requirements, and constraints in order to provide product recommendations. The ontology will act as a foundational element in the InnoSale project, supporting semantic search and serving as a knowledge base.

What is the scope of the ontology?

The ontology will allow the representation of the products related to different industrial domains (waste management, cranes, digital products, etc). The ontology covers concepts and relationships regarding the description of the products, and also have into account the user characteristics in order to offer accurated recommendations. The ontology was developed in accordance with widely recognized standards (e.g., OWL, RDF) to facilitate interoperability with other systems and ontologies. This ensures that the ontology can seamlessly integrate with existing industrial information systems, enhancing its utility and adoption across different domains.

The ontology is structured following an upper and domain ontology approach. The upper ontology serves as a foundational framework that defines the most general concepts, relationships, and properties applicable across all industrial domains. This provides a consistent and unified structure that supports interoperability and ensures that all domain ontologies adhere to the same fundamental principles.

Each industrial domain—such as waste management, crane systems, digital products, and others—can then build its own domain ontology as an extension of the upper ontology. These domain-specific ontologies will define concepts, relationships, and properties unique to their respective industries. The modularity of this approach ensures that domain ontologies can be developed and expanded independently, without altering or disrupting the core upper ontology.

This design allows the ontology to be both flexible and scalable, easily accommodating the diverse needs of various industries while maintaining consistency and coherence across the entire system. As new domains are introduced or existing ones evolve, their corresponding domain ontologies can be seamlessly integrated into the overall structure.

Additionally, the modular structure will simplify the maintenance and evolution of the ontology. As industrial domains evolve or new ones emerge, their corresponding domain ontologies can be updated or created without affecting the integrity of the upper ontology. This ensures that the ontology remains relevant and up-to-date as industry standards and technologies advance.

Who are the endo users of the ontology? (1) system integrators for industrial information systems and softwares, (2) industrial software developers.

We also checked similar ontologies in this domain that can be reused.

[21] proposes a user profile ontology and a vehicle ontology, and [22] proposes a user profile ontology. We used the following concepts:

User: This class represents any user involved in the InnoSale domains (sales engineer, customer, etc). Here we can find the user Identification (name, surname, etc), and other information of interest such as bank account number, etc.

User Profile: Each user has a profile. This information can be groped and structured in the form or a set of attributes. Each attribute is characterized by a name, and content (describes the possible values that are associated with each attribute). For example, a user can have a InterestProfile as a subclass of UserProfile, and the attributes could be interest_level, interest_type.

In [23] and [24] the authors define the following concepts:

Product: "*a product is an artifact, a substance, information, or a service. A configurable product is composed of several entities, produced by a natural or artificial process and is, or is* *intended to be, sold. A product in our definition is the final product, the product instance, the product individual or the final product that is described after the configuration process*". A product has a ProductID and a ProductName.

Physical characteristic*: "The physical characteristic (e.g., length, weight, etc..) of a product or product component"*. A characteriscic has: characteristicName, characteristicValue

Product component: "*One of the hardware, electronic or software (e.g., parts, sub-assemblies) that make up a product. A component may be subdivided into other components, which combine into sub-assemblies and assemblies to define products. Each product can be made of several components, and the same component can be used by different products"*. A product component has a PartID and a PartName

4.3 Conceptualization

In this activity, we organize and structure the knowledge from the related literature, existing ontologies and experts in the field. The primary concepts obtaided are used to develop a class hierarchy, where we define class properties associated with these concepts and determine class properties associated with these concepts.

The ontology include synonym sets and equivalent class relationships to manage ambiguity in natural language queries. For each concept, synonyms will be defined and linked through properties such as *hasSynonym*. This will allow the ontology to map different terms with the same or similar meanings to the correct concept in the knowledge base. For example:

Product: The term "device" might be a synonym for "product."

User: Terms like "client," "customer," and "buyer" can all be mapped to the User class.

These synonym sets will be utilized during the query transformation process, enabling the system to recognize and process queries regardless of the specific terms used by the customer.

The ontology addresses varying levels of product complexity by implementing a detailed product component hierarchy. This hierarchy allows the representation of products at multiple levels of granularity, from high-level assemblies to individual sub-components. For instance:

Top-Level Product: Represents the complete product or system (e.g., a crane, or an enterprise resource planning (ERP) system).

Sub-Assembly: Represents major subsystems or assemblies within the product (e.g., the lifting mechanism, or the financial management module within the ERP system.).

Sub-Component: Represents smaller parts or elements that make up a product component (e.g., motor windings within the lifting mechanism or a specific function or algorithm within the financial management module, such as the payroll calculation function.).

This hierarchical approach ensures that the ontology can accurately represent complex products, capturing the relationships and dependencies between different product components. Additionally, the ontology will support varying levels of detail depending on the needs of the user, whether they require a broad overview or an in-depth analysis of specific product parts.

In the table below, we list the concepts involved in the ontology:

Table 4 Upper Ontology Concepts

Concept	Description	Attributes
Functionality	In order to accomplish the task for which it was designed, the product performs one or more functions	description
User	This class represents any user involved in the InnoSale domains (sales engineer, customer, etc). Person or entity that has an interest in or acquires a product	e-mail, address, first_language
User Profile	Each user has a profile. This information can be groped and structured in the form or a set of attributes. Each attribute is characterized by a name, and content (describes the possible values that are associated with each attribute). For example, a user can have a InterestProfile as a subclass of UserProfile, and the attributes could be interest_level, interest_type	Attribute_name, attribute_content
Product	A product is a description of any item that can be or is offered for sale by vendors or manufacturers.	Name, identificator, description, type. producer, warranty, certification.
Physical characteristic	The physical characteristic (e.g., length, weight, etc..) of a product or product component	characteristicName, characteristicValue
Product component	One of the hardware, electronic or software (e.g., parts, sub-assemblies) that make up a product. A component may be subdivided into other components, which combine into sub-assemblies and assemblies to define products. Each product can be made of several components, and the same component can be used by different products.	PartID and a PartName
Sub-Assembly	Represents major subsystems or assemblies within a product, which are considered as a type of Product component.	SubAssemblyID, SubAssemblyName
Sub-Component	Represents smaller parts or elements that make up a product component.	SubComponentID, SubComponentName
Regular Prive Specification	Describe the price of the product	Currency, VATIncluded, maxValue, maxValue, minValue.
LegalEntity	represents a company or organization	Legal_name, primary mailling address

Information Chanel	means of information with which the customer found out about the company's product or service	Information_chanel_name, information_chanel_description
Official web	Company official website	WebsiteURL
Email	A specific "Information Channel" that uses email as a communication medium.	EmailAddress, EmailContent, EmailDate
Demographic statistic	Demographic or statistical information associated with a "User" (e.g., age, gender, etc.).	StatisticType, StatisticValue
Inquiry	A request for information or a query made by a "User" through an "Information Channel.	InquiryID, InquiryDate, InquiryContent
Quotation	A formal offer specifying the price and terms of sale for a specific product.	QuotationID, QuotationAmount, QuotationDate
Configuration	The way a "Product" can be configured according to specific customer needs or technical requirements.	ConfigurationID, ConfigurationOptions, ConfigurationDate
Customer	A specific type of "User" involved in the process of purchasing or requesting products.	

Table 5 A outline and description of the object properties

Object property	Description
hasUserProfile	Every user has one user profile
makesInquiry	A user makes an inquiry.
inquiryThough	An inquiry is made through an information channel
hasComponent	A product has one or more product components, , which may include sub-assemblies, and sub-components.
hasPhysicalCharacteristic	A product or product component has physical characteristics like length, weight, etc
isQuotedIn	A product or product component can be included in one or more quotations.
hasRegularPriceSpecification	A product has a regular price specification.
hasDemographicStatistic	A user can have demographic or statistical information such as age, gender, etc.
usesInformationChannel	A user uses an information channel to interact with or inquire about a product.
hasFunctionality	A product or product component is associated with one or more functionalities to accomplish tasks.
generatesQuotation	An inquiry generates a quotation.
includesProduct	A quotation includes one or more products.

4.4 Formalization and Implementation

We used Protégé to convert our formal model into an OWL-DL. InnoSale Upper Ontology consists of 18 classes. Figure 7 shows the hierarchy.

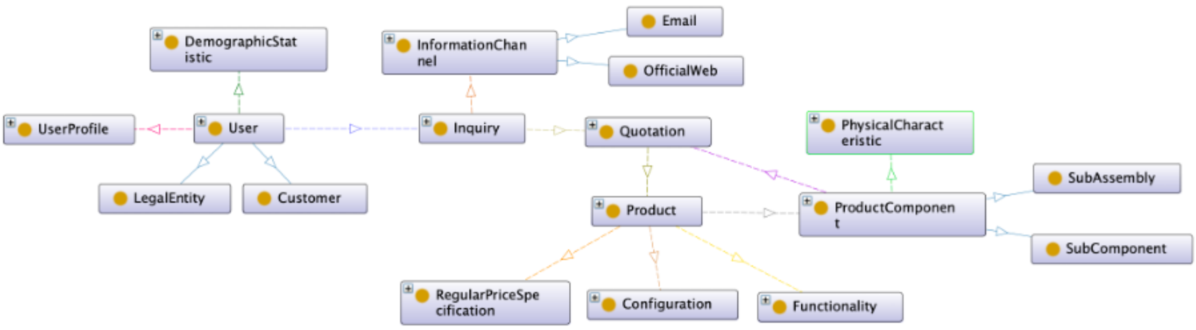

Figure 7: InnoSale Upper Ontology

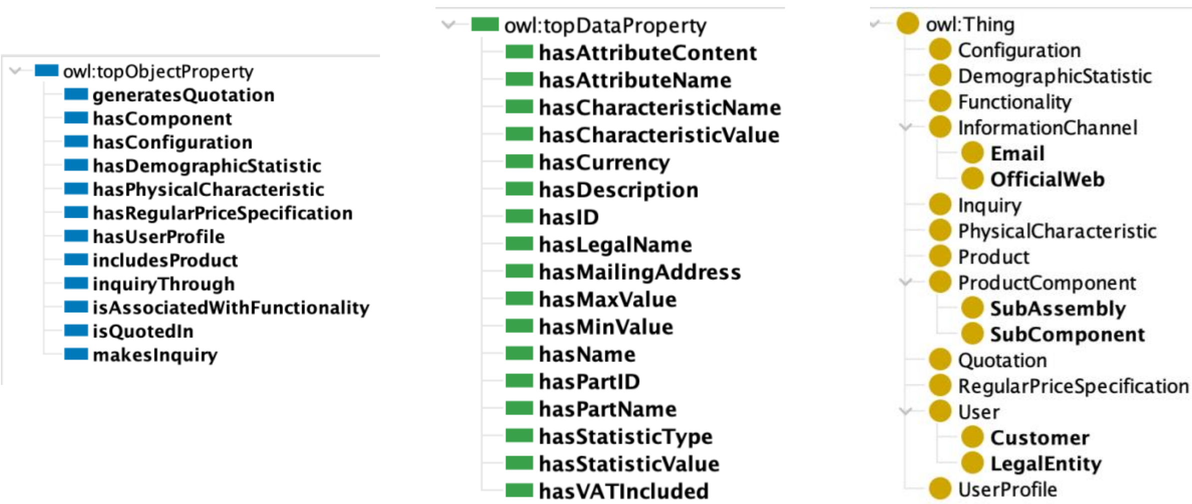

Figure 8 An overview of the InnoSale Ontology classes, object properties and data properties

4.5 Maintenance

This phase is important for ensuring the longevity, accuracu and relevance of the InnoSale Ontology over time. Some key activities in the maintenance phase are:

- Continuously monitor the domain for new concepts, relationships and attributes and update the ontology to reflect any changes or expansions in the domain knowledge.
- Keep detailed documentation of all changes made to the ontology during maintenance.
- Identify and correct errors or inconsistencies by regularly validating the ontology against feedback from domain experts to ensure its accuracy and correctness.

5 Conclusion

This deliverable has provided a comprehensive exploration into the development of the Knowledge Model, with a focus on enhancing semantic search capabilities within InnoSale project. Chapter 2 provides a foundational understanding of ontologies and their role in semantic search.

Chapter 3 describes a data driven approach to create an ontology, which is intended to be used in semantic search of previous offers or projects in relation to an incoming inquiry text. The approach is not full-automatic, since this would lead to a poor quality of term definitions and term relations. Thus an Ontology Editor will provide a semi-automatic ontology creation process, which supports the sales engineers by suggestions of terms and relations as precise as possible. A bigger initial effort for maintaining the ontology is expected than in the long run of the system.

Chapter 4 describes the adoption of the Upper/Domain Ontology approach, bu establishing a clear opper ontology as the foundation, the ontology can accommodate domain-specific stentios, allowing for the seamless integration of additional industrial domains without compromising the overall structure.

By providing a structured and scalable knowledge base, the ontology enhances the project's ability to deliver accurate product configurations and recommendations based on user requirements. Moving forward, the ontology will serve as a critical component in the InnoSale project, driving innovation in sales processes for complex and variable industrial equipment.

6 Abbreviations

JSON	JavaScript Object Notation
NLP	Natural Language Processing
TKF	Total Keyword Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
RAKE	Rapid Automatic Keyword Extraction
YAKE	Yet Another Keyword Extraction
GRAPH	Graph-based Keyword Extraction
NER	Named Entity Recognition

7 References

[1] F. Baader, I. Horrocks, and U. Sattler, "Description Logics as Ontology Languages for the Semantic Web," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2605 LNAI, pp. 228–248, 2005, doi: 10.1007/978-3-540-32254-2_14.

[2] R. Batres *et al.*, "An upper ontology based on ISO 15926," *Comput Chem Eng*, vol. 31, no. 5, pp. 519–534, 2007, doi: https://doi.org/10.1016/j.compchemeng.2006.07.004.

[3] S. Malik and S. Jain, "Sup_Ont: An Upper Ontology," *https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJWLTT.20210501.oa6*, vol. 16, no. 3, pp. 79–99, Jan. 1AD, doi: 10.4018/IJWLTT.20210501.OA6.

[4] P. Cimiano, A. Mädche, S. Staab, and J. Völker, "Ontology Learning," *Handbook on Ontologies*, pp. 245–267, 2009, doi: 10.1007/978-3-540-92673-3_11.

[5] C. Mangold, "A survey and classification of semantic search approaches," *Int. J. Metadata, Semantics and Ontology*, vol. 2, no. 1, pp. 23–34, 2007.

[6] L. Ding *et al.*, "Swoogle: a search and metadata engine for the semantic web," *International Conference on Information and Knowledge Management*, pp. 652–659, Nov. 2004, doi: 10.1145/1031171.1031289.

[7] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "{BibSonomy}: A Social Bookmark and Publication Sharing System," *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pp. 87–102, 2006, Accessed: Aug. 08, 2024. [Online]. Available: http://www.kde.cs.uni-kassel.de/hotho/pub/2006/iccs_tools_ws_final.pdf

[8] O. Corcho, "Ontology based document annotation: trends and open research problems," *Int J Metadata Semant Ontol*, vol. 1, no. 1, pp. 47–57, Jan. 2006, doi: 10.1504/IJMSO.2006.008769.

[9] M. Tkachenko and A. Simanovsky, "Named entity recognition: Exploring features," in *Proceedings of KONVENS 2012*, J. Jancsary, Ed., ÖGAI, Sep. 2012, pp. 118–127. [Online]. Available: http://www.oegai.at/konvens2012/proceedings/17_tkachenko12o/

[10] V. G and D. Chandrasekaran, "Sentiment Analysis and Opinion Mining: A Survey," *Int J Adv Res Comput Sci Technol*, vol. 2, Aug. 2012.

[11] S. I. Nikolenko, S. Koltcov, and O. Koltsova, "Topic modelling for qualitative studies," *J Inf Sci*, vol. 43, no. 1, pp. 88–102, Feb. 2017, doi: 10.1177/0165551515617393.

[12] M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Syst Appl*, vol. 106, pp. 36–54, Sep. 2018, doi: 10.1016/J.ESWA.2018.03.058.

[13] T. Al-Moslmi, M. Gallofre Ocana, A. L. Opdahl, and C. Veres, "Named Entity Extraction for Knowledge Graphs: A Literature Overview," *IEEE Access*, vol. 8, pp. 32862–32881, 2020, doi: 10.1109/ACCESS.2020.2973928.

[14] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, "Deep Semantic Role Labeling: What Works and What's Next," *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 473–483, 2017, doi: 10.18653/V1/P17-1044.

[15] V. Ng and C. Cardie, "Improving Machine Learning Approaches to Coreference Resolution," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, vol. 2002-July, pp. 104–111, 2002, doi: 10.3115/1073083.1073102.

[16] M. Funk, S. Hosemann, J. C. Jung, and C. Lutz, “Towards Ontology Construction with Language Models,” *CEUR Workshop Proc*, vol. 3577, 2023.

[17] J. Piskorski, N. Stefanovitch, G. Jacquet, and A. Podavini, “Exploring Linguistically-Lightweight Keyword Extraction Techniques for Indexing News Articles in a Multilingual Set-up,” in *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, H. Toivonen and M. Boggia, Eds., Online: Association for Computational Linguistics, Apr. 2021, pp. 35–44. [Online]. Available: https://aclanthology.org/2021.hackashop-1.6

[18] M. Straka, J. Hajič, and J. Straková, “UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds., Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 4290–4297. [Online]. Available: https://aclanthology.org/L16-1680

[19] D. Jurafsky and J. H. Martin, “Named Entity Recognition and Classification,” in *Speech and Language Processing* , 3rd ed., 2020, ch. Chapter 23.

[20] L. W. Lacy, *Owl: Representing Information Using the Web Ontology Language*. Trafford, 2005. [Online]. Available: https://books.google.com.co/books?id=71sGYlby2uwC

[21] M.-H. and G. P. Le Ngoc Luyen and Abel, “Towards an Ontology-Based Recommender System for the Vehicle Sales Area,” in *Progresses in Artificial Intelligence & Robotics: Algorithms & Applications*, A. and K. N. and D. R. I. and B. I. Troiano Luigi and Vaccaro, Ed., Cham: Springer International Publishing, 2022, pp. 126–136.

[22] Y. Elallioui and O. El Beqqali, “User profile Ontology for the Personalization approach,” *Int J Comput Appl*, vol. 41, pp. 31–40, Mar. 2012, doi: 10.5120/5531-7577.

[23] G. Oddsson and K. R. Ladeby, “From a literature review of product configuration definitions to a reference framework,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 28, no. 4, pp. 413–428, 2014, doi: DOI: 10.1017/S0890060413000620.

[24] G. Bruno, D. Antonelli, and A. Villa, “A Reference Ontology to Support Product Lifecycle Management,” *Procedia CIRP*, vol. 33, pp. 41–46, Jan. 2015, doi: 10.1016/J.PROCIR.2015.06.009.