Investigating Contextual Ontologies and Document Corpus Characteristics for Information Access in Engineering Settings

Authors: Mahsa Mehrpoor[1], Dirk Ahlers[2], Jon Atle Gulla[2], Kjetil Kristensen[1], Ole Ivar Sivertsen[1]

[1] Department of Engineering Design and Materials, Faculty of Engineering Science and Technology, NTNU, Trondheim, Norway

[2] Department of Computer Science, Faculty of Information Technology and Electrical Engineering, NTNU, Trondheim, Norway

mahsa.mehrpoor@ntnu.no, dirk.ahlers@ntnu.no, jon.atle.gulla@ntnu.no, kjetil.kristensen@ntnu.no, ole.ivar.sivertsen@ntnu.no

Abstract

Knowledge and information resources play a pivotal role in enterprises and are valuable for solution reuse and learning through information access. However, identifying relevant information from a rapidly growing number of unstructured resources is challenging for users. We discuss a personalized information access tool for professional workplaces based on the recommender systems to provide relevant documents for users in specific work contexts based on domain-specific ontologies. Our use case is a multidisciplinary engineering project building an energy-efficient vehicle. We provide an in-depth analysis of document corpus characteristics of this real-life shared engineering workspace to understand the content and context of documents using information retrieval methods and semantic annotations. Upon this, we build a contextual ontology as our knowledge domain for the recommender system. We validate our ontology-based content matching approach by evaluating the level of retrievability and coverage of the ontology against the indexed document corpus through experiments on the corpus and ontology. Our results provide insight into engineers' document workspaces and show that even a simple domain ontology is able to match a majority of documents from a domain-oriented corpus. The findings support our approach of using ontology-based recommendation for domain-specific workspaces.

Keywords: Document corpus analysis, Information Access, Engineering settings, Work context, Ontology, Relevance, Recommender Systems

Introduction and Motivation

Knowledge and information resources have a highly important place in enterprises. In a study of 1998-2005, 70% of all US jobs could be classified primarily as "tacit jobs" - typical knowledge intensive jobs, drawing on deep experience and "tacit knowledge", making complex decisions based on knowledge, judgment, experience, and instinct (Johnson, Manyika, & Yee, 2005). Similar patterns

This is a preprint of an article whose final and definitive form has been published in the Journal of Information Technology Case and Application Research, Volume 19, Issue 1, 2017, Routledge, Taylor & Francis.
http://www.tandfonline.com/doi/full/10.1080/15228053.2017.1313557 http://dx.doi.org/10.1080/15228053.2017.1313557

can be observed in other economies and knowledge workers involved in tacit interactions represent the quickest growing segment of workers. As knowledge workers, people in an enterprise – engineers in particular – have different levels of expertise and require specific knowledge and information embedded in different types of knowledge objects stored in internal or external resources. Retrieving effective and efficient "tacit knowledge" remains challenging. Better identification, transfer and management of knowledge helps organizations to retain their resources of knowledge and reuse them in other projects instead of having to recreate it (Owen, Burstein, & Mitchell, 2004).

A survey performed in (Williams, Figueiredo, & Trevelyan, 2013) classifies different types of interactions that engineers have for information access into three groups of face-to-face, reading documents, and interactions with abstract systems and data. Abstract systems comprise conceptual and non-physical systems such as software-based systems. Interaction with systems comprises searching for information in file systems, databases, the Web, and other resources for design, modeling, simulation, and programming. File systems continue to be one of the common systems used in engineering projects for managing data. Many companies try to follow a standard naming convention for a document names and paths to improve accessibility (Eck & Schaefer, 2011). Yet directory hierarchies have challenges such as re-finding or quick browsing of filed away information that could be easily forgotten, since it is out of sight and the folder hierarchy can be rather large and complex (Jones, Phuwanartnurak, Gill, & Bruce, 2005; Ahlers & Mehrpoor, 2015). Other approaches to help groups of users access unstructured knowledge can be bookmarking or tagging systems to support knowledge sharing within organizations (Parise, Guinan, Iyer, Cuomo, & Donaldson, 2009).

Further, in many engineering projects, there is little consistent standard for creation of documents, naming and locating them. Engineers spend a lot of time on browsing and searching directories. Furthermore, the most obvious solution, such as personalized desktop search tools or existing search built into the operating system as tools to search users' own computer files, may not be sufficient for engineers' expectations and needs. A solution would have to enable for example refinement along workflow, classifications, topics, and other domain-specific features (Ahlers & Mehrpoor, 2015). These shortcomings make it difficult to assess which documents are available in the first place, where they are located, and how relevant they are to a specific task. The problem keeps increasing in complexity and scope with the constant growth of archived documents.

To enhance engineers' productivity, available data sources should be efficiently re-useable and re-findable without expensive user annotations to avoid wasting time on searching knowledge that already exists within the organization, contributing to a lean enterprise (Kristensen, Krogstie, Ahlers, & Mehrpoor 2016). We aim for a system that identifies users' information needs and relevant documents fitting their needs. Such systems are known as search engines and in particular as recommender systems that are useful tools for interacting with large and complex information spaces, as e.g. used in e-commerce (Ricci, Rokach, Shapira, & Kantor, 2011. We adapt existing recommender system approaches for dealing with the information access challenge to filter and prepare information according to engineers' project work context (Mehrpoor, Gjarde, & Sivertsen, 2014).

The purpose of this research is improving access to relevant existing resources in an enterprise, for engineers in multi-disciplinary engineering projects. In our prior research, we discussed challenges for information access in collaborative engineering workplace settings (Ahlers, Mehrpoor, Kristensen, & Krogstie, 2015), dealing with heterogeneous documents stored in shared networked file systems (Ahlers & Mehrpoor, 2015), and the framework of our proposed system in an engineering use case (Mehrpoor et al., 2015).

In this paper, our contributions are twofold. First, we present a detailed corpus analysis of a real-life shared workspace file system from an engineering setting to better understand the available documents from an indexing and retrievability perspective and understand the characteristics and distribution of files; and second, to evaluate the document corpus against a developed domain ontology for document coverage and retrievability to validate the degree to which an ontology-based content matching approach can annotate this corpus and make the documents accessible for improved information access.

Literature review

This section discusses fundamental issues in the proposed information access system. It includes concepts such as search engines, file systems, information retrieval, recommender systems, ontologies as knowledge management tools in enterprises, and content extraction and analysis.

Information retrieval and search engines in file systems

People use different types of information retrieval systems in their daily life to satisfy their information needs such as web search engines, for example Google or Bing. Search engines are tools that index and search documents for specific keywords and return a list of documents that match query keywords. This is an application of information retrieval technology. "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" (Manning, Raghavan, & Schütze, 2008). Apart from well-known public search engines, there are other more specialized search tools. One example designed to work offline is Desktop Search for searching internal documents stored in a local computer. File systems of local computers or shared servers contain loads of data and information embedded in different files and directories. In order to search and access such files, prototypical search engines have been developed for different use cases. Eureka (Bhagwat & Polyzotis, 2005) is a file system search engine that infers the relations among files for improving the ranking of search results. For inferring semantic links between files, it defines three types of semantic links, content overlap, name overlap, and name reference link. They are automatically identified based on content and metadata of files. Another file system search tool called Connection (Soules & Ganger, 2005) combines traditional content-based search with context information that is collected from user activity. It identifies relations among files by tracing file system calls and uses them for reordering traditional content-based results. As discussed in the introduction, such functionalities of file system search are not powerful enough for relevant information retrieval and especially recommendation in our setting. A next step in information access leads to using additional semantic solutions and hierarchical naming towards attribute-based naming in early work in semantic file systems (Gifford, Jouvelot, Sheldon, & James W. O'Toole, 1991). Thus, analysis and retrieval of engineering documents stored in file system could be augmented through utilizing semantic-based systems such as semantic search engines to provide more relevant results (Esa, Taib, & Thi, 2010; Jayavel, Anouncia, & Kapoor, 2013). This also feeds into the topic of enterprise search. Regarding measurement and evaluation, the effect of a retrieval system on users' ability for information access is investigated and a method to capture the level of retrievability of documents is proposed (Azzopardi & Vinay, 2008).

Yet existing systems either are too general or do not apply domain-specific tools and knowledge to improve information retrieval against stored data and information in file systems with the engineering focus and cannot satisfy engineering needs, which is why we propose a new system matching these requirements. We follow the approach of domain-specific understanding of the documents to allow suitable filtering. In short, we use some existing software, but combine, adapt, and extend it in a novel way to be able to add semantic capabilities through the domain-specific ontology into a search system that we adapt to recommend documents based on semantic user context.

Recommender systems

Recommender systems are tools that suggest relevant items to users according to their interest identification (R. Burke, Felfernig, & Göker, 2011; Mahmood & Ricci, 2009). Recommender systems follow different approaches to match knowledge sources of a given domain. Knowledge sources are classified in three groups of social, individual and content; depending on what sources are available in the domain, the appropriate recommendation approach is selected (R. D. Burke & Ramezani, 2011). The two major approaches are content-based filtering and collaborative filtering. In content-based filtering, an item is recommended based on the similarities between item specifications and the profile of user's interests (Pazzani & Billsus, 2007). When the number of users is large, collaborative filtering can utilize the similarities between the preferences of a given user with other users. Items highly rated by similar users are then ranked higher in the recommendation (Schafer, Frankowski, Herlocker, & Sen, 2007). Context-aware recommendation is another approach that considers contextual information such as time and location along with the two basic entities of users and items in order to provide more personalized recommendations for target users (Adomavicius & Tuzhilin, 2011). Any effective contextual variable of the situation of the user is considered to individualize recommended items that fit the given user in certain circumstances (Verbert et al., 2012). Each of the recommendation approaches has its own strengths and weaknesses. For augmenting their performance, hybrid systems can combine recommendation approaches so that they complement the performance of each other and enable better recommendations.

Ontologies as knowledge management tools in enterprises

An ontology can be defined as a formal description of concepts and their relationships (Staab & Studer, 2013). The role of ontologies is widely examined in knowledge sharing and reuse and they have been accepted as an important tool for managing and integrating knowledge in enterprises. Ontologies have been used for knowledge integration of processes and to support complex workflow systems (Huang & Diao, 2008). In product life-cycle management, PLM, data integrity can be realized through defining a modular extendable reference ontology and integrating data along the whole product life cycle to allow semantic search and knowledge reuse (Bruno, Antonelli, Korf, Lentes, & Zimmermann, 2014). In another studied industrial case, an ontology for knowledge management is developed to support designers in generating design concepts. The developed ontology promotes knowledge reuse and systematic capture of design knowledge and helps the integration of the heterogeneous data sources (Chang, Sahin, & Terpenny, 2008). The use of ontologies for knowledge management in engineering industry is increasing as evidenced by (El Kadiri & Kiritsis, 2015; Rao, Mansingh, & Osei-Bryson, 2012; Zhen, Wang, & Li, 2013).

Furthermore, ontologies have been used in many recommender systems to improve the shortcomings that some recommendation approaches may have in different domains. A domain ontology based on users' interest in a heterogeneous environment for personalizing recommendation can minimize repetitive and tedious retrieved information (Ge, Chen, Peng, & Li, 2012). The recommendation mechanism proposed in (Zhen, Huang, & Jiang, 2010) formalizes an ontology-based context model from both the user and knowledge side and performs a semantic matching between for a more proactive way of recommendation.

Content-based filtering, content analysis and semantic annotation

Recommender systems, as well as search engines, need to build internal representations of documents. There are different item representation techniques from traditional text representation to more advanced techniques such as integrating ontologies for exploring features of the objects to be recommended and allowing to take domain-specific document features into account (Ge et al., 2012; Kang & Choi, 2011). A general concept is that only document features that are extracted and indexed can be used for search, filtering, and recommendation. This poses specific challenges to highly domain-specific systems, but also makes them more powerful than for example general search engines such as Google or Bing or general Desktop Search Tools. A high level architecture described in (Lops, De Gemmis, & Semeraro, 2011), defines three main components for a content-based recommendation process: content analyzer, profile learner and filtering component. The content analyzer component performs information and feature extraction from unstructured or semi structured documents that are machine-readable. Structured information becomes input for the profile learner and filtering components. Our focus in this paper is mainly on the content analyzer stage to study the document corpus and explore structured information and extractable features for our specific domain.

To enable information retrieval, documents need to become searchable and semantically annotated. Key information and features of documents are identified, and meta-data is extracted for documents to assist document retrieval. Meta-data represents a set of properties of the documents (Horn, 2016) and can be understood as semantic annotations for each document. An efficient retrieval process needs a central database (an index) for content and meta-data storing (Eck & Schaefer, 2011).

Case study: A multi-disciplinary engineering context

We choose the engineering scenario of interdisciplinary student groups working at our university towards building an ultra-energy-efficient vehicle. The scenario mirrors many aspects of real-world scenarios in companies (heterogeneous data, unknown file structures, large amounts of data, personnel turnover, aim to reuse knowledge, need for learning, multidisciplinary teams) and has the added benefit that we have easy access to the actual users. The Shell Eco Marathon[1] competition (SEM) is held yearly and encourages student teams around the world to design and build an ultra-energy-efficient vehicle. Since 2007, a new student team of NTNU has participated in SEM each year and worked on a vehicle called DNV GL fuel fighter[2] (Buodd & Halsøy, 2015). The main engineering disciplines involved are mechanics, electronics, and cybernetics with others involved such as design planning, aerodynamics, materials, and safety. During the project, students from different technical backgrounds work together with different levels of expertise. Since the project is repeated annually, the experiences and lessons learned of past teams are important for the current team. Building on the documented experience of past teams, every team can formulate innovative plans and consequently deliver improved results along with saving significant time, while avoiding starting from scratch.

Common ways of handover communication are sending emails or organizing meetings to transfer knowledge to the new team which is time-consuming and not efficient and sustainable. Document sources contain loads of unstructured documents of different file types that do not follow standard convention for naming and distributing in the file system. Therefore, the team still has the challenge of exploring relevant knowledge related to their assigned tasks from early to late project phases. As discussed above, ordinary desktop search tools have shortcomings when searching through a huge amount of domain-specific data. For example, documents in the Windows file system search are indexed based on generic metadata from the file system. Specialized file formats may still contain

[1] www.shell.com/energy-and-innovation/shell-ecomarathon.html
[2] www.fuel-fighter.com/

valuable content that cannot be recognized, indexed, and therefore not searched. Moreover, some documents might be only related to a particular field such as electronics or might be related to both fields of electronics and cybernetics. In such a situation, while a document is located in a folder under the root directory of electronics, it is hard to be found by a cybernetics engineer since he might not know how electronic documents are organized or not all the key terms of electronics are familiar to him to search for the required ones. These valuable and potentially reusable resources should be organized in a way to be more accessible for the team.

Approach

To be able to tailor the development of the engineering-domain recommender system, we start by examining the available document corpus to understand content and context. In this case it is a typical shared file system with content from the engineering domain as discussed above in the case description. Existing resources and their storage are studied in detail. Both qualitative and quantitative analyses are provided. Users' requirements and their expectations are studied during the process of building the specific knowledge domain in the form of an ontology. Furthermore, we perform a user study in an early part of the development process to elicit requirements and context; this is accomplished through semi-structured interviews with a user group from our scenario comprising 15 students from different engineering backgrounds that might have multiple roles and responsibilities due to the limited number of team members. After designing the conceptual knowledge model, the ontology building process is continued to formalize the domain knowledge and finally development phases following an evolutionary method.

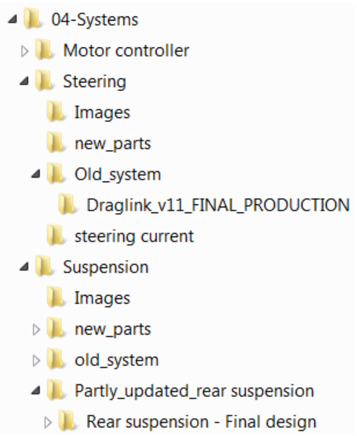
According to the characteristics of information resources, appropriate techniques for extracting information from unstructured documents are identified. Documents are semantically annotated and indexed based on a list of existing metadata from file system and additional metadata extracted from content of documents. Thereafter, the built knowledge domain is examined against indexed documents to evaluate the retrievability of documents through ontology concepts. The goal is to evaluate to what extent the ontology concepts cover existing knowledge and information resources. These individual contributions are described and evaluated in detail in the following sections.

System corpus analysis

Any Information Retrieval or Recommendation system works by matching features of a user query or context to available features within or about the items to be found. Therefore, for specialized systems, the starting point is an in-depth analysis on the available document corpus, to understand the content and context. In this section, we elaborate the results of our analysis of the file storage.

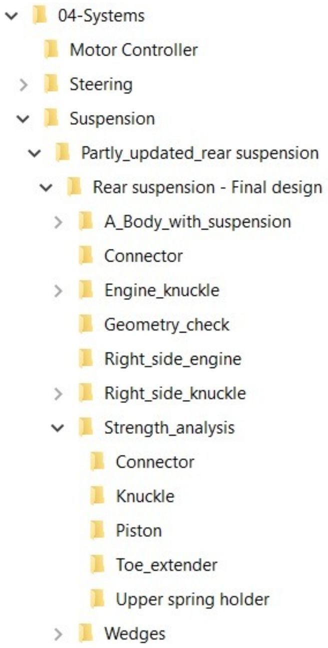
Knowledge and information storage

The data storage used in our case study is a shared networked file system on a university server containing files from the past three years of the competition. A sample of the file system structure for one year is shown in Figure 1, which shows top levels of a hierarchical directory structure. The top levels of the directory structure are similar in the years studied. However, there is not much consistency in organizing deeper levels as demonstrated in Figure 2 in terms of common names per discipline and common methods of arranging them.



```
04-Systems
  Motor controller
  Steering
    Images
    new_parts
    Old_system
      Draglink_v11_FINAL_PRODUCTION
    steering current
  Suspension
    Images
    new_parts
    old_system
    Partly_updated_rear suspension
      Rear suspension - Final design
```

Figure 1. A sample of the file system of the DNV GL project



```
04-Systems
  Motor Controller
  Steering
  Suspension
    Partly_updated_rear suspension
      Rear suspension - Final design
        A_Body_with_suspension
        Connector
        Engine_knuckle
        Geometry_check
        Right_side_engine
        Right_side_knuckle
        Strength_analysis
          Connector
          Knuckle
          Piston
          Toe_extender
          Upper spring holder
        Wedges
```

Figure 2. The same part of the hierarchical directories shown in Figure 1, demonstrated in deeper levels

Knowledge and information resources

Many different document types from the engineering domain are found in the collection. They included domain-specific formats such as computer aided design (CAD), finite element analysis modeling, programming documents etc. as well as common document types such as multimedia and text-based documents, such as office documents, PDFs, or various graphics formats.

To gain a quantitative understanding, a detailed overview of different types of documents is required in terms of their amount, size, format, associated metadata and other important aspects. We first examine metadata features such as document formats and sizes. Then, we examine the textual content of the documents in more detail to understand availability and broadness of keywords and the presence of technical terms.

The initial document type analysis is depicted in Figure 3 for multiple years. Documents are classified initially in two rough groups of textual and non-textual. Textual documents contain Microsoft Office documents, PDFs, texts, code, and other text-based document formats, while non-textual documents contain multimedia documents such as images or photos, as well as CAD models, analysis sheets, and other document types derived from specified engineering applications with no substantially or easily accessible textual content. As shown in Figure 4, a significant number of the non-textual documents are images of different formats.

In terms of naming convention, similar to the problem with directory name, there is not any standard for naming documents of different types. Some of the documents have information-rich filenames and give an idea about the content of the file but there are many documents that are named with seemingly arbitrary alphanumerical combinations or are otherwise not informative enough to describe the content. This impedes the search system aiming at keyword extraction.

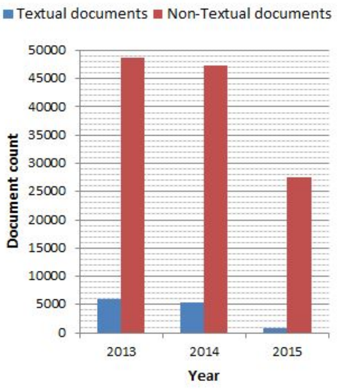


Figure 3. Number of documents in two groups of textual and non-textual in recent years.

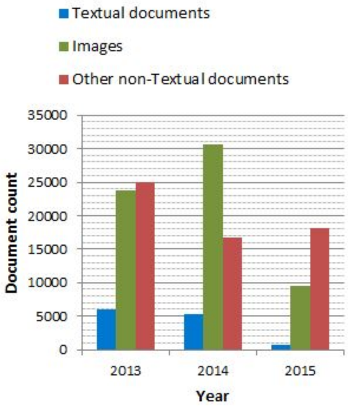


Figure 4. A detailed view of Fig3; non-textual documents and substantial amount of images.

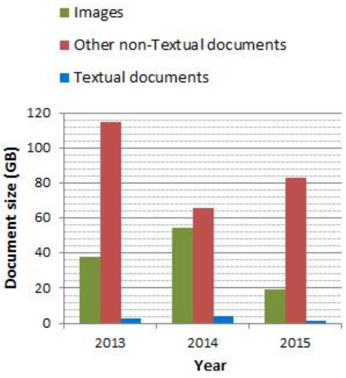


Figure 5. Size of documents from different groups represented in Fig4.

In addition, the documents from both groups are semi- or unstructured which means that documents are not based on a fixed organized model or template. Potentially valuable information is stored in archived documents but there is not any consistency in information structure in the majority of the documents. Thus, classical information retrieval systems extract keywords, but not necessarily strong semantic relations between them. As mentioned earlier, files are linked to associated metadata of the file system that can be used to explore some documents' features. However, there is a lack of metadata about content and semantic information to support browsing documents, which needs to be derived by the search system. In other scenarios where documents are stored in a document management system (DMS), this may change as metadata entry is mandated for users, but many work groups are using the shared file system for ease-of-use. In the following, both groups of documents are analyzed in detail.

Textual documents in the corpus

As shown in Figure 6, archived textual documents of the past three years are analyzed. The result shows that most of the archived knowledge is stored as PDF documents, with Word documents and

Excel files following. We also see a decline in overall numbers. This may be due to more information from previous years available and useful, and also possible due to a stronger use of online collaboration tools with their own storage. This needs to be explored in future work and possibly included in our system.

Next, we examine the actual textual content of these documents. To access the textual content, we use application-specific adapters such as Apache Tika[1] for text extraction. In addition, the information retrieval tool, Elasticsearch[2] is applied for document indexing and statistical analysis. The bar charts in Figure 7, 8 and 9 illustrate the numbers of terms (individual words) for each format in recent years.

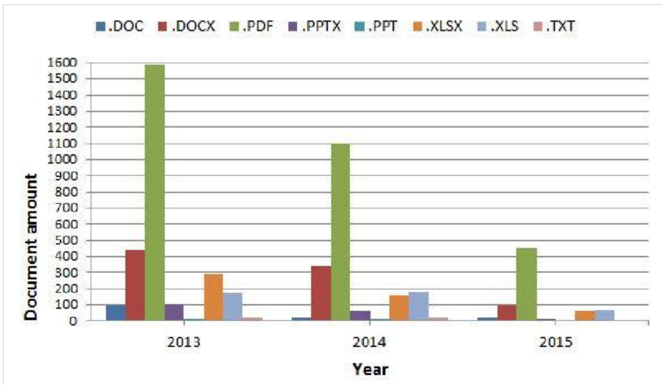


Figure 6. Existing document formats in textual documents group

The results are classified in 5 ranges to study the amount of stored content to understand how much retrievable content exists to later make it searchable and accessible for users. On the low end, a large number of PDFs and Word documents contain no textual content. We noticed that they are actually scanned PDFs or only images copied into documents with no textual content. These documents are very hard to access through textual retrieval and can only be searched by metadata. The range of terms between 0 and 1000 represents documents that contain up to two pages of textual content. The results show that the majority of archived documents are in this range. Documents with more than 1000 terms are mainly PDF and Word documents.

[1]. www.tika.apache.org
[2]. www.elastic.co/products/elasticsearch

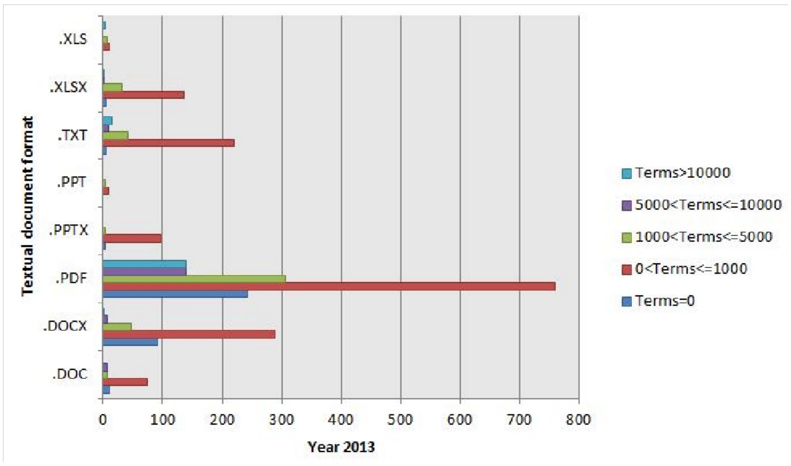


Figure 7. Number of terms per document format in year 2013

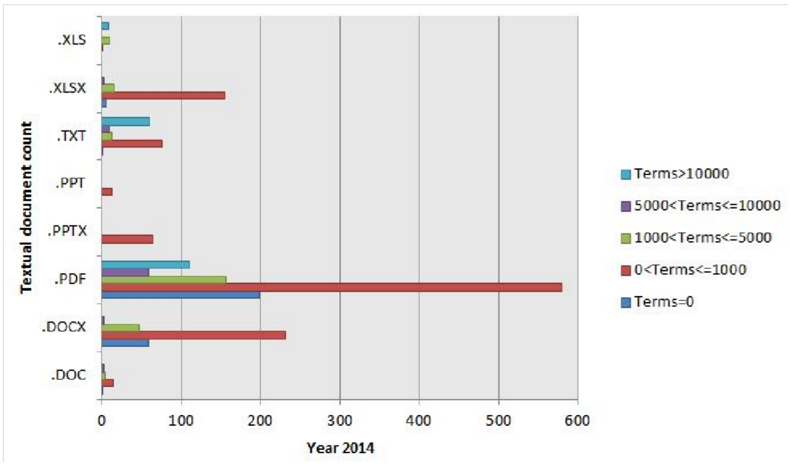


Figure 8. Number of terms per document format in year 2014

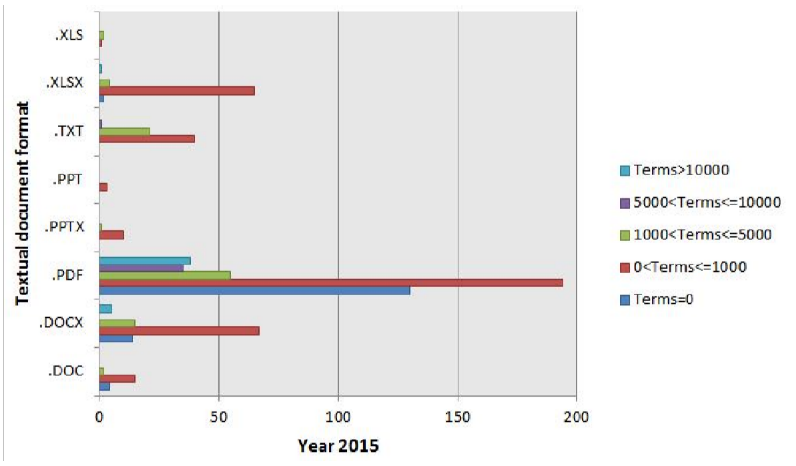


Figure 9. Number of terms per document format in year 2015

Overall, among these documents, the number of documents with no textual content is considerable. Although scanned PDFs might contain valuable information to reuse, the embedded input is hard to extract and there is same challenge for word documents and any other format with no textual content. On the other side, through file system investigation, it is found that embedded images in scanned

PDFs are often duplicated individually and stored in other locations which could make them accessible. Also, if these documents are named properly to define their content, they can be explored by taking the file name and path name into account as a feature in the system (Ahlers & Mehrpoor, 2015). Overall, we see that other documents contain a suitable amount of content to make them searchable.

Non-Textual documents in the corpus

We repeat the analysis of the textual documents for the non-textual documents. As shown in Figure 4, images and photos make up the bulk of non-textual documents. Other non-textual document formats are quite varied; average of about 250 varied formats in each year. Figure 10 shows a list of the most frequent formats in this group. The number of .PRT files is significantly higher than other formats. PRT is a common file format in CAD applications for designing 3D components such as NX, SolidWorks and Abaqus. Other formats are also strongly related to specific disciplines such as .C files that refer to programming and cybernetics tasks (which would formally count as textual documents). Others, such as .O or .LOG are output files of processes that will not be useful for recommendation and should therefore be filtered out in the system (Ahlers & Mehrpoor, 2015). More generic formats such as .PRT, .SLDPRT (Solidwork part) and .FEM (Finite element analysis) refer to design and development of mechanical tasks for users from mechanics, body, aerodynamics, or fuel cell disciplines.

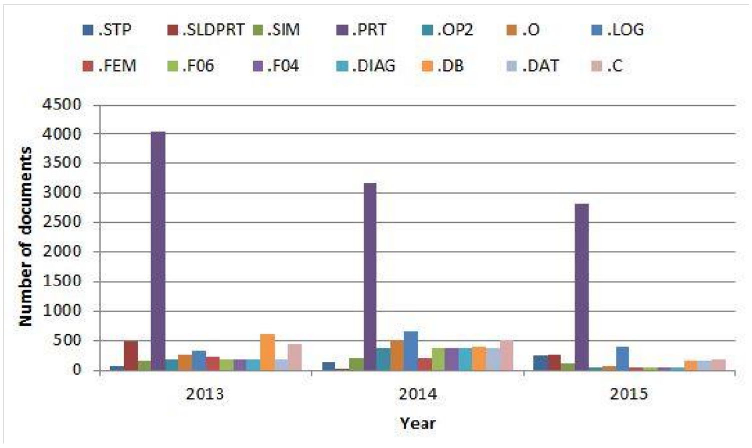


Figure 10. Some of the most frequent technical non-textual document formats

Overall, varied document formats makes it challenging to extract potentially embedded textual content. Several convertor applications are applied for image processing to extract text associated with images and also from more frequent technical document formats. However, the text extraction results were not satisfactory due to incomplete or incorrect spelling of extracted terms which is not sufficiently meaningful to give a description of the document. Similar to what was mentioned for textual documents, document name and path might contain useful content that helps in identifying subject of the given document. In the content analysis section below, we elaborate the use of other available textual metadata.

Contextual ontology as knowledge domain

As mentioned in literature review, ontologies have been utilized by enterprises for knowledge management and sharing in many cases. Our objective is to describe and structure our knowledge domain by developing a tailored ontology and to investigate how the built ontology can improve knowledge sharing and reusability through the recommender system.

For creating the ontology, we chose the NeOn methodology since it supports a knowledge reuse approach (Suárez-Figueroa, Gómez-Pérez, & Fernández-López, 2012). The first scenario among 9 different scenarios by this methodology is suitable for our system which consists of 8 tasks (Mehrpoor et al., 2015). In the early tasks of Ontology Requirements Specification Activity, ORSA, the purpose and scope of creating the ontology is identified by holding a set of semi-structured interviews with users from different disciplines (more particularly with the system engineer) with focus of users' regular tasks in the project, their information needs and information seeking behavior (see Appendix A). In addition, other available resources such as master theses of past years were used to improve the collected information about ontology environment. According to the results of the interviews, the scope of our knowledge domain should cover main concepts that characterize the work context of users during the project life-cycle.

During the middle tasks of ORSA, functional and non-functional requirements of the ontology are identified. As a non-functional requirement, the terminology of the ontology should be able to support users' identified requirements. For identifying functional requirements, a list of Competency Questions (CQ) is prepared and posed to users (See Appendix B). The collected results out of CQs and their answers are categorized in three groups of engineers' discipline, engineers' work tasks, and machine components. Thereafter, generic concepts of the ontology are identified that cover main aspects of engineers' work context and also a pre-glossary of the terms is extracted as specific concepts and assigned to their respective generic concepts. Figure 11 depicts the generic and specific concepts of the ontology developed by Protégé[1]. The conceptualization stage is iterated to check the validity of concepts with leaders of each discipline. Then, validated terminologies are formalized by means of ontology super-classes and their associated sub-classes. A detailed partial view of the final ontology is represented in Appendix C; the whole ontology consists of 134 concepts.

[1]. protege.stanford.edu/

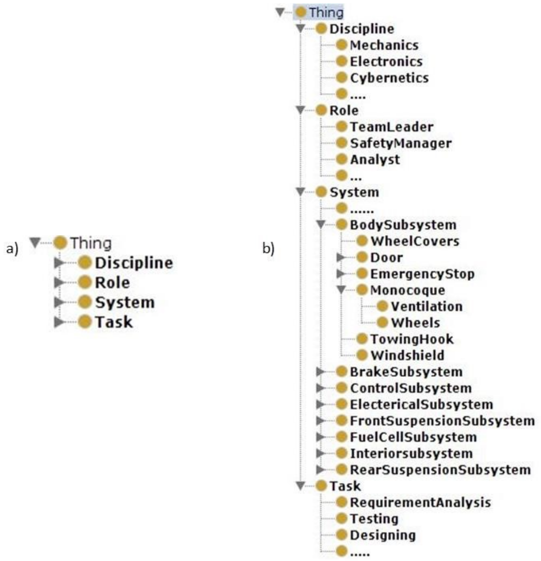


Figure 11. a) Generic concepts of the ontology (The upper ontology section). b) A detailed view of some sub-classes for each concept (The domain-specific ontology section)

Collecting required metadata for document indexing and annotation

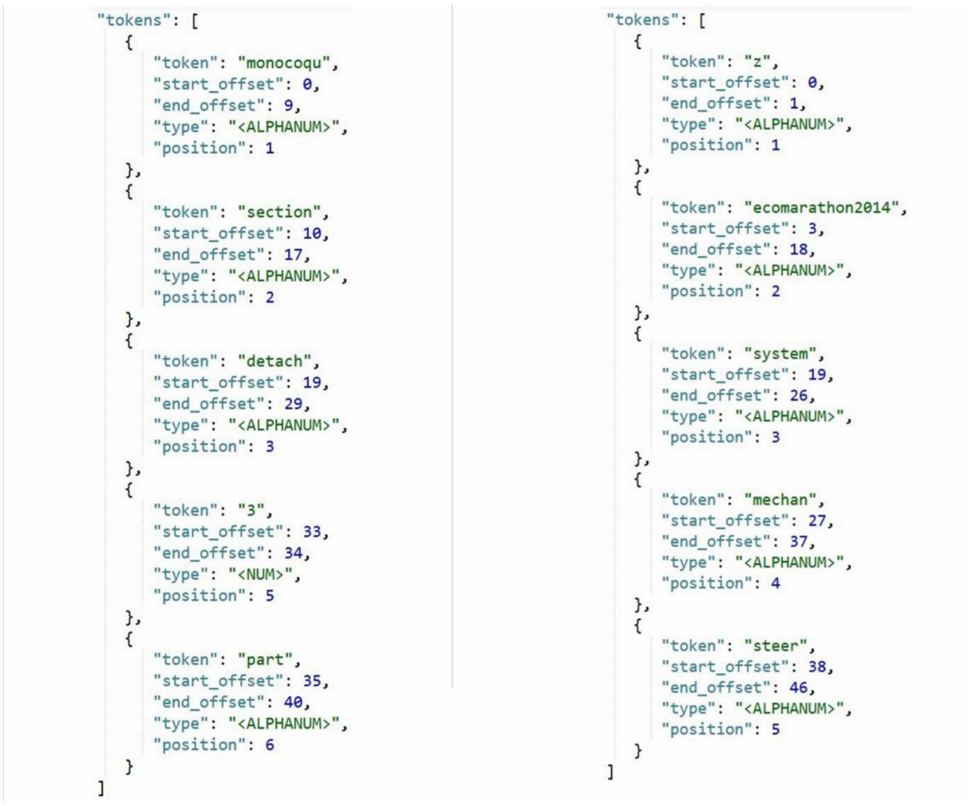
In order to make documents searchable, the first step is to assess what information is extractable from documents if there is any metadata available. A list of key features is collected from different resources and the reasons of choosing them are elaborated as well. Each of the features could be a lead in users' decision making process in identifying relevant information. These features are stored in so-called fields in the indexing process. Some of them are explored from file system metadata such as document name, size, format, year and path. Textual content and possible short description are extracted using Apache Tika. Note that these are only provided for textual documents. For non-textual documents, no content is extracted, leading to a shorter list of features. All the fields for indexing the documents are listed below:

- Document name: it is one of the main properties that users look into while looking for relevant information.
- Document short description: early sentences of a document might be a lead to its subject.
- Document content: it contains substantially more text to match with users' information need and therefore a rich resource to identify relevant information.
- Document format: it is closely related to users' technical background, roles and also their assigned tasks.
- Document size: larger documents identified as more relevant would contain more valuable and reusable inputs. Conversely, although some documents might have quite relevant names, not much information is embedded in them, which could be inferable by their size.
- Year of competition: users are interested to see what is done in a particular year.

- Document path: quite useful because of the hierarchical structure of the file system; it guides users to those resources that are located in deeper levels of hierarchies, and surrounding documents might be interesting for users as well. Moreover, the document path is useful in annotation and classification of documents (Ahlers & Mehrpoor, 2015).

Information extraction and retrieval libraries are applied for document analysis in content-based filtering. In our research work, Elasticsearch is applied to perform this task. To extract specific textual content, specific analyzers are required. For indexing a document, an analyzer tokenizes a given content into individual terms and stores them in an index. It also performs other operations such as removing punctuations and common words, stemming (reducing words to a root form) and any other required specific operations.

In terms of language, the majority of documents are in English and our knowledge domain is also in the same language. Therefore, the Elasticsearch English analyzer is selected since it understands the rules of English grammar and also supports stemming. According to our recommendation approach, the frequency of indexed terms is important for each document. The analyzer enables counting any existing word from the same root. For example, the root term for terms "detachable", "detached" and "detachment" will be "detach". This leads to improved term frequency counts and improves the recommendation process which will be discussed in our later works. As using the analyzer changes textual content, certain fields such as document name and document path and metadata are also kept in the raw (original) format for the purpose of presentation in the recommender system interface. Figure 12 illustrates how fields with different textual structures are indexed and how terms are transformed. As shown in part a), a simple text is split into five stemmed tokens and common words in English are removed and not indexed. Part b) shows the analysis result of a document path that is split into five tokens. Finally, all archived documents are indexed and become structured and searchable. Following the recommendation approach, we need to evaluate how well the built ontology is compatible with indexed documents. The results of the evaluation are addressed in the next section.



```
"tokens": [
  {
    "token": "monocoqu",
    "start_offset": 0,
    "end_offset": 9,
    "type": "<ALPHANUM>",
    "position": 1
  },
  {
    "token": "section",
    "start_offset": 10,
    "end_offset": 17,
    "type": "<ALPHANUM>",
    "position": 2
  },
  {
    "token": "detach",
    "start_offset": 19,
    "end_offset": 29,
    "type": "<ALPHANUM>",
    "position": 3
  },
  {
    "token": "3",
    "start_offset": 33,
    "end_offset": 34,
    "type": "<NUM>",
    "position": 5
  },
  {
    "token": "part",
    "start_offset": 35,
    "end_offset": 40,
    "type": "<ALPHANUM>",
    "position": 6
  }
]
```

```
"tokens": [
  {
    "token": "z",
    "start_offset": 0,
    "end_offset": 1,
    "type": "<ALPHANUM>",
    "position": 1
  },
  {
    "token": "ecomarathon2014",
    "start_offset": 3,
    "end_offset": 18,
    "type": "<ALPHANUM>",
    "position": 2
  },
  {
    "token": "system",
    "start_offset": 19,
    "end_offset": 26,
    "type": "<ALPHANUM>",
    "position": 3
  },
  {
    "token": "mechan",
    "start_offset": 27,
    "end_offset": 37,
    "type": "<ALPHANUM>",
    "position": 4
  },
  {
    "token": "steer",
    "start_offset": 38,
    "end_offset": 46,
    "type": "<ALPHANUM>",
    "position": 5
  }
]
```

Figure 12. An English analyzer is applied for different types of textual fields; a) an example of indexing a simple text. Input text: "monocoque section, detachable in 3 parts" b) an example of indexing text with path structure. Input text: "Z:\ecomarathod2014\Systems\Mechanical\Steering\"

Evaluation of the constructed ontology against the corpus

The evaluation is designed to estimate which terms from the ontology can, in raw or processed form, be used to match documents that contain the same or similar keywords. To ease this process, we use an Elasticsearch function called percolator. The normal operation of a search system is to retrieve those documents from a corpus that match a given query. The percolator works in the opposite direction. Generated queries from the ontology are indexed based on the same schema used for documents and then the documents are matched against the indexed queries to see which queries match with documents. Figure 13 depicts the functionality of percolator. This gives us an easy initial measure of the number of documents that can be matched by terms from our developed ontology.

To give a more technical description, a list of queries are generated that each query is pointing to a specific ontology concept that might contain one or several terms. Therefore, the number of generated queries is equal to the number of ontology concepts, 134 queries. The important point in query indexing process is to index queries exactly based on the same schema that is used for indexing documents. Otherwise, documents will not match to the query since the given field might not be compatible. After two indexes are ready, the evaluation process is started by running the query index against indexed documents one by one. This experiment reveals the level of retrievability and coverage of our ontology against the document corpus.

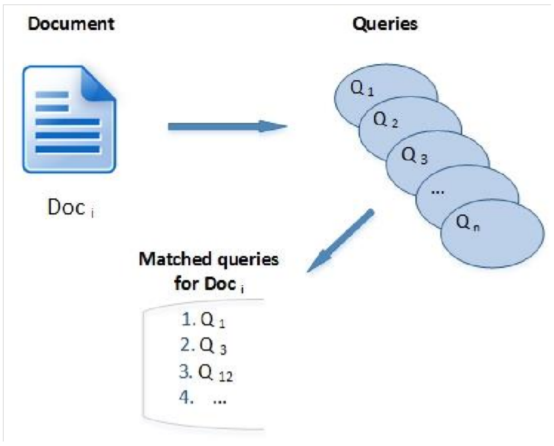


Figure 13. Percolator functionality: Matching a document against a group of queries

Evaluation results for textual documents

We start with statistical results shown in Figure 14 on the numbers of documents that generated queries could catch. Generated percolators represented on the x-axis show all the ontology concepts and the y-axis shows the number of matched documents per built percolator/ontology concept. The results show that more core and generic concepts of the ontology could match more documents. Those ontology concepts that contain any of the terms “control”, “motor” or “system” could catch substantial number of documents as a match, which is not that surprising when comparing to more specialized terms. Note that a document could match with multiple queries. Specific concepts such as components of machine subsystems catch less number of documents compared to generic terms, but will have a higher matching specificity. Certain terms match only a few documents, but all ontology concepts could be matched to documents in the corpus.

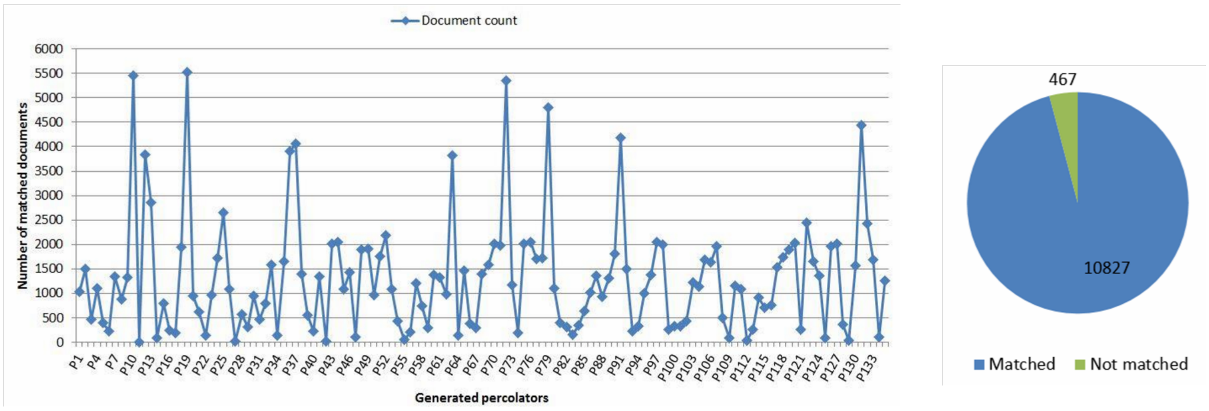


Figure 14. Number of matched textual documents with ontology concepts

Looking at this from the document side, the pie chart in Figure 14 shows that around 95 percent of available textual documents retrievable by direct ontology concepts without any adaptations or relaxations in the matching and only 5 percent of documents are not retrievable, i.e. are not matched by any ontology concept term. This would be an important fraction, so we examine these in detail in terms of their name, format and content. In Figure 15, these 5% of non-matched documents are examined based on their file type. We see that more than half of them are PDF documents, and around two fifth are Word and text documents, with a small number of other document formats.

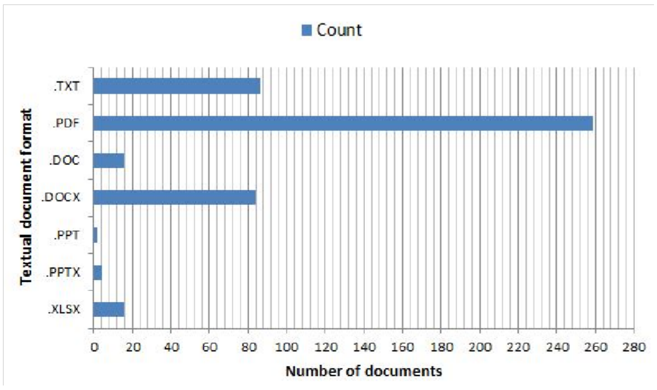


Figure 15. Number of non-matched document from different formats in textual group

In order to assess why these documents could not be matched, we examine their textual content in more detail. Figure 16 represents how many terms could be extracted from non-matched documents. No terms could be extracted from about a third of documents, mostly from PDFs. These documents can be scanned PDFs or documents with image content only as we had discussed above. One fifth of documents contain a maximum of 100 extractable terms and only around one third of documents contain more than 100 terms. These documents mainly belong to administrative, financial and personal directories where non-technical content is stored such as traveling information, expenses etc. In addition, a few documents are in Norwegian language. As the created ontology is in English, this limits the matching process for non-English terms. It shows also that future work could work towards a multilingual ontology.

It is worth mentioning that all the document names and document paths of those non-matched textual documents did not have meaningful technical terms to be matched with ontology concepts. Usually, file name and path are more valuable in the absence of file content, but in this case both main features failed to be useful. It is observed that documents and paths are mostly named with people's name, numbers, combination of letters, joint words e.g., “systemanalysis” or Norwegian terms.

Evaluation results for non-textual documents

We repeat the previous process for non-textual documents where we can match only file name and file path against ontology concepts. With less content to work with, fewer matched documents are expected as represented in Figure 17. The matching results show a drop to around 43% compared to the previous 95%. Among matched documents, there is a similar trend as for textual documents. Generic and core concepts of the ontology could again match with more documents and as terms become more specific, the number of matched documents decreases. Yet, only 6 queries could not find any match, which is surprising, given the usually lower information density in these features. Although no textual content is extracted from documents of this group, fields of document name and path contain useful content to make a group of non-textual documents retrievable by ontology concepts.

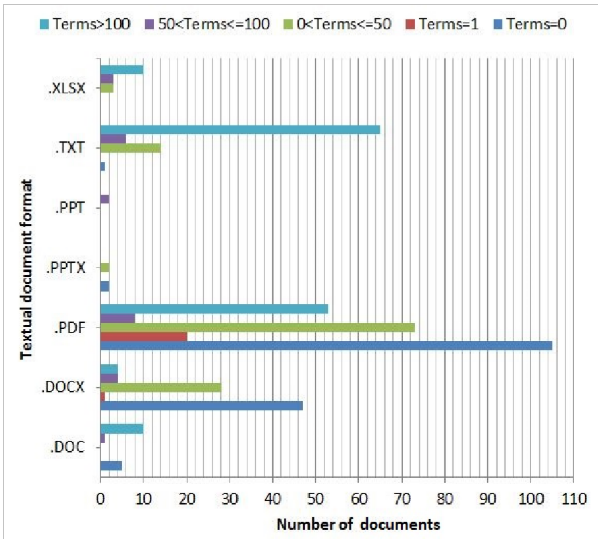


Figure 16. Number of terms embedded in non-matched textual documents

We then investigate the reason of mismatching more than half of the non-textual documents as seen in the pie chart of Figure 17. Choosing a random sample, we observe that we have the same challenge of insufficient naming system for both document name and path. Analytical results reveal that many of the documents that could not match to any of the queries are from the same cluster as before. This cluster covers personal information that refers to team members, many photos captured during project development from team activities per year etc. Furthermore, duplicated documents are observed quite often in both groups of textual and non-textual documents even within the same year.

Overall, mostly those documents that are located in directories with specific and suitable names or having such terms in the file name – often containing machine system structure terms – are retrievable by ontology concepts. Besides the content-based recommendation approach, a complementary approach is required to identify those important non-textual documents that could not be caught by ontology concepts. This would be an argument for extending the approach we outlined in (Ahlers & Mehrpoor, 2015) of how to use clustering and relation analysis to spread semantic labels.

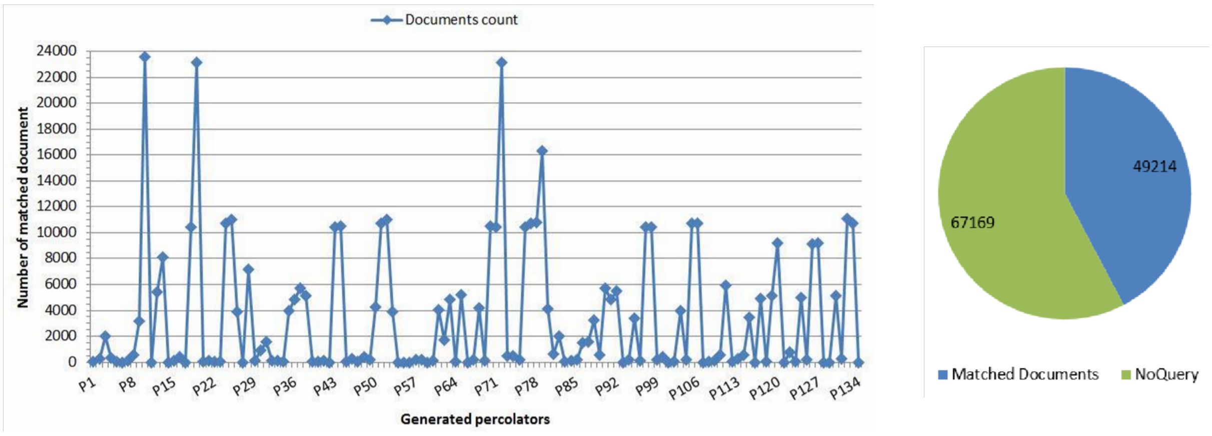
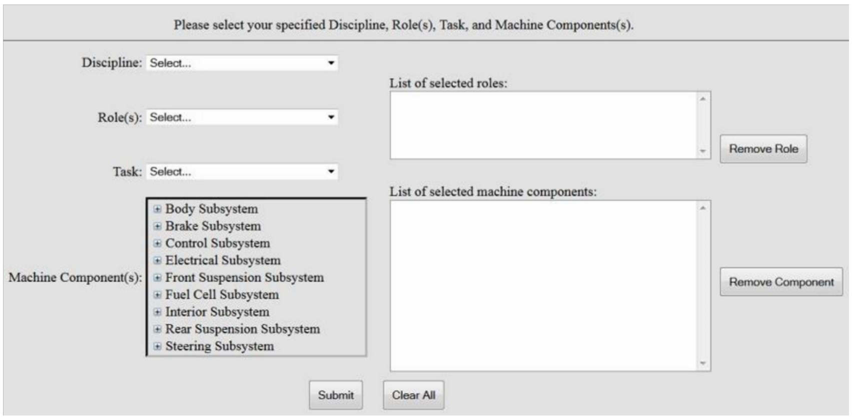


Figure 17. Number of matched non-textual documents with ontology concepts

Constructed ontology and profiling approach

In the professional scope, the context of users describes their interests. Our ontology can model users' preferences since it consists of different dimensions of users' work context. As described in (Mehrpoor et al., 2015), our ontology can be understood as the combination of all possible static user profiles which means that we actually create profiles at ontology level and not as user level in the target recommender system. The content of these profiles will be used later in the developed recommender system to provide more relevant documents for users. Figure 18 depicts the user interface for identifying users' work context from ontology concepts. A user's work context is mutable during project development. By combining different concepts of the ontology, dynamic user profiles are built. Using dynamic profiles enables users to create varied fine-grained user profiles on the fly and narrow down their scope of context. Figure 19 illustrates a preview of a selected work context.



Please select your specified Discipline, Role(s), Task, and Machine Components(s).

Discipline: Select...

Role(s): Select...

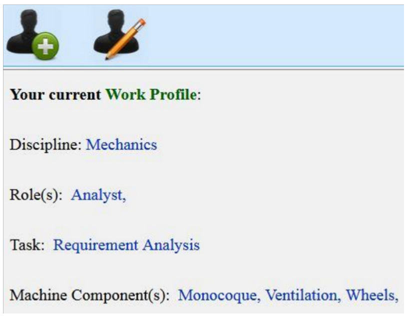
Task: Select...

Machine Component(s):
- ⊞ Body Subsystem
- ⊞ Brake Subsystem
- ⊞ Control Subsystem
- ⊞ Electrical Subsystem
- ⊞ Front Suspension Subsystem
- ⊞ Fuel Cell Subsystem
- ⊞ Interior Subsystem
- ⊞ Rear Suspension Subsystem
- ⊞ Steering Subsystem

List of selected roles:

List of selected machine components:

Figure 18. User interface snapshot for selecting work concept driven from ontology



Your current Work Profile:

Discipline: Mechanics

Role(s): Analyst,

Task: Requirement Analysis

Machine Component(s): Monocoque, Ventilation, Wheels,

Figure 19. A view of selected work context (Dynamic work-profile)

Future work

In our future work, we aim to further evaluate our proposed system in the engineering context by evaluating the initial and revised systems with a group of users. We aim to investigate the level of accuracy and relevance of search results against different users' work context in the traditional engineering project. Moreover, we investigate our non-personal profiling approach and assess the idea of generating semantic content for ontology concept profiles using the results of users' explicit feedbacks and improving the profile content gradually during the system lifecycle. Through this approach, collected terms in concept profiles could be useful in redesigning the ontology and improving the potential cold-start problem with existing ontology concepts. This would give us an interesting angle to extend and grow the ontology naturally. Overall, in our future work we will evaluate the proposed system from different quality perspectives to be able to deliver more precise recommendation results for this challenging domain.

Discussion and Conclusion

In this work, we have analyzed a document corpus from a multidisciplinary engineering project, to better understand the setting of an engineering-domain recommender system. The objective of such a system is to alleviate the problem of information overload by providing relevant knowledge and information for engineers from different disciplines by focusing on their work context. Our study consists of three main contributions.

First, we analyzed in depth a live, in-the-field document corpus from real projects to understand the structure of existing documents in the engineering context. While this has been only one project, we have seen other very similar systems in our work, even if we could not analyze them in depth. Also, appropriate information extraction and retrieval tools have been applied to analyze valuable input of documents along with annotating the documents semantically.

Second, we investigated the requirements and specifications of users through semi-structured interviews with engineers of different disciplines. The focus of the interviews was to identify users' regular tasks, their information needs and information seeking challenges. The information gained out of the interview results outlined the scope of our knowledge domain to cover the work context of users during the project.

Third, we constructed an ontology as our knowledge domain and verified its use and suitability for the engineering domain by validating it against the document corpus. All the concepts that characterize users' work context are collected and represented by building domain specific ontology. In order to assess the retrievability and coverage of ontology concepts against the documents, we examined our ontology-based content matching approach with advanced search tool to measure the performance of created knowledge domain.

The results show that the provided knowledge domain could cover a majority of documents, with up to 95% for textual and still 43% for non-textual files. Regarding non-textual documents, additional metadata and annotations could be a complementary solution to our approach to improve information access and retrievability. In addition, the results indicate a cold-start problem for some concepts that are either too wide or too narrow according to the level of matching with the document corpus. This causes retrieving too many or too few documents. However, it proves valuable insight to refine the matching towards better specificity for certain concepts based on the document corpus, especially for very specific concepts. Refinements and improvements in the ontology development can then focus on relations between concepts and identifying more semantically related terms per concept to cover more semantically relevant results and improve the cold-start problem. Any combination of ontology concepts could create dynamic work profile for users in different stages of the project. Therefore varied fine-grained work profiles could be created from user side in real-time. This feature enables the recommender system to dynamically narrow or widen the scope of search and focus on current user preferences and thus provide recommendation that is closer to his information needs.

Our findings here and the overall research have relevant implications for academia and industry. We point to a common challenge of information overload and knowledge reuse in multidisciplinary engineering teams. This links to improved engineering processes in all fields to improve efficiency and reduce waste. We expect the research to be generalizable with limited constraints. The results of our experiments on the specifications of knowledge and information resources reveal key aspects that should be considered for analysis of such contexts and domain ontologies. In addition, we examined and validated that the approach of building a domain-specific ontology for the engineering domain for a semantically improved recommender system is suitable for a typical scenario that collects a huge number of varied documents in a shared file system for an engineering project; and this approach forms a suitable basis for our information access system for the engineering case.

References

Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems *Recommender systems handbook* (pp. 217-253). Springer.

Ahlers, D., & Mehrpoor, M. (2015). *Everything is filed under 'File': Conceptual Challenges in Applying Semantic Search to Network Shares for Collaborative Work.* Proceedings of the 26th ACM Conference on Hypertext & Social Media Hypertext 2015.

Ahlers, D., Mehrpoor, M., Kristensen, K., & Krogstie, J. (2015). *Challenges for information access in multi-disciplinary product design and engineering settings.* Tenth International Conference on Digital Information Management (ICDIM 2015).

Azzopardi, L., & Vinay, V. (2008). *Retrievability: an evaluation measure for higher order information access tasks.* Proceedings of the 17th ACM conference on Information and knowledge management CIKM.

Bhagwat, D., & Polyzotis, N. (2005). Searching a file system using inferred semantic links. Proceedings of the sixteenth ACM conference on Hypertext and hypermedia, Austria.

Bruno, G., Antonelli, D., Korf, R., Lentes, J., & Zimmermann, N. (2014). Exploitation of a semantic platform to store and reuse PLM knowledge. IFIP International Conference on Advances in Production Management Systems, Springer Berlin Heidelberg.

Buodd, M., & Halsøy, B. (2015). DNV GL Fuel Fighter towards Shell Eco-marathon 2015. Master thesis: NTNU, Trondheim.

Burke, R., Felfernig, A., & Göker, M. H. (2011). Recommender systems: An overview. *AI Magazine, 32*(3), 13-18.

Burke, R. D., & Ramezani, M. (2011). Matching Recommendation Technologies and Domains chapter 11. *Recommender systems handbook, 1*, 367.

Chang, X., Sahin, A., & Terpenny, J. (2008). An ontology-based support for product conceptual design. *Robotics and Computer-Integrated Manufacturing, 24*(6), 755-762.

Eck, O., & Schaefer, D. (2011). A semantic file system for integrated product data management. *Advanced engineering informatics, 25*(2), 177-184.

El Kadiri, S., & Kiritsis, D. (2015). Ontologies in the context of product lifecycle management: state of the art literature review. *International Journal of Production Research, 53*(18), 5657-5668.

Esa, A. M., Taib, S. M., & Thi, H. N. (2010). *Prototype of semantic search engine using ontology.* IEEE Conference on Open Systems (ICOS), 109-114.

Ge, J., Chen, Z., Peng, J., & Li, T. (2012). *An ontology-based method for personalized recommendation.* 11th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), 2012 IEEE.

Gifford, D. K., Jouvelot, P., Sheldon, M. A., & James W. O'Toole, J. (1991). Semantic file systems. *SIGOPS Oper. Syst. Rev., 25*(5), 16-25.

Horn, B. L. (2016). Computer System For Automatic Organization, Indexing And Viewing Of Information From Multiple Sources: US Patent 20,160,117,071.

Huang, N., & Diao, S. (2008). Ontology-based enterprise knowledge integration. *Robotics and Computer-Integrated Manufacturing, 24*(4), 562-571.

Jayavel, S., Anouncia, M., & Kapoor, A. (2013). Semantic Search Engine. *International Journal of Recent Contributions from Engineering, Science & IT (iJES), 1*(2), pp. 19-21.

Johnson, B. C., Manyika, J. M., & Yee, L. A. (2005). The next revolution in interactions. *McKinsey Quarterly, 4*, 20-33.

Jones, W., Phuwanartnurak, A. J., Gill, R., & Bruce, H. (2005). *Don't take my folders away!: organizing personal information to get things done.* CHI'05 extended abstracts on Human factors in computing systems.

Kang, J., & Choi, J. (2011). *An ontology-based recommendation system using long-term and short-term preferences.* International Conference on Information Science and Applications (ICISA), 2011.

Kristensen, K., Krogstie, J., Ahlers, D. & Mehrpoor, M. (2016). LEAP Collaboration System. Chapter 5, in: The Methods and Tools of the Linked Engineering and Manufacturing Platform (LEAP), Academic Press.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends *Recommender systems handbook* (pp. 73-105): Springer.

Mahmood, T., & Ricci, F. (2009). *Improving recommender systems with adaptive conversational strategies.* Proceedings of the 20th ACM conference on Hypertext and hypermedia.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1): Cambridge university press Cambridge.

Mehrpoor, M., Gjarde, A., & Sivertsen, O. I. (2014). *Intelligent services: A semantic recommender system for knowledge representation in industry.* International ICE Conference on Engineering, Technology and Innovation (ICE), 2014.

Mehrpoor, M., Gulla, J. A., Ahlers, D., Kristensen, K., Ghodrat, S., & Sivertsen, O. I. (2015). *Using process ontologies to contextualize recommender systems in engineering projects for knowledge access improvement.* ECKM2015.

Owen, J., Burstein, F., & Mitchell, S. (2004). Knowledge Reuse and Transfer in a Project Management Environment. *Journal of Information Technology Case and Application Research, 6*(4), 21-35. doi:10.1080/15228053.2004.10856052

Parise, S., Guinan, P. J., Iyer, B., Cuomo, D. L., & Donaldson, B. (2009). Harnessing Unstructured Knowledge: The Business Value Of Social Bookmarking At Mitre. Journal of Information Technology Case and Application Research, 11(2), 51-76.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. The adaptive web (pp. 325-341): Springer.

Rao, L., Mansingh, G., & Osei-Bryson, K.-M. (2012). Building ontology based knowledge maps to assist business process re-engineering. Decision Support Systems, 52(3), 577-589.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2011). Recommender systems handbook (Vol. 1): Springer.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems *The adaptive web* (pp. 291-324): Springer.

Soules, C. A. N., & Ganger, G. R. (2005). Connections: using context to enhance file search. *SIGOPS Oper. Syst. Rev., 39*(5), 119-132. doi:10.1145/1095809.1095822

Staab, S., & Studer, R. (2013). *Handbook on ontologies*: Springer Science & Business Media.

Suárez-Figueroa, M., Gómez-Pérez, A., & Fernández-López, M. (2012). The NeOn Methodology for Ontology Engineering. In M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, & A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 9-34): Springer Berlin Heidelberg.

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., & Duval, E. (2012). Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies, 5*(4), 318-335.

Williams, B., Figueiredo, J., & Trevelyan, J. (2013). Finding workable solutions: Portuguese engineering experience. *Engineering Practice in a Global Context: Understanding the Technical and the Social*.

Zhen, L., Huang, G. Q., & Jiang, Z. (2010). An inner-enterprise knowledge recommender system. *Expert Systems with Applications, 37*(2), 1703-1712.

Zhen, L., Wang, L., & Li, J.-G. (2013). A design of knowledge management tool for supporting product development. *Information Processing & Management, 49*(4), 884-894.

Appendix A

List of questions and issues for interview with users

Introduction:

The interview is started with asking about engineer's/user's work tasks during project development.

Questions:

1. How engineers/users usually work with documents?
   a) How do you decide if a document is the right one for you to take a look?
   b) What is important about the document to take your attention in first overview?
      - Title, Format, Date modified any other information?
2. Which sorts of documents are related to your discipline?
3. How much do you use the archived documents of past projects?
4. Are different disciplines related to each other? If yes, which ones? And on which parts and activities? (Collaboration aim)
5. Are there any overlaps in the responsibilities of different roles with each other?
6. What are the project stages?
   a) Are the project stages something regular each year?
   b) In each project stage which disciplines are involved and in which responsibilities?
7. How much duplication is there in the document storage in each year?
8. How much the predefined technical rules from SEM affect the documents that you want to reuse from past years?
9. What sorts of documents are more likely to be reused from past projects?
10. What sort of information sources associates with each task?
11. How do you prioritize the disciplines according to their level of importance in the project?
12. How do you prioritize documents according to their types in your work area?
    a) Descriptions, modeling, guidelines, rules, etc.
13. What are the challenges that you face with while looking for required information and knowledge?
14. What kinds of tasks are assigned to experts and what kind of tasks are assigned to novices?
15. What document storages do you have in DNV GL? And by which one you usually work with?
16. How disciplined the documents are stored in the folders in the information resources?

Appendix B

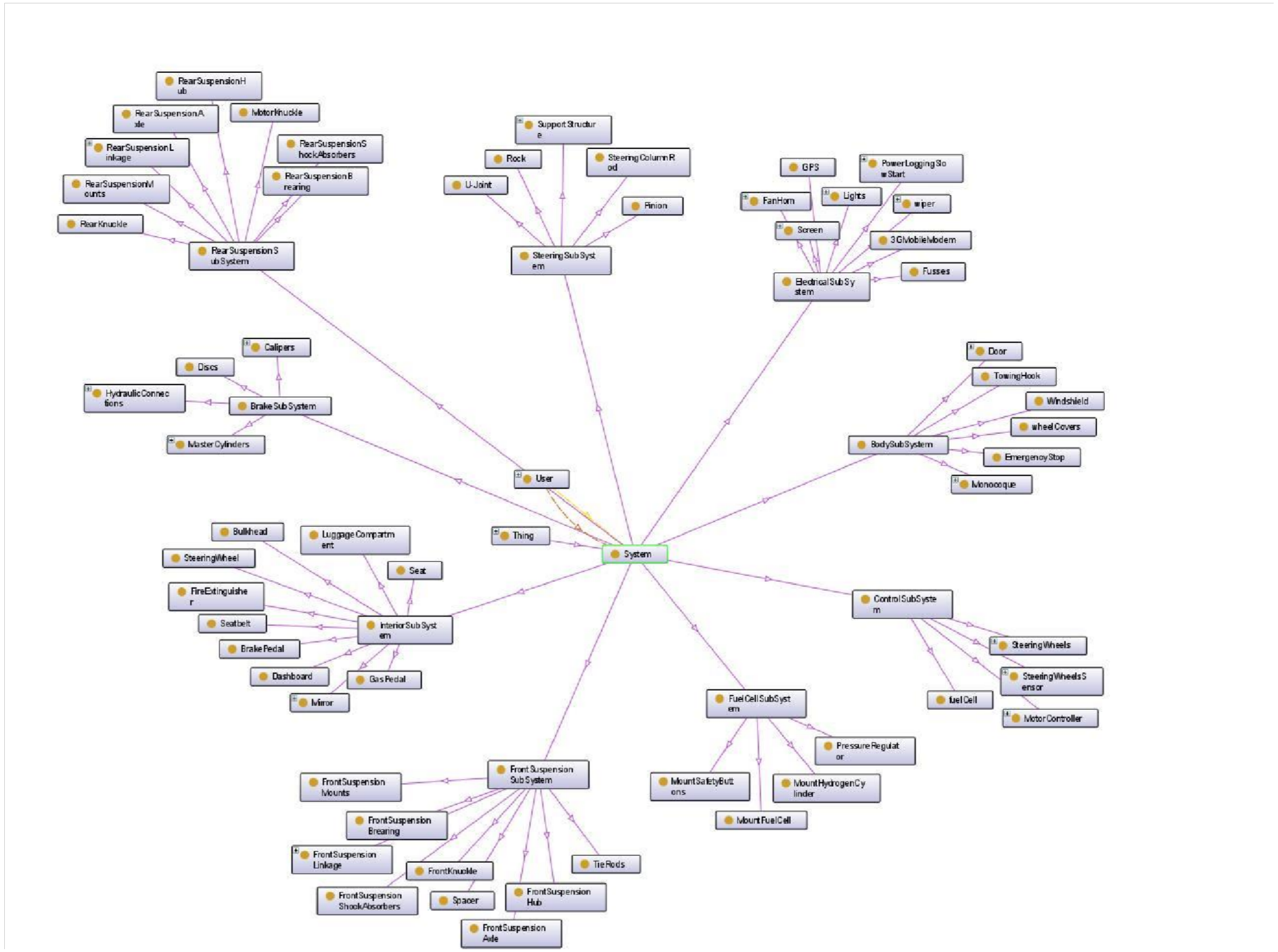
Competency Questions

1. What are different disciplines in DNV GL project?
2. Are different disciplines related to each other? If yes, which ones?
3. What are different roles in each discipline?
4. What are different tasks for each role?
5. Is there any overlapping in the responsibilities?
6. Elaborate each task in each project stage, in each discipline, for each engineering role?
7. Which disciplines are involved in each project stage?
8. What are the main subsystems of the vehicle to be designed?
9. What are the components embedded in each of the subsystems?
10. Which subsystems are related to each other and how?

Appendix C

A Part of implemented ontology

Super classes and subclasses of entity “System”


RearSuspensionHub
RearSuspensionAxle
MotorKnuckle
RearSuspensionLinkage
RearSuspensionShockAbsorbers
RearSuspensionMounts
RearSuspensionBrearing
RearKnuckle
RearSuspensionSubSystem
Support Structure
Rock
Steering ColumnRod
U-Joint
Pinion
Steering SubSystem
GPS
PowerLoggingSlowStart
FanHorn
Lights
wiper
Screen
3GMobileModem
Fusses
ElectricalSubSystem
Calipers
Discs
HydraulicConnections
Brake SubSystem
Master Cylinders
User
Door
TowingHook
Windshield
wheel Covers
BodySubSystem
Emergency Stop
Monocoque
Bulkhead
Luggage Compartment
Thing
System
SteeringWheel
Seat
FireExtinguisher
Seatbelt
InteriorSubSystem
Brake Pedal
Dashboard
Gas Pedal
Mirror
Control SubSystem
SteeringWheels
SteeringWheelsSensor
fuel Cell
Motor Controller
FuelCell SubSystem
Pressure Regulator
MountSafetyButtons
MountHydrogenCylinder
MountFuelCell
Front Suspension Sub System
FrontSuspension Mounts
FrontSuspension Brearing
FrontSuspension Linkage
FrontKnuckle
TieRods
FrontSuspension ShockAbsorbers
Spacer
FrontSuspension Hub
FrontSuspension Axle