Computers in Industry 125 (2021) 103377

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

CACDA: A knowledge graph for a context-aware cognitive design assistant

Armand Huet[a,*], Romain Pinquié[b], Philippe Véron[c], Antoine Mallet[d], Frédéric Segonds[a]

[a] Arts et Metiers Institute of Technology, LCPI, HESAM Université, 75013 Paris, France
[b] Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, Grenoble, France
[c] Arts et Metiers Institute of Technology, LISPEN, HESAM Université, F-13617 Aix-en-Provence, France
[d] Capgemini DEMS, Toulouse, France

ARTICLE INFO

Article history:
Received 3 July 2020
Received in revised form
17 November 2020
Accepted 2 December 2020

Keywords:
Design rule
Knowledge graph
Knowledge management
Context-awareness
Cognitive assistant

ABSTRACT

The design of complex engineered systems highly relies on a laborious zigzagging between computer-aided design (CAD) software and design rules prescribed by design manuals. Despite the emergence of knowledge management techniques (ontology, expert system, text mining, etc.), companies continue to store design rules in large and unstructured documents. To facilitate the integration of design rules and CAD software, we propose a knowledge graph that structures a large set of design rules in a computable format. The knowledge graph organises entities of design rules (nodes), relationships among design rules (edges), as well as contextual information. The categorisation of entities and relationships in four sub-contexts: semantic, social, engineering, and IT – facilitates the development of the data model, especially the definition of the "design context" concept. The knowledge graph paves the way to a context-aware cognitive design assistant. Indeed, connected to or embedded in a CAD software, a context-aware cognitive design assistant will capture the design context in near real time and run reasoning operations on the knowledge graph to extend traditional CAD capabilities, such as the recommendation of design rules, the verification of design solutions, or the automation of design routines. Our validation experiment shows that the current version of the context-aware cognitive design assistant is more efficient than the traditional document-based design. On average, participants using an unstructured design rules document have a precision of 0.36 whereas participants using our demonstrator obtain a 0.61 precision score. Finally, designers supported by the design assistant spend more time designing than searching for applicable design rules compared to the traditional design approach.

1. Introduction

1.1. Context

Although a major skill of designers is their creativeness, industrial design consists in exploring a bounded space of possible designs – the design space. Indeed, the main goal of routine design is to search a constrained design space to locate an optimal solution (Gero et al., 1999). To avoid flawed solutions that are outside of the range of possible designs, companies create design manuals that collect a bewildering array of design rules. A design rule is a constraining prescriptive statement (Fu et al., 2016) – often an unstructured blend of text and graphical objects (equation, table, chart, sketch, etc.) – that limits the freedom of a designer. A large amount of common design principles is available in commercial handbooks, but most design rules are specific to a domain and therefore stored in internal design manuals. A design manual aims at supporting deployed designers for the achievement of a proof design, in compliance with best practices, applicable regulations, as well as design for X (manufacturing, assembly, recycling, etc.) constraints. The content of a design manual reflects the experience acquired by the company in previous programs, the latest technology progress status, and any design improvement driven by recent in-service or manufacturing experience. Deviations from the design principles shall be justified and authorised.

1.2. Problem

In industry, design manuals are large and unstructured documents. When the set of design rules does not lead to a cognitive

* Corresponding author.
E-mail address: armand.huet@ensam.eu (A. Huet).

https://doi.org/10.1016/j.compind.2020.103377
0166-3615/

overload for a designer (Gruszka and Nęcka, 2017), documents are the most efficient storage solution. However, today, for various reasons, a document-based approach is not suitable anymore.

First, the large number of experts and the geographically dispersed teams make the collection and management of design rules in documents cumbersome. Second, when a designer must provide a design free of errors, he or she has no other alternative than to go through the big data and spend a large amount of time to retrieve the applicable subset of design rules. For instance, from our collaboration with industrialists, the collection of design manuals applicable to the design of a recent civil aircraft contains no more than 1771 pages, that is, tens of thousands of design rules. The third limit of unstructured documents is that they make computation difficult not to say impossible without a pre-processing phase to model design rules in a formal structure.

Because of the aforementioned problems and the promising full model-based engineering approach, we state that large and unstructured documents can no longer serve as an efficient solution for storing design rules. There is a need for a context-aware cognitive design assistant that aids designers in the collection, organisation, retrieval, use, and management of design rules.

1.3. Proposal

The term "context-aware cognitive design assistant", that represents our proposal, needs to be discussed. By context-aware (van Engelenburg et al., 2019), we mean that the cognitive design assistant will have the ability to sense and adapt to the current design context so as to provide, without explicit user intervention, relevant information and/or services to the user, where relevancy depends on the user's task (Dey, 2001; Baldauf et al., 2007). Adapted from (Abowd et al., 1999) and (Alexopoulos et al., 2016), a design context is any information that can be used to characterise the situation of a designer. Cognitive assistants, which are also known as expert systems or knowledge-based agents, are "intelligent" computer programs that learn more or less complex problem-solving expertise from human experts to assist human nonexpert in solving similar problems (Marcu et al., 2016). Based on the existing literature, we define a context-aware cognitive design assistant as follows:

*A **context-aware cognitive design assistant** is a seamless, ubiquitous and intelligent computer program that senses relevant information that can be used to characterise the situation of a designer and provide, without explicit user intervention, relevant information and/or services to the designer, where relevancy depends on the designer's task* (Fig. 1).

The main contribution of this paper is twofold. First, we will analyse the operational (black box) view of the context-aware cognitive design assistant to identify the stakeholders and the services the cognitive assistant shall provide to them. Second, based on the identified services, we will derive the knowledge graph that will structure both the design rules and the design context in a format that is computable by the cognitive design assistant.

2. Literature review

Before detailing our contributions, we will discuss the existing cognitive assistants, especially in the engineering domain, as well as the candidate approaches for structuring the knowledge in a computable format that enables a cognitive assistant to provide services that augment the designer intelligence.

2.1. Cognitive assistants

Information retrieval techniques make it possible to search relevant information in a large collection of texts (Faloutsos and Oard, 1998). The main information retrieval capability – keywords search – facilitates the access to textual content, but the lack of a structured representation degrades the performance (Li et al., 2008). Moreover, search engines require the user to have clues for initiating searches in large databases. When a new designer joins a company, he has almost no idea about what pieces of information could help him to find the applicable design rules in a massive dataset. Designers consequently need an assistant with a higher degree of autonomy level that is capable of exploring the dataset before distilling relevant design rules.

Cognitive assistants, also known as expert systems or knowledge-based agents, are seamless "intelligent" computer programs that learn more or less complex problem-solving expertise from human experts so as to assist human nonexpert in solving similar problems (Marcu et al., 2016). To be precise, a cognitive assistant does not have any intelligence except computing power. However, it augments human intelligence to substantially enhance human-machine capabilities and performance (Rouse, 2020). The situation when artificial intelligence (AI) performs as a human's partner or assistant is known as "AI-augmented human" (Madni, 2020).

Cognitive assistants for product design are software that check a set of design rules. Two main approaches exist in the state of the art: algorithmic approach and semantic approach. Given a CAD part, an algorithmic approach consists in checking the geometry and topology of a part with algorithms that look for unsatisfied design constraints (Rouse, 2020; Madni, 2020). However, exploration algorithms do not support informal design rules and their edition requires skills (programming, knowledge engineering, mathematics, etc.) that do not belong to the designer background. Alternatively, several research studies propose a semantic approach (Hariya et al., 2010; García et al., 2020; Huang et al., 2015; Moitra and Palla, 2016). Given a CAD part, this approach translates the geometry and topology into an ontology that allows the detection of design errors by semantic reasoning. For instance, based on this approach, Rangarajan et al. (Rangarajan et al., 2013) develop a design rule checker for sheet metal design. The semantic approach supports various types of design rule and the identification of tacit design rules (Mackenzie and Gero, 1987). However, it requires a laborious manual semantic modelling of the part and design rules. This limits semantic design rule checkers to narrow design domains and a small number of design rules. Moreover, in both approaches, the *a-posteriori* checking of design rules requires at least one rework iteration and consequently avoidable design extra time and costs. To get the design right the first time, there is a need to replace detection by prevention. This is why CAD model quality is a crucial part of design education. Company *et* al. (Company et al., 2015) propose to tackle this issue early in CAD trainees learning process with the introduction of coordinated rubrics of CAD model quality criteria. Design quality education must be continuous during the career of a designer. So far, we have not found any computer-aided preventive approach for the application of design rules. Consequently, only experienced designers in a specific design domain are capable of avoiding design errors by anticipating design constraints.

Boy et al. (1990) explain that in order to maximize a task performances, a cognitive assistant must adapt the automation level of the system depending on the user's expertise. This ability to adapt itself to a given context – context-awareness – is a key feature of cognitive assistant. A system is **context-aware** if it has the ability to sense and adapt to any information that can be used to characterise the situation (the context) so as to provide relevant information and/or services to the user, where relevancy depends on the user's task (van Engelenburg et al., 2019; Dey, 2001; Baldauf et al., 2007; Abowd et al., 1999; Alexopoulos et al., 2016). Dey (Dey, 2001) argues that context is one of those suitcase-like words that we use to conceal the complexity of very large ranges

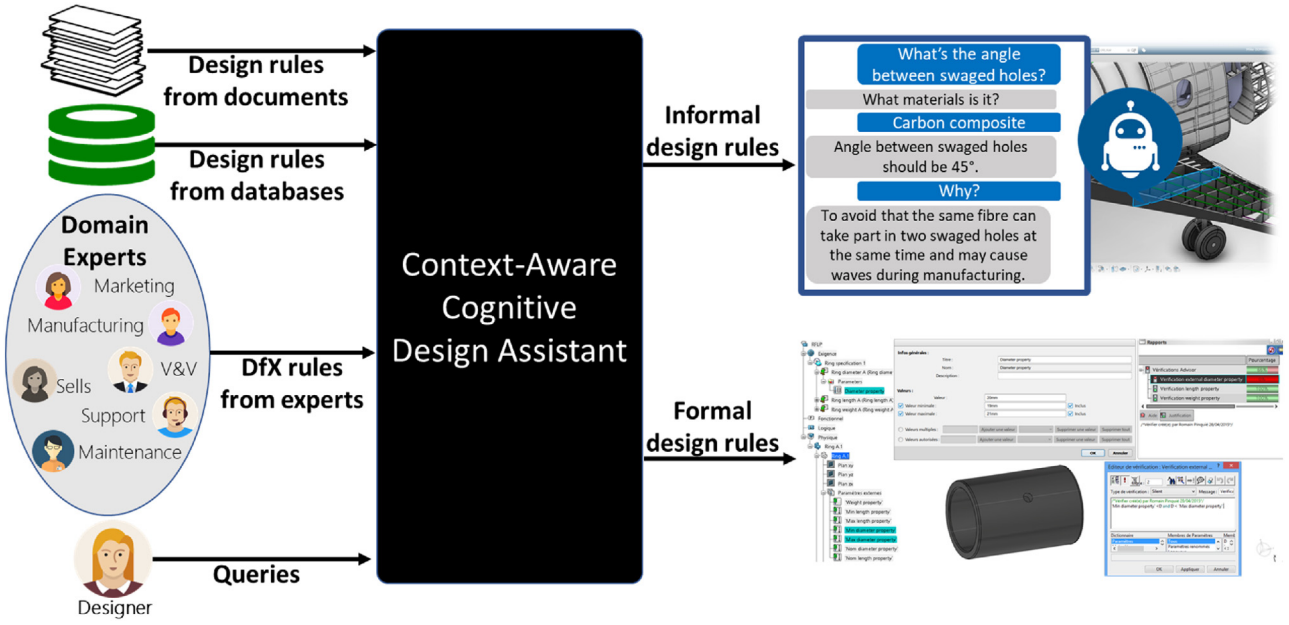


Fig. 1. Context-aware cognitive design assistant.

of different things and attempts to provide a definition : *"Context is any information that can be used to* characterize *the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*. A context includes two kinds of information: domain specific and activity specific (Cassens and Kofod-Petersen, 2006). Domain specific information represents the working environment of the user. It does not evolve according to the user's actions. Activity specific information represents the real time activity of the user. It is therefore evolving according to the user's actions. Da Cunha Mattos et al. (2014) propose a formal representation for context-aware businesses. In the field of manufacturing, context-aware approaches enable workers to receive manufacturing information according to their job and experience (Dhuieb et al., 2016). Although manufacturing differs from design, this example provides us with some details on the definition of the context that includes three viewpoints: operational (activities and tasks of the worker), organisational (team and role of the worker), and user-centric (expertise and skills). Related to our research goal, Rowson et al. (Rowson et al., 2018) investigate the idea of building reusable expert knowledge using screen monitoring and contextual similarity. Nevertheless, the authors assume too many aspects of the framework, such as the form and the content of the knowledge base, the way information is collected, the query language and patterns, the similarity measure, etc. Moreover, we can notice that the context is limited to the interaction of the designer with the CAD software.

The comparison of our research goal with the existing cognitive assistants and context-aware systems shows no solution that helps designers to handle large sets of heterogeneous design rules. However, to provide personalised information and/or services to the user, a cognitive assistant needs a computable structure of the relevant information: a knowledge graph.

2.2. Knowledge graph

A **knowledge graph**, also known as ontology (Ehrlinger and Wöß, 2016), in its broadest sense, is a structured description providing a shared understating of a given domain. The structure of a knowledge graph consists in factual triples where one or many relationships link two entities (Ji et al., 2020). The knowledge graph structure can be used for recommendation and object discovery (Strobin and Niewiadomski, 2014). Researchers have used semantic web techniques including modelling languages (e.g. RDF, RDFS, OWL), query languages (e.g. SQWRL), and software (e.g. Protégé) for structuring and reasoning about domain-specific information such as geometry and topology (Tessier and Wang, 2013; Sanya and Shehab, 2014), feature recognition (Wang and Yu, 2014), generative modelling (Skarka, 2007), connectivistic design reuse (Johansson et al., 2018) or nuclear design rules (Fortineau et al., 2014). Nevertheless, knowledge graphs are extremely time-consuming to be developed and managed. It is therefore interesting to automate the acquisition and processing of knowledge. For example, Johansson *et* al. (Johansson et al., 2018) automate the extraction of structured engineering knowledge from spreadsheets, knowledge-based engineering programs and CAD models. Natural language processing and text mining techniques can be used to process unstructured information sources (Shi et al., 2017; Kang et al., 2015).

A knowledge graph is an abstract representation that does not directly enable a machine to reason. However, when it involves formal semantics, it can be taken as a knowledge base for interpretation and inference over facts (Ji et al., 2020). For instance, the graph-oriented database Neo4j offers the property graph data model where data structures for the schema and/or instances are modelled as graphs and the data manipulation is expressed by graph-oriented operations using the graph query language CYPHER (Angles, 2018). Using NoSQL graph-oriented database systems such as Neo4J is also flexible as one can create, read, update, and delete nodes and relationships without affecting the schema. This is a key advantage since no one would ever come up with a complete knowledge graph at the first attempt.

In fact, there is no consensus on how to methodologically model such a knowledge graph. We rely on previous works (Pinquié et al., 2020; Huet et al., 2020) that propose a first draft of a graph data model. In order to build a knowledge graph adapted to a specific functionality, we can use an inductive approach. It consists in getting back to a service, for example: *"As a designer, I want to know which design rules my design shall satisfy, so that I can provide proof design."*, and to follow a systematic 4-step modelling process:

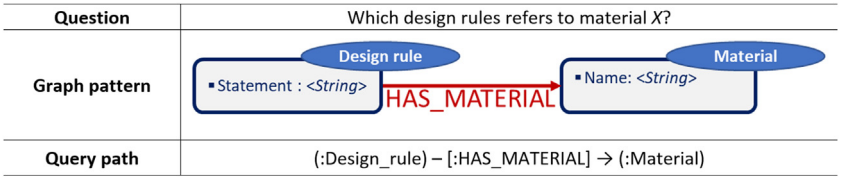
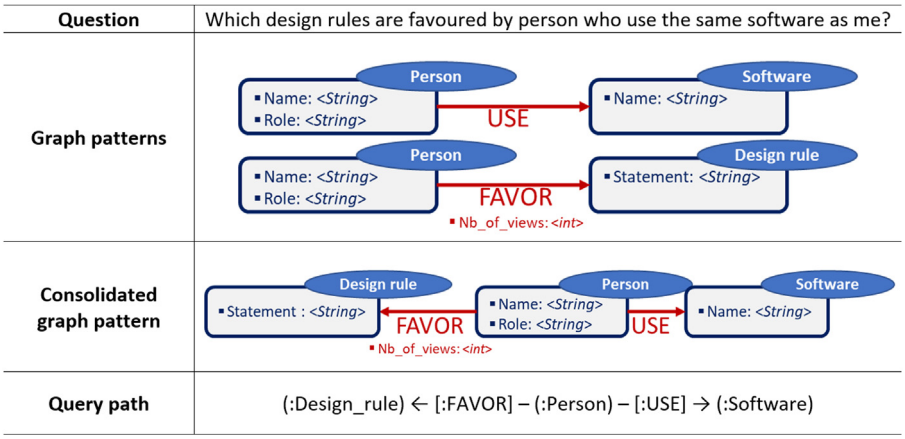


Fig. 2. Graph pattern of the relation HAS_MATERIAL.



Figs. 3 and 4. Presentation of USE and FAVOR relations. Graph pattern of relations FAVOR and USE.

1 Find what questions the design assistant shall help designers to answer;
2 For each question, identify entities (nodes) and relationships (edges);
3 Express each question as a graph pattern.
4 Translate the graph pattern into a query path.

The simplest question to answer is a graph pattern corresponding to a predicate, that is, a triple (Subject – Predicate → Object) as presented in Fig. 2.

A graph-oriented data model brings an added-value when queries traverse richly interconnected data to answer more sophisticated questions as described in Fig. 3.

It is challenging and time consuming to enumerate all questions that a context-aware system shall answer. This is why a complementary deductive approach is needed. Alami Merrouni et al. (Merrouni et al., 2019) assert that a context modeling framework is essential to model the user's contextual information. A context-modelling framework is a taxonomy that identifies and organizes context into relevant categories. Indeed, every context-aware application presented in this review base their context-modelling framework on a taxonomy that defines context categories also known as dimensions. We prefer to use the term of *sub-context* because such elements refine the definition of the context.

However, each context-aware application has its own contextual framework. From one application to another, the number of sub-contexts and their definitions vary greatly depending on the application domain, its services etc. For example, in the information retrieval domain, the most advanced context taxonomy (Tamine-Lechani et al., 2010) contains the sub-contexts "Device", "Spatio-temporal", "User", "Task/problem", "Document" (Fig. 5). Table 1 regroups all existing sub-contexts found in our review of the literature on context-aware applications in information retrieval, education and industrial domain.

We notice that few context-aware models are proposed in industrial domains. The context-aware information retrieval software that supports workers in factories (Dhuieb et al., 2016) is an exception. To the best of our knowledge, there is no knowledge graph for structuring and reasoning about a design context and large sets of heterogeneous design rules.

3. Towards a context-aware cognitive design assistant

To derive the knowledge graph that will structure the design rules in a computable format, we must analyse how end-users will operate the Context-Aware Cognitive Design Assistant. Thus, in this section, we conduct an operational "black-box" analysis to identify who are the stakeholders and which services the assistant shall provide to them.

3.1. Stakeholders

The main stakeholder is the CAD designer. A CAD designer is proficient with a CAD software but not necessarily an expert of the design domain and consequently does not have all applicable design rules in mind. Indeed, for instance, when a large company such as Airbus Aircraft designs a new airplane, the significant part of the team members are short-term engineers provided by a consulting company such as Capgemini. With a high turnover, it is no surprise that designers need an assistance for providing an error-free design. The assistant is also of interest for supporting freshly graduated design engineers.

The knowledge engineer has information retrieval and automation skills. He or she is also familiar with the company different knowledge sources like taxonomies, databases or design manuals. The knowledge engineer makes sure that the assistant is functional and can access all needed domain specific information.

Finally, our assistant needs accesses to the sources of design rules. The primary source is recognised and codified knowledge (Yoshikawa, 1993), that is, design rules explicitly stated in unstructured (e.g. Word, PDF, etc.) or semi-structured (e.g. Excel, XML, etc.) documents. Large manufacturing companies often work with sev-

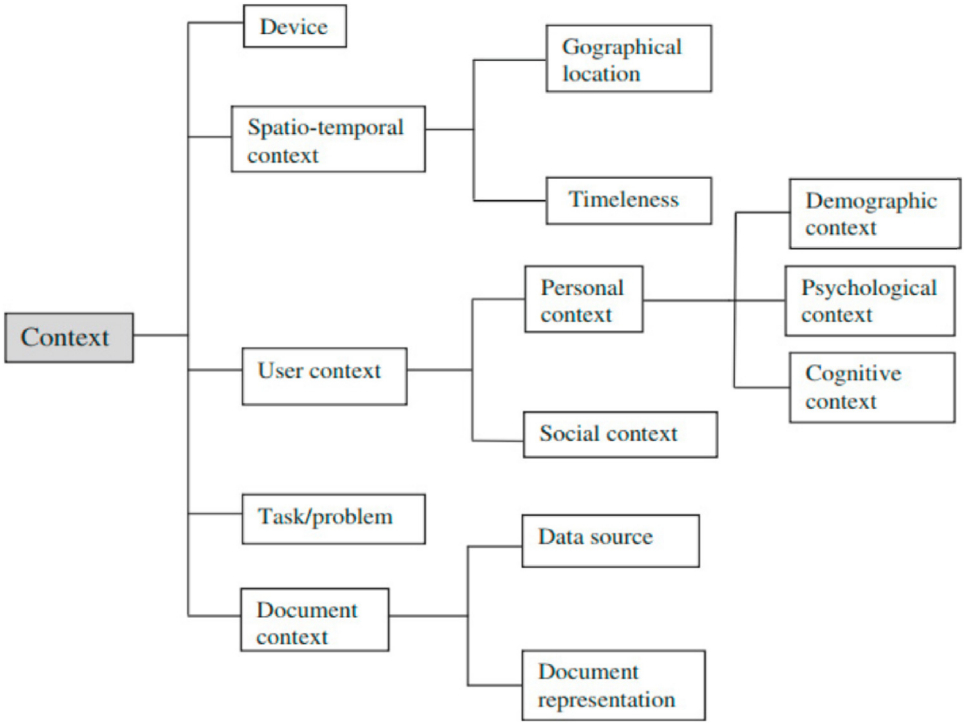


Fig. 5. Multi-dimensional concept of context in information retrieval (Skarka, 2007).

Table 1
Existing sub-contexts in literature.

Sub-Context	Also known as. . .	Definition	Applications
User (Tamine-Lechani et al., 2010; Dhuieb et al., 2015; Hasanov et al., 2019)	Personal context (Kofod-Petersen and Aamodt, 2003) Individual (Ingwersen, 2007) Person (Gu et al., 2004)	Information on the user's profile (e.g. expertise level)	Factories, Learning, Information retrieval, Intelligent mobile applications, General context-awareness
Spatio-temporal (Tamine-Lechani et al., 2010; Hasanov et al., 2019; Kofod-Petersen and Aamodt, 2003)	Location (Gu et al., 2004)	Information on the user's position and movements (e.g. GPS information)	Mobile applications, Learning, Information retrieval, General context-awareness
Operational (Dhuieb et al., 2015)	Task/activity (Tamine-Lechani et al., 2010), (Kofod-Petersen and Aamodt, 2003) Pedagogical (Hasanov et al., 2019) Activity (Gu et al., 2004)	Information on the user's activity (e.g. goals and progression)	Factories, Learning, Information retrieval, Intelligent mobile applications, General context-awareness
Social (Kofod-Petersen and Aamodt, 2003)	Organizational (Dhuieb et al., 2015), Collective (Ingwersen, 2007)	Information on the social interactions among users (e.g. friend of relationship)	Factories, Mobile applications, Information retrieval
Document (Tamine-Lechani et al., 2010)	Intra/inter object (Ingwersen, 2007)	Information on documents (e.g. author name)	Information retrieval
Session (Ingwersen, 2007)	CompEntity (Gu et al., 2004), Technical (Hasanov et al., 2019)	Information on the digital environment the user is interacting with. (e.g. running software)	General context-awareness, Learning, information retrieval
Environment (Hasanov et al., 2019), (Kofod-Petersen and Aamodt, 2003)		Information on the user's physical environment (e.g. noise or humidity level)	Mobile applications, Learning
Device (Tamine-Lechani et al., 2010)		Information on the hardware device the user is using to search information (e.g. display size)	Information retrieval

eral design offices in different countries. This is why design manuals use technical English. It is possible to translate other languages into English before processing. If, by chance, a database stores structured design rules, then it should be accessible by the assistant too. For instance, it is very likely that the design rules stored in a knowledge graph serves for a new version of the product (e.g. aircraft A320 and A320 Neo). The secondary source of design rules includes recognized tacit knowledge (e.g. commonsense) and unrecognized knowledge (e.g. expertise and skill) (Yoshikawa, 1993) belonging to the head of domain experts (e.g. marketing, manufacturing, V&V, sells, support, maintenance, etc.). The assistant will encourage experts to systematize (Yoshikawa, 1993) *design for X* rules.

3.2. Services and constraints

This section details the services the assistant shall provide to the end-users as well as the constraints that stakeholders impose to the cognitive assistant (Fig. 6).

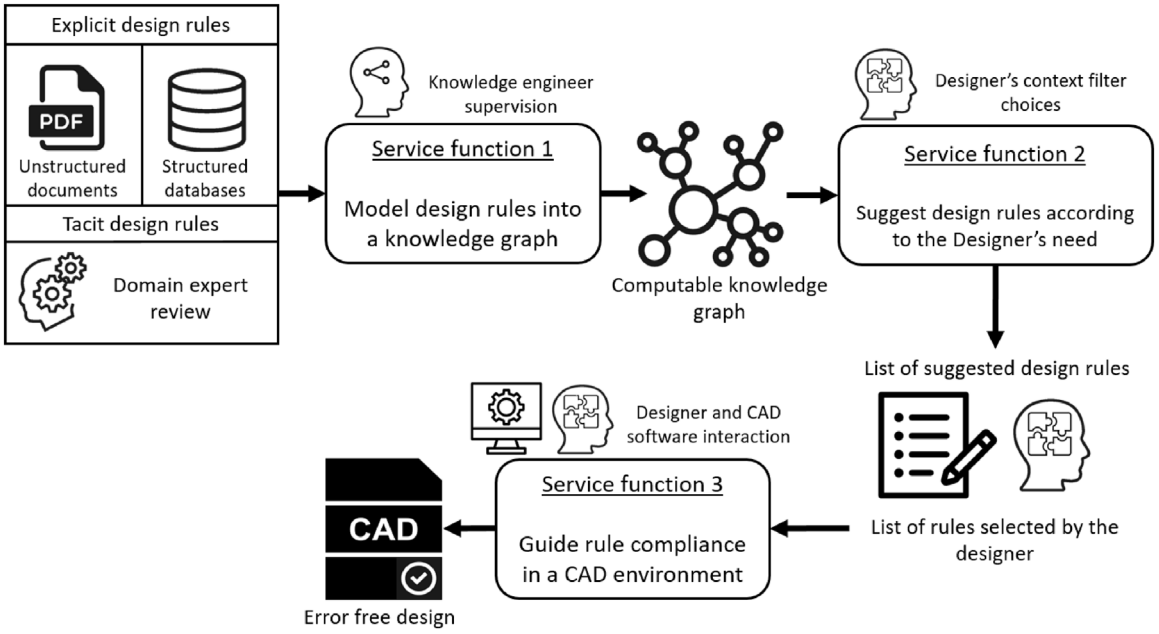


Fig. 6. Full functional process of the Context-Aware Cognitive Design Assistant.

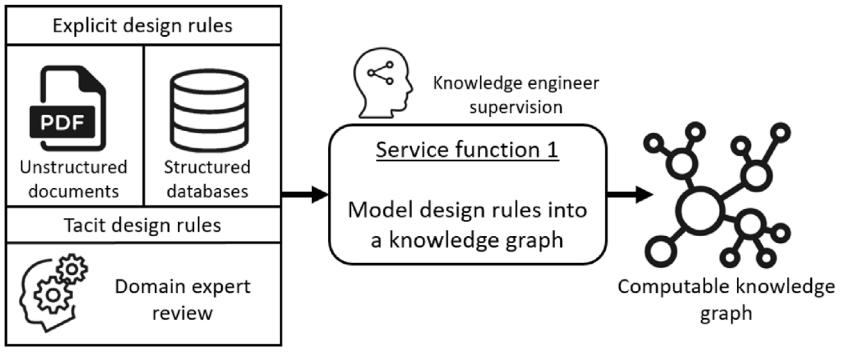


Fig. 7. Schema of service function 1.

■ ***To model design rules in a computable knowledge graph***

First, from an end-user perspective, the assistant shall enable the knowledge engineer to structure the design rules in a computable knowledge graph (Fig. 7). Thus, the assistant shall help the knowledge engineer to identify and extract design rules from unstructured, semi-structured, and structured sources before structuring them in a knowledge graph. The knowledge engineer is also in charge of the change management task if some design rules are updated or deleted. In addition, our assistant shall enable domain experts to systematize tacit rules.

■ ***To suggest design rules according to the designer's need***

o ***To retrieve design rules***

Once the design rules are structured in the knowledge graph, the assistant shall enable the design engineer to retrieve applicable design rules. If the designer knows what design rules he or she is looking for, the conventional approach is to use a full-text search. Based on some keywords, the assistant shall return a list of relevant design rules. Fig. 8 is an operational prototype of the full-text search capability. After searching for the keywords "cutter" and "milling", the assistant returns a subset of relevant design rules.

o ***To navigate design rules***

The list of design rules resulting from a full-text query may be insufficient. Therefore, the assistant shall enable the designer to navigate to similar design rules using query expansion techniques such as dynamic faceted search (Roy et al., 2009; Ben-Yitzhak et al., 2008). For instance, lexical semantics ameliorates some of the problems of mismatched vocabularies by expanding query vectors with words that are lexically related to the original query (Voorhees, 1994). Fig. 8 shows that the assistant dynamically updates the facets on the left according to the queries. By selecting some of the facets, the designers can navigate to similar design rules that are potentially applicable to the design in progress (The facet "tool" is selected in Fig. 8).

o ***To recommend design rules***

Query expansion helps to uncover potentially applicable design rules but it relies on an initial full-text search that requires the designer to have some clues regarding the query. Alternatively, the assistant shall recommend relevant design rules to the designer based on the design context. Thus, by sensing relevant information that can be used to characterise the situation of a designer, the cognitive design assistant can recommend, without explicit user

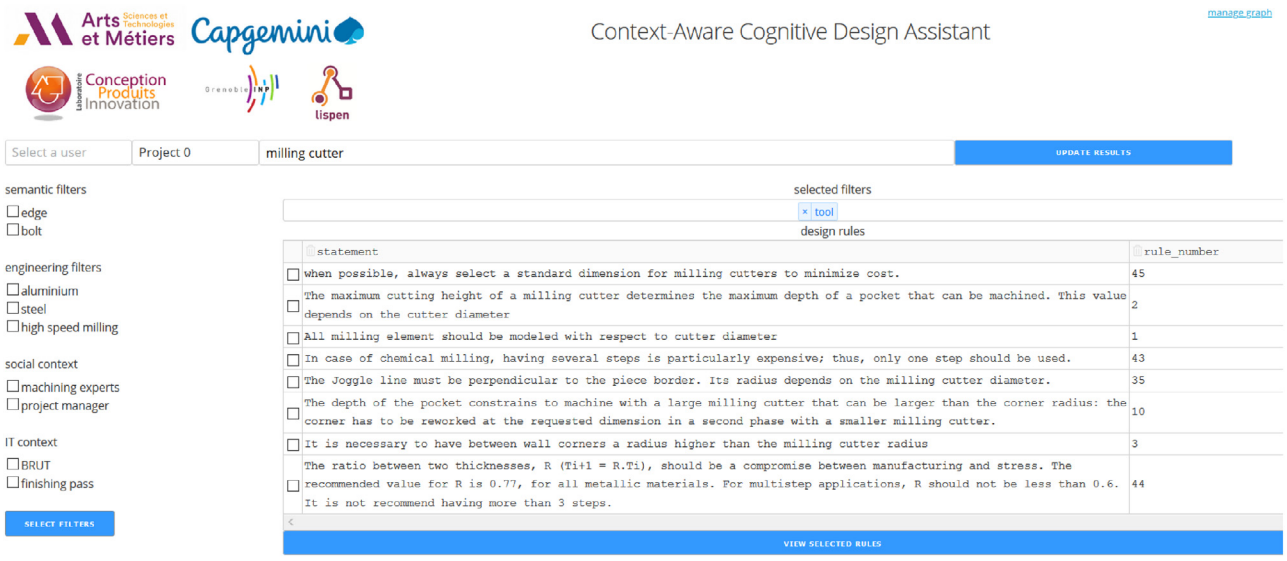


Fig. 8. Design rules resulting from the full-text query "milling cutter" with the dynamic facets on the left. The query results is refined with the facet "tool".

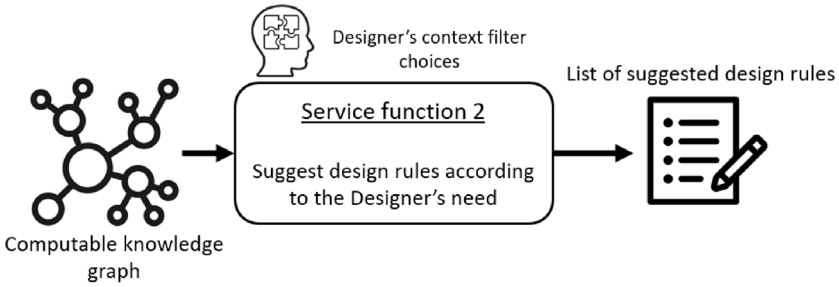


Fig. 9. Schema of service function 2.

intervention, relevant design rules, where relevancy depends on the designer's task (Fig. 9).

■ ***To guide rule compliance in a CAD environment***

After filtering a subset of potentially applicable design rules, the assistant shall enable the designer to associate some formal design rules (e.g. geometrical, dimensional, and topological constraints) with CAD models and make sure they are satisfied (Fig. 10). The association of design rules stored in the design assistant with CAD features is possible using the API of the CAD software. Several approaches are worth considering to present relevant design rules information to the designer. For example, CAD annotations can be used to guide the designer while embedding this information into the CAD model for a possible reuse (Camba et al., 2017).

4. A knowledge graph for a context-aware cognitive design assistant

Based on the services the assistant shall provide to the stakeholders, we will derive the data model of the knowledge graph.

4.1. Modelling method

As presented during the state of the art, we intend to use a combination of two modelling approaches. With the inductive approach, we can deduce graph structures from the system expected functionalities. With the deductive approach, we can deduce graph structures from a coherent description of the context. More details on these approaches are given in the state of the art review.

The previous chapter presents the assistant's service functions. Functionalities and behaviours deduced from these main services can be used as inputs to apply an inductive modelling approach to our knowledge graph. It will ensure that the final knowledge graph is computable and service oriented.

In order to apply a deductive modelling approach, we need to divide the design context into a coherent set of sub-contexts. Table 2 presents the relevancy of all sub-contexts identified during the state of the art review for each service function of the assistant.

We can conclude from Table 2 that five sub-contexts may be relevant for the assistant: user, operational, social, document, and Session. From these sub-contexts and our analysis of the design domain, we deduce the four sub-contexts of our assistant:

The semantic sub-context is inspired from the document sub-context. It models the contextual information extracted from contextual sources in natural language, mainly design rules main statements and domain taxonomies.

The engineering sub-context is also derived from the document sub-context. However, it models a different type of domain specific contextual information. This sub-context models technical design information, build of materials and available manufacturing processes for example.

The social sub-context is built from the user and social sub-contexts found in literature. It models both the user information and his or her social environment.

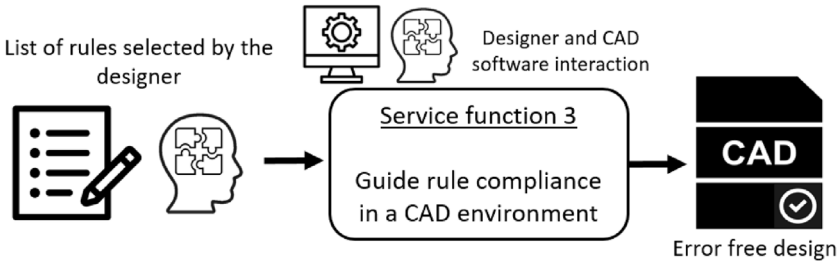


Fig. 10. Schema of service function 3.

Table 2
Relevancy of identified sub-context to the assistant's service functions.

Sub-context	Remarks	Service 1	Service 2	Service 3
User	Contextual information on the user is essential to perform user centric services	X	O	O
Spatio-temporal	We assume that in a design context, the user is always at his/her office	X	X	X
Operational	Understanding the user's goal is essential to build a cognitive assistant	X	O	O
Social	Social interactions are of major interest for recommendation systems	X	O	X
Document	It is an essential sub-context for information retrieval domain. In our domain, a specific sub-context must be dedicated to design rule modeling	O	O	X
Session	The majority of the user's activity during the design process is performed on a digital environment, especially on CAD software.	X	O	O
Environment	We make the hypothesis that the physical environment has no impact on the design process	X	X	X
Device	We assume that hardware differences are not relevant in a design context where all designers are working on a professional computer.	X	X	X

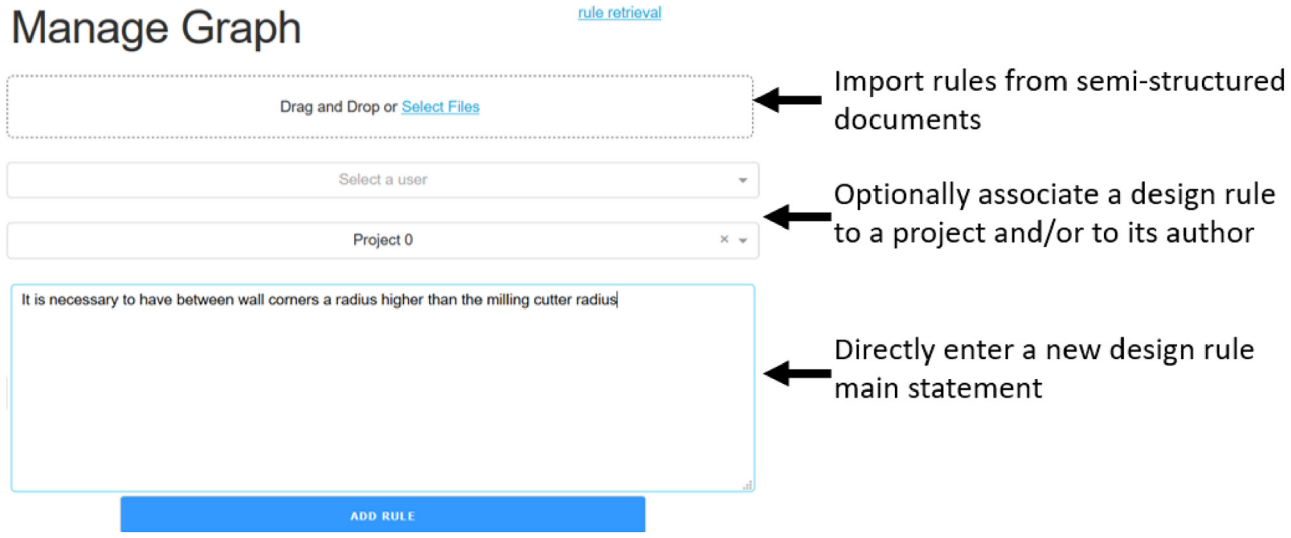


Fig. 11. User interface for design rule addition to the knowledge base.

The IT sub-context is similar to the session sub-context and models the user's digital environment with a domain focus on the CAD software used in the design process.

4.2. Semantic sub-context

Let us consider a design rule from an aerospace design manual: "*It is necessary to have between wall corners a radius higher than the milling cutter radius*". If we want the assistant to facilitate the access to such a design rule, then there is a need to provide a semantic input. Thus, the knowledge engineer enters a design rule statement or uploads a document (Fig. 11) which will be processed to structure the linguistic features in the semantic sub-context.

Fig. 12 illustrates the semantic sub-context, which is a sub-graph of the knowledge graph, collecting linguistic features such as keywords of the design rules. Keywords are words tagged with a part-of-speech corresponding to a noun, a verb, an adjective, or an adverb.

However, a full-text search of a keyword requires an exact matching and supposes that the designer knows what to look for.

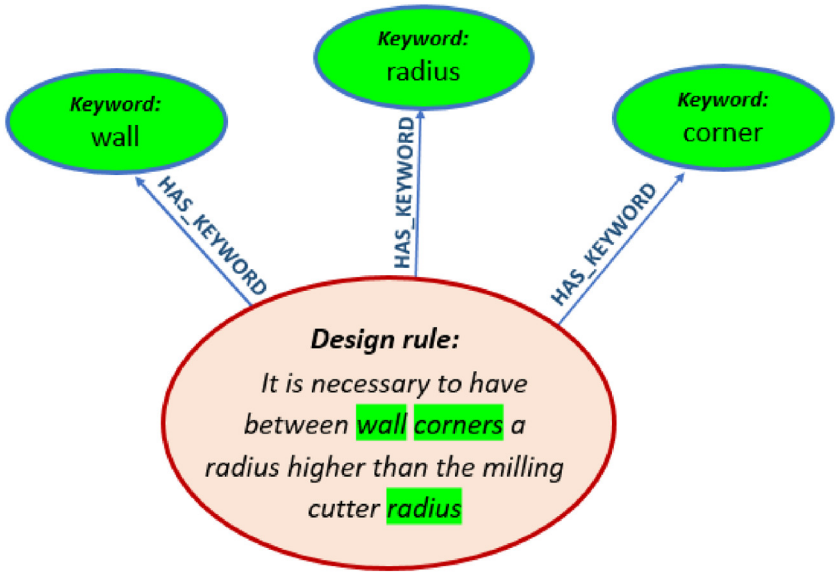


Fig. 12. Example of the design rule linked with 3 keywords.

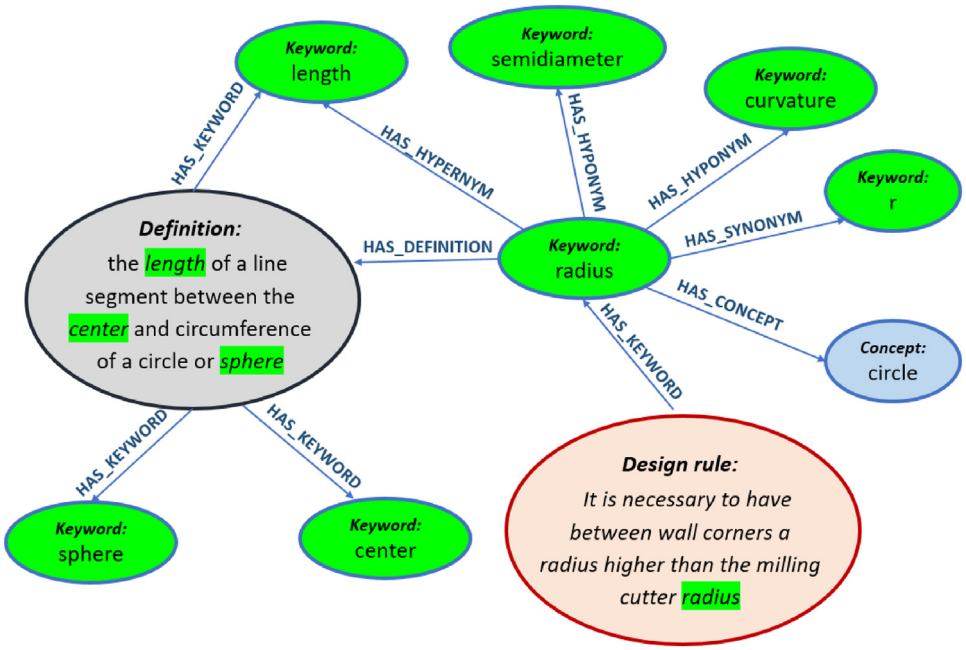


Fig. 13. Keyword "radius" with WordNet and ConceptNet information.

During the design, it is very likely that the designer has, at best, a vague idea of the applicable rules. There is therefore a need to expand queries with relevant keywords to navigate through the knowledge graph. State-of-the-art strategies for the analysis of short textual statements suggest enriching the data with external semantic resources (Shrestha, 2011; Abdalgader, 2014).

To improve the search capability, we enrich the existing semantic sub-context with additional linguistic features such as the definitions of keywords, terms that are lexically related to keywords (synonyms, holonyms, meronyms, hypernyms, derived related terms), and terms that are related to keywords by commonsense knowledge (Fig. 13). The assistant retrieves the lexically related terms from the thesaurus Wordnet (Miller, 1995) before removing the inflected forms of the keywords using the lemmatization capability of the natural language processing library CoreNLP (Manning et al., 2014). The features that stand as commonsense knowledge come from the semantic network ConceptNet (Speer et al., 2016).

These thesauruses contain commensense knowledge that may not be adapted to our domain. The assistant shall select the correct knowledge to add in the knowledge graph. For example, in ConceptNet, the keyword "radius" is linked with the concepts "circle" and "bone". ConceptNet links "circle" with the context "geometry" and "bone" with the context "animal". The assistant can therefore select the concepts related to the scientific or design contexts.

The assistant extracts keywords from the design rules, the titles and the chapters. Fig. 14 represents the model of the semantic sub-context in the graph-oriented database Neo4j.

Table 3 contains a detailed definition of the nodes, relationships, and the associated properties that make up the semantic sub-context. Each entity and relationship owns a property corresponding to a unique identifier that is not mentioned in the table.

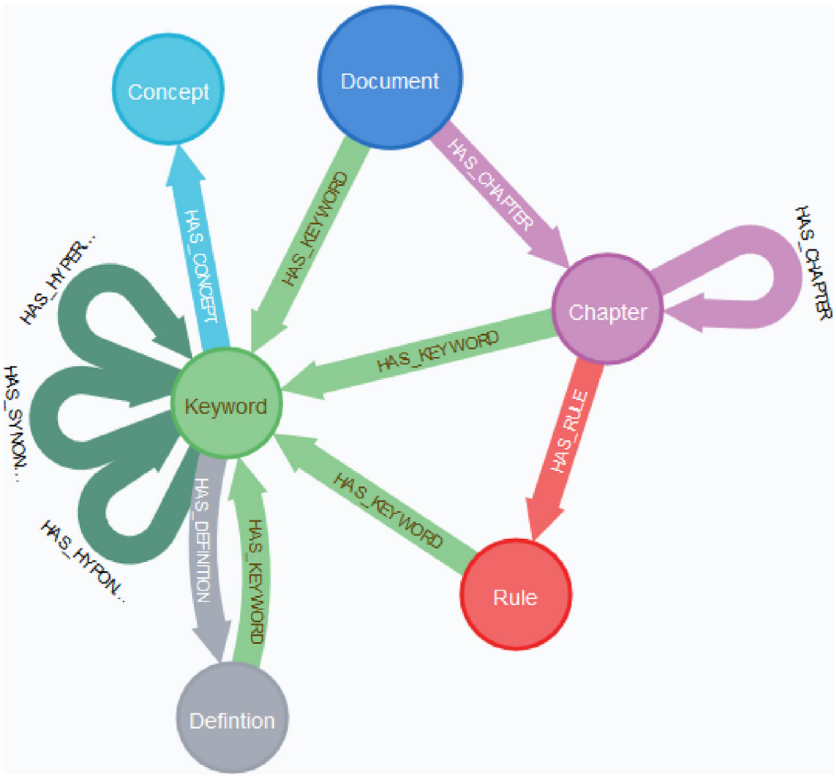


Fig. 14. Semantic sub-context of the knowledge graph.

Table 3
Nodes of the Semantic sub-context.

Concept *<Concept:String>: Name of the concept*	Commonsense concepts extracted from ConceptNet
Document *<Title:String>: Title of the document* (e.g. Design for metallic parts)	Any (semi-)structured or unstructured storage medium that contains design rules.
<Update:Date>: Date of the last time the document was modified	
Chapter *<Tilte:String>: Title of the chapter*	Structure elements of design rules manuals
<Page:Integer>: Page number of the chapter in the source document	
<Chapter_num:String>: Chapter number in the document structure (e.g. Chapter 1.2)	
<Source_id:UUID>: Id of the source document the chapter has been extracted from.	
Rule *<Main_Statement:String>: Natural language statement of the design rule*	Design rules
<Schema:String>: Design rules can be associated with an image or a schema. The Schema string points to the design rule's schema in the assistant resources.	
<Formula:String>: Design rules can be associated with a formula (e.g. "R = (milling cutter radius) + 1 mm")	
<Last modification:date>: Date of the last modification of the design rule	
Keyword *<Lemma:String>: Lemma of the keyword*	Keywords extracted from source documents
<Stem:String>: Stem of the keyword	
<Tag:String>: Part-of-speech	
Definition *<Statement:String>: Statement of the keyword definition in natural language*	**Keyword definition extracted from WordNet or from glossaries**
<Score:Float>: Similarity score of the definition computed during NLP process. It represents how strongly the definition of a keyword relates to the semantic sub-context.	

4.3. Engineering sub-context

The linguistic features stored in the semantic sub-context facilitate the retrieval of design rules, but product design relies more on technical words than common knowledge. Thus, we propose to enrich the knowledge graph with a second sub-context that stores engineering entities and relationships. So far, the engineering entities are technical words such as the name of a material,

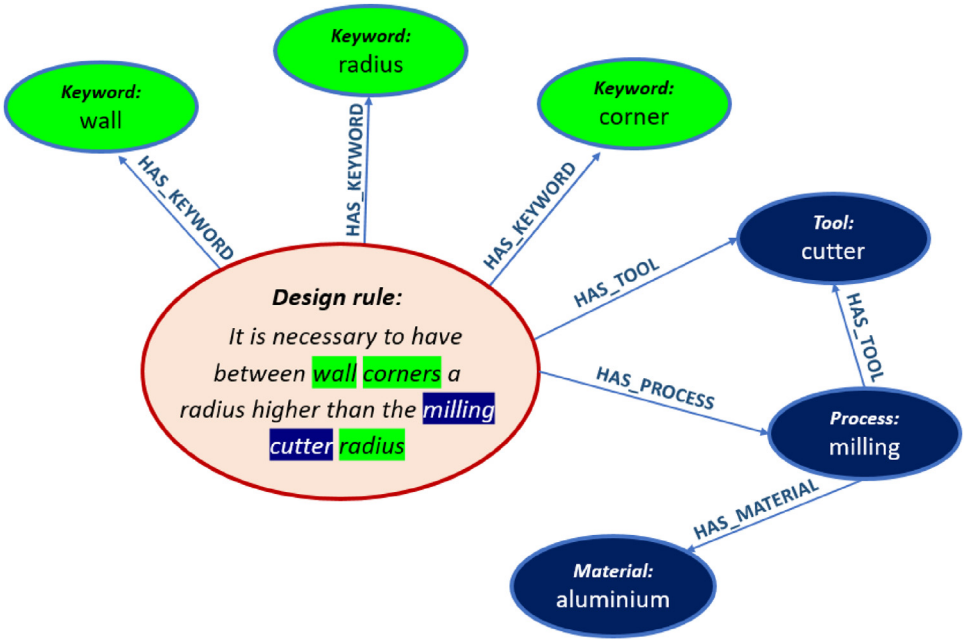


Fig. 15. Example rule with semantic and engineering sub-contexts.

a manufacturing process, and the domain of expertise the design rule belongs to (e.g. mechanics, electronics, etc.). Fig. 15 shows that a designer who is looking for applicable design rules related to the machining process "milling" will obtain more results than a query on the semantic sub-context.

Entities of the engineering sub-context (Fig. 16) are keywords with a specific tag. So far, the tagging process relies on a rule-based classifier feed by multiple internal sources of knowledge including glossaries, dictionaries, taxonomies, and bill of materials. The expertise domain can also be inferred using a supervised machine learning-based classifier (Pinquié et al., 2018) (Table 4).

4.4. Social sub-context

Accessing relevant design rules is possible through linguistic, commonsense, and engineering features, but one key characteristic of context-aware recommendation systems is the social dimension. The social sub-context is present in all contextual models presented in the literature review. In information retrieval, it is used to perform recommendations based on users' preferences. Thus, the context-aware cognitive design assistant could sense the activity of designers and recommend design rules based on social activities and affinities. For instance, if a designer with a similar profile as mine systematically searches, consults and implements design rules that I have never came across, then it is very likely that they should be recommended to me (Fig. 17).

The social sub-context (Fig. 18) is in two parts. The first part is static and captured after each designer has filled a user profile. This includes the entities ":Role", ":Expertise", ":Skill", ":Team", ":Company". The ":FRIEND_OF" relationships come from a list of contacts of the social platform used by the company (e.g. a PLM software, or communication platforms such as Teams, Slack, Skype Enterprise, etc.). The second part is dynamic and captures the activity of designers. So far, the assistant memorises the full-text queries and the design rules a designer has consulted (Table 5).

4.5. IT sub-context

Many rules constrain the computer-aided modelling practices and engineering features occurs in CAD environment. For instance,

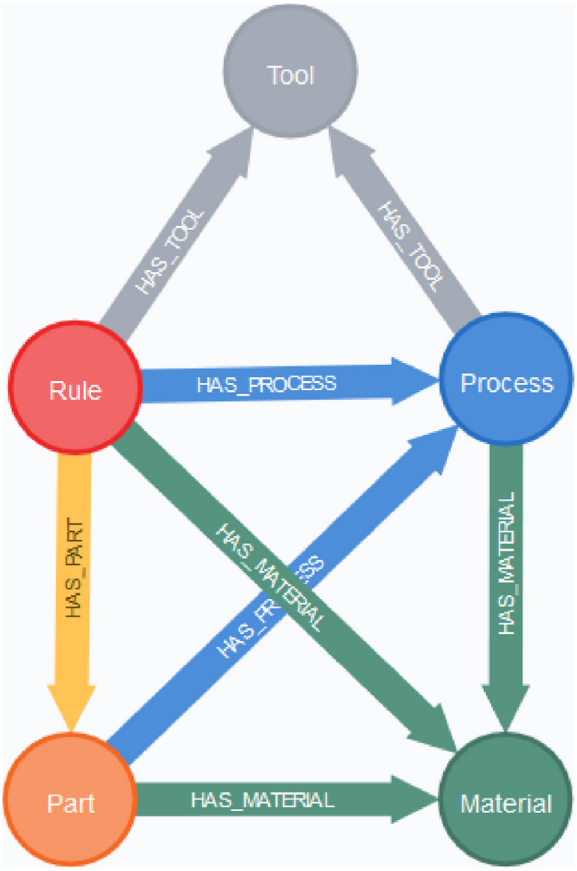


Fig. 16. Engineering sub-context of the knowledge graph.

when a designer is modelling a feature named "fillet", the assistant shall use its context-aware capability to recommend applicable design rules. The dynamic capture of the current working IT situation corresponds to the IT sub-context. Entities of the IT sub-context include the software (e.g. CATIA), the workbench (e.g. Part Design),

Table 4
Nodes of the Engineering sub-context.

Tool *<Name:String>: Name of the tool*	Tool used in some manufacturing process (e.g cutting tools for milling)
<Dimensions:String>: String describing the tool dimension.	
<Tool_type:String>: Type of tool	
Part *<Name:String>: Name of the part.*	Part of an engineering product (e.g. aircraft wings)
<Description:String>: Description of the design part.	
Process *<Name:String>: Name of the process*	Manufacturing process that the company can use to create their design (e.g. milling)
<Documentation:String>: Link to retrieve technical information about the process in the assistant resources	
Rule C.f. Table 3	Design rules
Material *<Name:String>: Name of the material* (e.g. aluminium)	A material available to the company to create their designs (e.g. Titanium)
<Type:String>: Type of material (e.g. metal)	
<Characteristics:String>: Mechanical characteristics of the material	

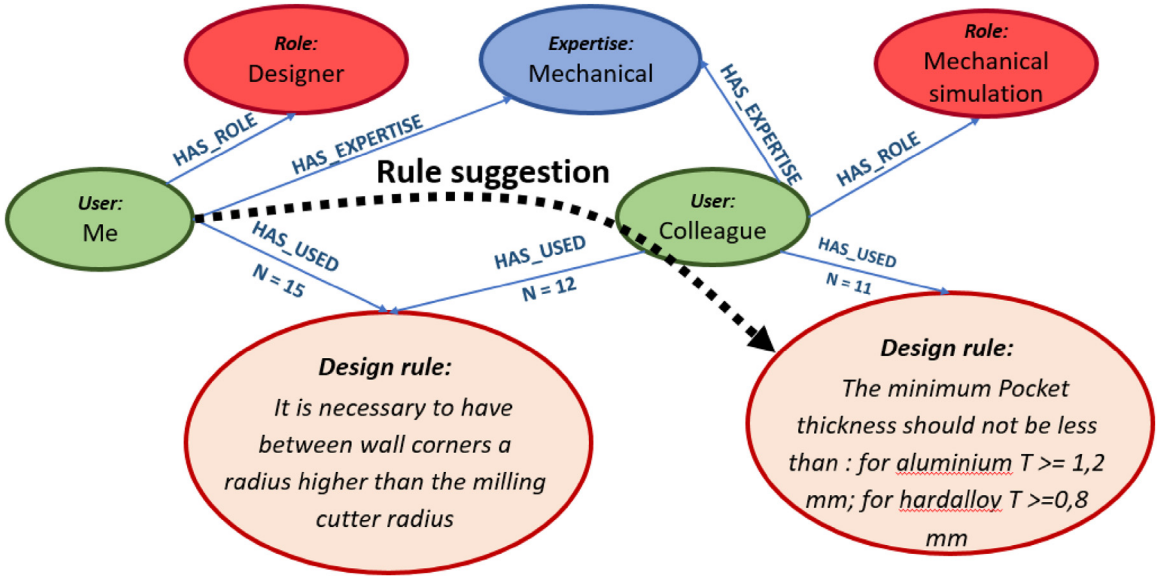


Fig. 17. Path between two design rules in the social sub-context.

and the feature (e.g. Extrusion), the data being edited (e.g. Beam), and the name of the PDM project that stores the data. The self-relationship [:LINK_TO] represents a link between data in the PDM software (e.g. link between a CAD model, its FEA mesh, and its 2D drawing). Fig. 19 illustrates how the example design rule relates to the IT sub-context.

The link between the IT sub-context and design rules can be more complex than this example. We can see that the IT sub-context is not directly connected to design rules but uses domain specific contextual information from semantic and engineering sub-contexts as intermediary. Therefore, the capability of the assistant to deduce design rule recommendations from the IT sub-context depends heavily on the quality of the domain specific knowledge available to the assistant. For example, an aeronautical company will have to implement specific semantic and engineering information about aircraft in order to interpret correctly the IT sub-context of an Aircraft part modelling. Fig. 20 represents the knowledge graph of the IT sub-context (Table 6).

4.6. Consolidation of sub-contexts

Graph pattern queries on a sub-context help to answer questions such as "*What design rules apply for material < Aluminum>?*", "*What design rules apply for process < Milling>?*", "*What design rules apply for part < Rib>?*"... However, results will be limited to a specific sub-context. What happens if we want to make a query involving several sub-contexts? For example, to answer the question "*Which colleague is considered as an expert for the part I am designing?*", there is a need to consolidate the social sub-context and the engineering sub-context using a relationship named [:INCLUDES] (Fig. 21). To answer this question, the assistant traverses the graph with the path: "(:Part) <- [:INCLUDES] - (:Expertise) <- [:HAS_EXPERTISE] - (:User)".

The knowledge graph resulting from the literature review and the functional analysis might be incomplete and will be continuously refined according to the experiments we will carry out with the context-aware cognitive design assistant prototype.

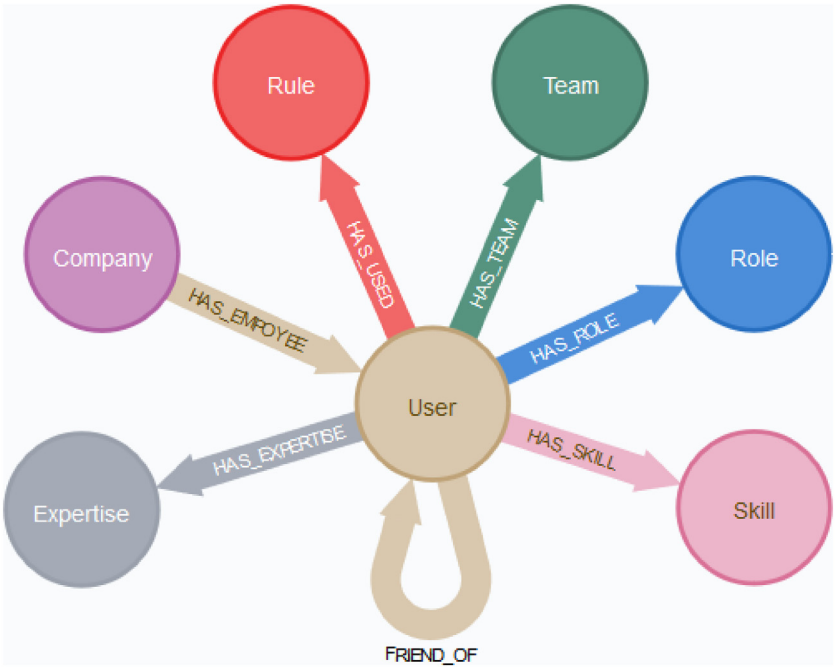


Fig. 18. Knowledge graph of the social sub-context.

Table 5
Nodes of the Social sub-context.

User *<Name:String>: Name of the user*	User of the assistant
<Seniority:Integer>: Professional experience of the user in years.	
<Mail:String>: Professional mail address of the user	
Team *<Name:String>: Name of the team*	Group of users working on a common goal
<Team_goal:String>: Goal of the team described in a small string. (e.g. fuselage design)	
Company *<Name:String>: Name of the company*	Company
<Activity:String>: Company's domain of activity (e.g. aircraft industry)	
Role *<Name:String>: Name of the role*	Role of an employee in a company (e.g. surface designer)
<Description:String>: Description of the role (e.g. "High quality final surface modelling of the wing")	
Skill *<Name:String>: Name of the skill*	Skill (e.g. CAD surface design)
<Description:String>: Description of the skill (e.g. "Technical A5 surface design with specialized CAD software")	
Expertise *<Name:String>: Name of the expertise*	Subject or topic in which a user is considered expert
<Description:String>: Description of the expertise	

5. CADCA preliminary validation

5.1. Conditions

To demonstrate the feasibility of our approach, we developed a demonstrator implementing the design rules recommendation engine (service function number 2) of the Context-Aware Cognitive Design Assistant. In this experiment, only the semantic sub-context is used to perform recommendations. Participants are asked to realize several design operations with the support of a set of 45 design rules that make up the "Design for Milling" section of an aeronautical design manual.

We divided designers into two groups: a control group and a test group. As the Context-Aware-Cognitive Design Assistant aims at replacing unstructured design rules documents, participants of the control group can access the design rules set via an unstructured PDF document. Participants of the test group access the same rules using the demonstrator of the Context-Aware Cognitive Design Assistant. Fig. 8 illustrates the user interface of the demonstrator. Participants of the test group are able to type queries into the search

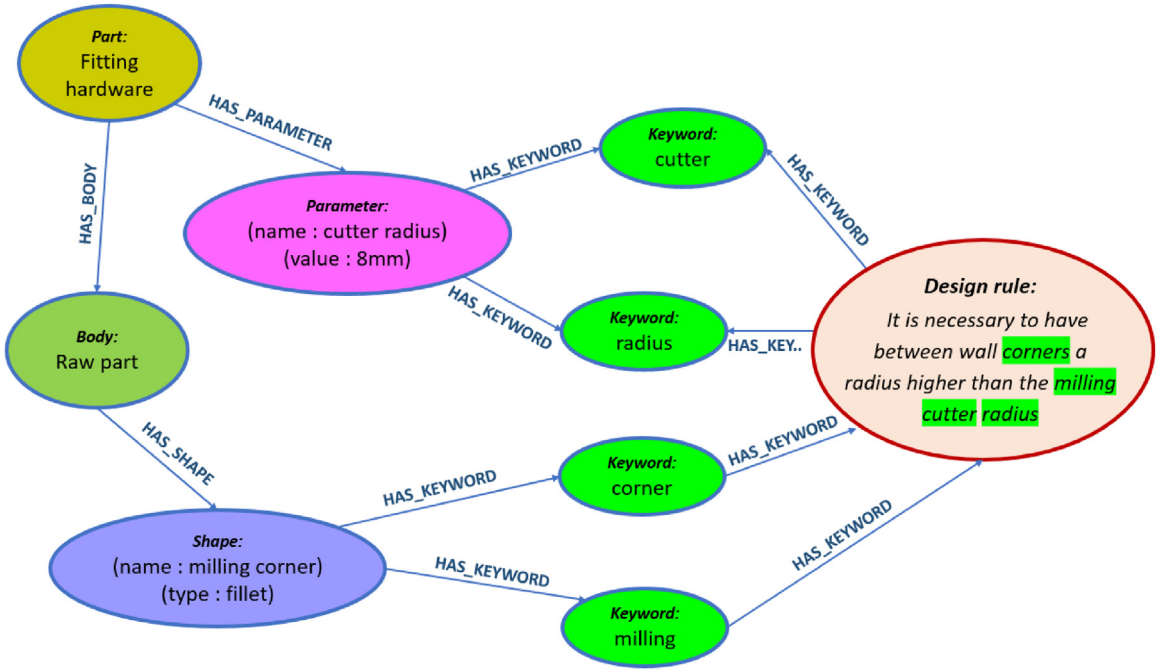


Fig. 19. Path between the IT sub-context and a design rule.

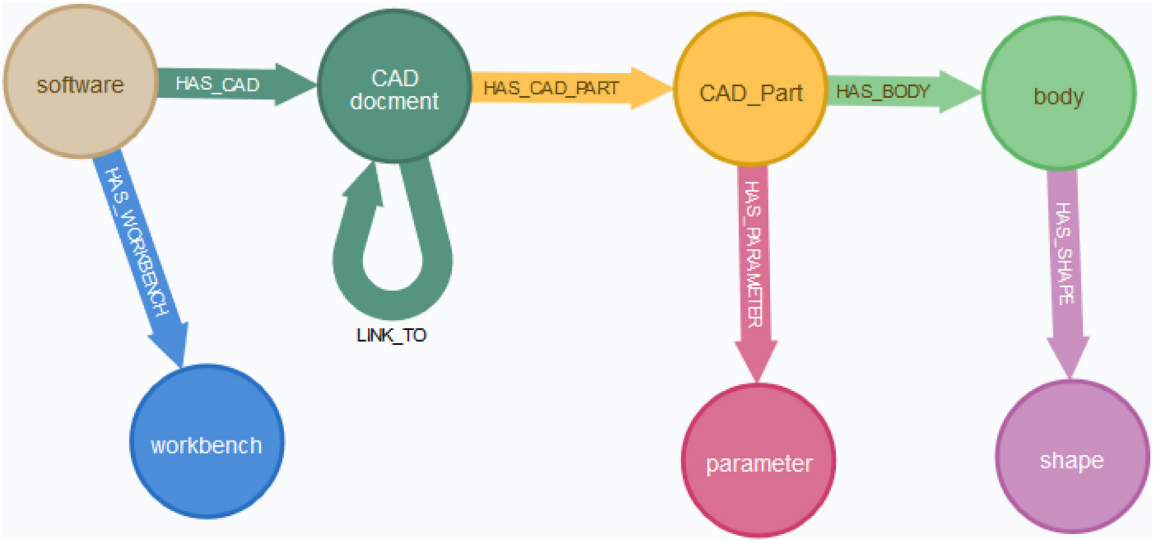


Fig. 20. Knowledge graph of the IT sub-context.

bar and to explore related design rules using the recommended dynamic semantic filters that stand as query expansions.

5.2. Task

Participants have the role of a designer in a manufacturing company. They have to realize a design task while respecting applicable design rules of the set. Each participant is given an initial CAD part with the task to realize three pockets. The sketch of the first pocket is imposed to every participant. Then, they must realize the two pockets inside the first pocket, as presented in Figs. 22 and 23. Milling is the only manufacturing process authorized to realize these operations.

The design goal is to minimize the mass of the part and the design space is constrained by the design rules. Among the 45 design rules provided, only 9 are applicable. Thus, designers have to retrieve applicable design rules to provide an acceptable design.

5.3. Participants

Five participants (all male) with design skills and an average age of 30.6 years have been tested. Three of them declared a novice level with the CAD environment used for the test. All others declared an intermediate level. None of them had any knowledge on the design rules before the test. Three of them declared that they never used any design rules in their design experiences. The others only experienced designing with a small set of a dozen rules.

We consider that participants equipped with design skills but very small knowledge about design rules stand as inexperienced

Table 6
Nodes of the IT sub-context.

Software *<Name:String>: Name of the software*	CAD software used by the designer (e.g. CATIA)
<Functionnality:String>: Description of the software (e.g. general CAD design)	
CAD document *<Name:String>: Name of the document*	Document containing CAD data
<Last_modification:Date>: Date of the document last modification.	
CAD Part *<Name:String>: Name of the Part*	CAD model of a specific part
<Characteristics:List>: Technical characteristics of the CAD Part (e.g. Volume, Mass, etc.)	
< Last_modification:Date>: Date of the CAD Part last modification	
Body *<Name:String>: Name of the Body*	CAD element containing geometrical elements
< Last_modification:Date>: Date of the body last modification	
Shape *<Name:String>: Name of the shape*	Geometrical element
<Type:String>: Type of shape	
<Feature:String>: CAD operation at the shape's origin	
Parameter *<Name:String>: Name of the parameter*	Named variable. Its value influences the CAD design
<Value:Numerical>: Numerical value of the parameter	
Workbench *<Name:String>: Name of the Workbench*	Specific workbench of a CAD tool (e.g. Part Design in CATIA)
<List_of_tools:String>: List of CAD operation available to the designer in this workbench	

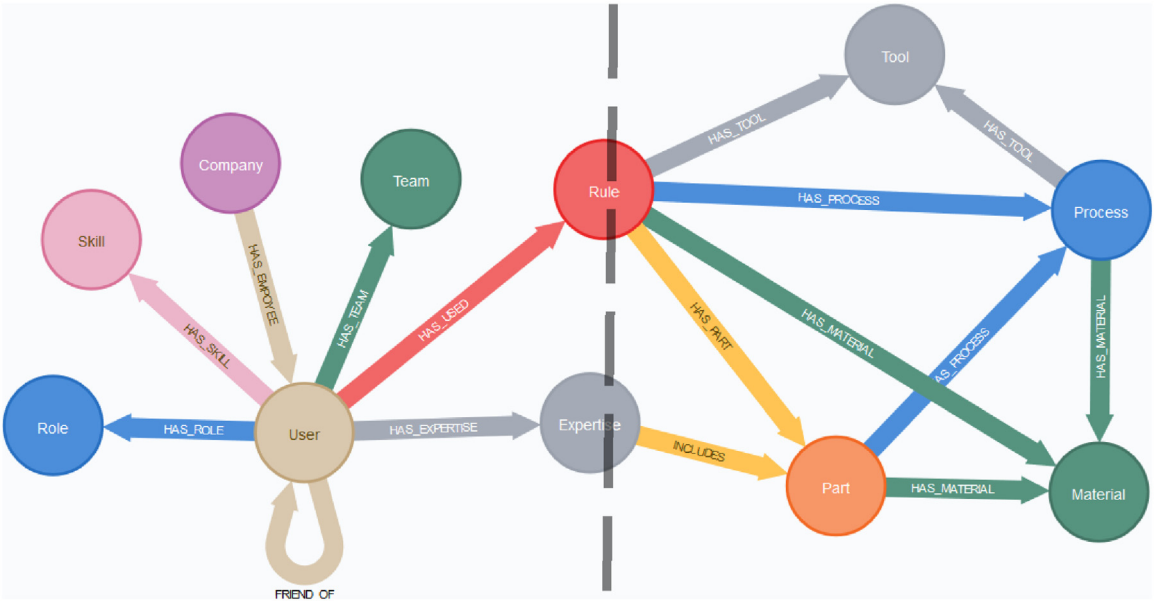


Fig. 21. Example of knowledge graph consolidation.

designers in a manufacturing company. 3 participants were part of the test group and 2 of the control group.

5.4. Results analysis and discussion

During this experiment, participants had to retrieve applicable design rules while modelling their CAD part. We measured the precision and recall for the design rules retrieval activity. Design performance is measured with the final CAD part of each participant. We take into consideration the number of design errors in the final part (one error is counted for each violated design rule) as well as the total volume of the part. We measured the total time of the experiment as well as the time dedicated to CAD mod-

Table 7
Design rules retrieval performances.

	precision	recall
Control group	0.36	0.28
Test group	0.61	0.3

Table 8
Design rules retrieval performances per participant.

Participant	1	2	3	4	5
Level of CAD expertise	Novice	Novice	Novice	Intermediate	Intermediate
Group	Control	Control	Test	Test	Test
Size of the design rule set	45	45	45	45	45
Number of applicable design rules	9	9	9	9	9
True positive	2	3	2	4	2
False positive	5	4	1	2	2
True negative	31	32	35	34	34
False negative	7	6	7	5	7
Precision	0.29	0.43	0.67	0.67	0.5
Recall	0.22	0.33	0.22	0.44	0.22

Table 9
Average design performances.

	Number of design errors	Final volume of the CAD model in cm^3
Control group	3.5	194.3
Test group	3.7	174.7

Table 10
Time measures of the experiment.

	Experimentation time	Percentage spent of CAD modeling activity
Control group	1h26min	37.2 %
Test group	1h17min	55.6 %

elling. Tables 7–10 present the experiment's results. These tables are associated with small paragraphs of our results interpretation.

Results show that precision is significantly higher for the test group while recall is similar. Participants in the test group tend to select fewer design rules in total but with less false positive.

The small recall number of all participants may be explained by the difficulty of the task. None of our participants was familiar with the technical English used in design rules. Moreover, the test places participants in a situation where they have to discover, analyse and apply a completely unknown set of design rules in one go. The fact that some of applicable design rules were not retrieved by any participant may indicate a limitation in our test scenario.

Design performance results do not indicate any significant difference in the number of design errors between our two groups. The average volume of the final part is 19.6 cm^3 *smaller in the test group. It represents a* 10 *% volume reduction. This reduction may be attributed to a better understanding and confidence over design rules limitations in the test group. One possible bias is the higher number of intermediate CAD designers in the test group.*

In general, designing a part at manufacturability limits only is a difficult task that sometimes go against participants' habits. Results on design performances are encouraging and must be confirmed and improved in future works.

hile the two groups spend approximately the same time on the experiment, results show that participants of the test group spend more time designing. As one of the goals of CACDA is to improve the retrieval of design rules in order to minimize its cognitive weight for the designer, these results are encouraging.

Several limits need to be considered in the final analysis of these results:

- The CACDA demonstrator used in the experiment only features the semantic sub-context. Therefore, results shall be further improved with the implementation of the entire knowledge graph.
- The set of design rules include 45 design rules whereas industrial design manuals feature hundreds of them. Moreover, the manufacturing process of the test part was imposed and only design rules related to that specific process were selected. A more complex part with more design rules would be a better representation of the reality.
- Participants were asked, in a small period, to find and understand design rules by their own among a set of design rules that was new for them. In industry, designers would potentially have more time to get familiar with the dataset.
- The small scale of this experiment limits the soundness of these results, but it establishes a quantitative reference that paves the way for future improvements. Further experiments with more participants need to be realized to confirm the interest and feasibility of the multiple design sub-contexts.

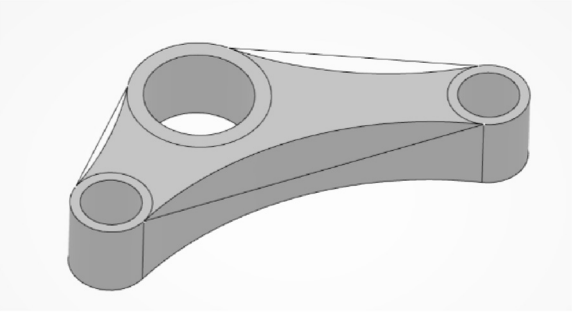


Fig. 22. Final part of a participant.

In general, results indicate that the use of our demonstrator for design rules retrieval generates several improvements in comparison with unstructured documentation. The test group performed better in terms of precision in design rules retrieval, volume reduction in final part and time efficiency. It sets a first reference for comparisons and improvements in future works.

6. Conclusion and future works

The storage of design rules in large and unstructured documents is no longer an efficient solution for design rules retrieval

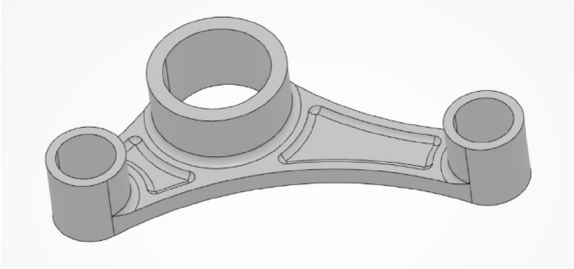


Fig. 23. Initial part of the experiment.

and application. Thus, we propose a Context-Aware Cognitive Design Assistant to simplify design rules application and improve computer-aided design efficiency. The design assistant shall provide three mains services. The first one is to structure design rules and contextual information into a computable knowledge graph. The second one is to retrieve applicable design rules. The third one is to support the compliance of design rules.

The first part of the paper details the main stakeholders of design assistants and the services it shall provide to them. This operational end-user analysis supports service-oriented modelling strategies to build a computable knowledge graph for the assistant. The second part of the paper proposes a knowledge graph that underlies the design assistant. This knowledge graph structures design rule information and design contextual knowledge in a computable format.

Finally, we present a preliminary validation experiment for a proof of concept of our design assistant. Results show that in a design context our cognitive assistant enables the participants to select applicable design rules with more precision while spending more time on the CAD modelling activity. In fact, participants using an unstructured design rules document achieved a precision of 0.36, and spend 37.2 % of their time on CAD modelling activity, whereas those using our assistant show a precision of 0.61 and spend 55.6 % of their time on CAD modelling.

Future works aim at improving both the demonstrator of the Context-Aware Cognitive Design Assistant and our test protocol to confirm the preliminary results presented in this paper. These future tests will include all sub-contexts of the knowledge graph, a more complex design part, and more participants to support our preliminary results. These future tasks will also enable us to refine the knowledge graph data-model.

CRediT authorship contribution statement

Armand Huet: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Romain Pinquié:** Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Philippe Véron:** Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Antoine Mallet:** Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Frédéric Segonds:** Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing, Visualization, Supervision.

Declaration of Competing Interest

The authors report no declarations of interest.

References

Abdalgader, K., 2014]. Word sense identification improves the measurement of short-text similarity. Proc. Int. Conf. Comput. Technol. Inf. Manag. (ICCTIM 2014), 233–243, http://dx.doi.org/10.1007/978-3-642-17432-2_44.

Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P., 1999]. Towards a Better Understanding of Context and Context-Awareness., pp. 304–307.

Alexopoulos, K., Makris, S., Xanthakis, V., Sipsas, K., Chryssolouris, G., 2016]. A concept for context-aware computing in manufacturing: the white goods case. Int. J. Comput. Integr. Manuf. 29 (August 8), 839–849, http://dx.doi.org/10.1080/0951192X.2015.1130257.

Angles, R., 2018]. The property graph database model. CEUR Workshop Proc., vol. 2100, no. Section 2.

Baldauf, M., Dustdar, S., Rosenberg, F., 2007]. A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2 (4), 263–277, http://dx.doi.org/10.1504/IJAHUC.2007.014070.

Ben-Yitzhak, O., et al., 2008]. Beyond basic faceted search. WSDM'08 - Proc. 2008 Int. Conf. Web Search Data Min., 33–43, http://dx.doi.org/10.1145/1341531.1341539.

Boy, G.A., Boy, G., Gruber, T.R., 1990]. Intelligent assistant systems : support for integrated human-machine systems knowledge systems. Symp. Knowledge-Based Human-Computer Commun., March, 1–6.

Camba, J.D., Contero, M., Company, P., Pérez, D., 2017]. On the integration of model-based feature information in Product Lifecycle Management systems. Int. J. Inf. Manage. 37 (6), 611–621, http://dx.doi.org/10.1016/j.ijinfomgt.2017.06.002.

Cassens, J., Kofod-Petersen, A., 2006]. Using activity theory to model context awareness: a qualitative case study. FLAIRS Conf., 619–624.

Company, P., Contero, M., Otey, J., Plumed, R., 2015]. Approach for developing coordinated rubrics to convey quality criteria in MCAD training. CAD Comput. Aided Des. 63, 101–117, http://dx.doi.org/10.1016/j.cad.2014.10.001.

Dey, A.K., 2001]. Understanding and using context. Pers. Ubiquitous Comput. 5 (1), 4–7, http://dx.doi.org/10.1007/s007790170019.

Dhuieb, M.A., Laroche, F., Belkadi, F., Bernard, A., 2015]. Activity theory based context model: application for enterprise intelligent assistant systems. IFACPapersOnLine 28 (3), 834–839, http://dx.doi.org/10.1016/j.ifacol.2015.06.187.

Dhuieb, M.A., Laroche, F., Bernard, A., 2016]. Context-awareness: a key enabler for ubiquitous access to manufacturing knowledge. Procedia CIRP 41, 484–489, http://dx.doi.org/10.1016/j.procir.2015.12.027.

Ehrlinger, L., Wöß, W., 2016]. Towards a definition of knowledge graphs. Semant. (Posters, Demos, SuCCESS), vol. 48., pp. 1–4.

Faloutsos, C., Oard, D.W., 1998]. A Survey of Information Retrieval and Filtering Methods, vol. 8958546., pp. 1–24.

Fortineau, V., Fiorentini, X., Paviot, T., Louis-Sidney, L., Lamouri, S., 2014]. Expressing formal rules within ontology-based models using SWRL: an application to the nuclear industry. Int. J. Prod. Lifecycle Manag. 7 (1), 75–93, http://dx.doi.org/10.1504/IJPLM.2014.065458.

Fu, K.K., Yang, M.C., Wood, K.L., 2016]. Design principles: literature review, analysis, and future directions. J. Mech. Des. Trans. ASME 138 (10), 1–13, http://dx.doi.org/10.1115/1.4034105.

L. E. R. García, A. Garcia, and J. Bateman, "An Ontology-Based Feature Recognition and Design Rules Checker for Engineering".

Gero, J.S., Gero, J.S., Lou Maher, M., Kazakov, V., 1999]. Adapting evolutionary computing for exploration in creative designing. Comput. Model. Creat. Des. IV, 175–186.

Gruszka, A., Nęcka, E., 2017]. Limitations of working memory capacity: The cognitive and social consequences. Eur. Manag. J. 35 (6), 776–784, http://dx.doi.org/10.1016/j.emj.2017.07.001.

Gu, T., Wang, X.H., Pung, H., Zhang, D.Q., 2004]. An ontology based context model in intelligent environments. Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference.

Hariya, M., Nonaka, N., Shimizu, Y., Konishi, K., Iwasaka, T., 2010]. Technique for checking design rules for three-dimensional CAD data. Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2010, vol. 1., pp. 296–300, http://dx.doi.org/10.1109/ICCSIT.2010.5565010.

Hasanov, A., Laine, T.H., Chung, T.S., 2019]. A survey of adaptive context-aware learning environments. J. Ambient Intell. Smart Environ. 11 (5), 403–428, http://dx.doi.org/10.3233/AIS-190534.

Huang, B., Xu, C., Huang, R., Zhang, S., 2015]. An automatic 3D CAD model errors detection method of aircraft structural part for NC machining. J. Comput. Des. Eng. 2 (4), 253–260, http://dx.doi.org/10.1016/j.jcde.2015.06.008.

Huet, A., Pinquié, R., Véron, P., Segonds, F., Fau, V., 2020]. Knowledge graph of design rules for a context-aware cognitive design assistant. Springer, Cham In: Product Lifecycle Management International Conference, 594., http://dx.doi.org/10.1007/978-3-030-62807-9_27.

Ingwersen, P., 2007]. Context in information interaction – revisited 2006. ProLISSA 2006 Proc. Fourth Bienn. DISSAnet Conf. Farm Inn, Pretoria, 2-3 November, 13–23.

Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S., 2020]. A survey on knowledge graphs: representation. Acquisit. Appl., 1–25.

Johansson, J., Contero, M., Company, P., Elgh, F., 2018]. Supporting connectivism in knowledge based engineering with graph theory, filtering techniques and model quality assurance. Adv. Eng. Informatics 38 (July), 252–263, http://dx.doi.org/10.1016/j.aei.2018.07.005.

Kang, S.K., et al., 2015]. Extraction of manufacturing rules from unstructured text using a semantic framework. Proceedings of the ASME Design Engineering Technical Conference, vol. 1B, 10.1115/DETC2015-47556.

Kofod-Petersen, A., Aamodt, A., 2003]. Case-based situation assessment in a mobile context-aware system. Artif. Intell. Mob. Syst., 41–49.
Li, Z., Raskin, V., Ramani, K., 2008]. Developing engineering ontology for information retrieval. J. Comput. Inf. Sci. Eng. 8 (1), http://dx.doi.org/10.1115/1.2830851.
Mackenzie, C.A., Gero, J.S., 1987]. Learning design rules from decisions and performances. Ai Edam 2 (1), 2–10, http://dx.doi.org/10.1016/0954-1810(87)90065-3.
Madni, A.M., 2020]. Exploiting augmented intelligence in systems engineering and engineered systems. Insight 23 (1), 31–36, http://dx.doi.org/10.1002/inst.12282.
Manning, C.D., Bauer, J., Finkel, J., Bethard, S.J., Aclweb.Org 2014]. The Stanford CoreNLP Natural Language Processing Toolkit., pp. 55–60.
Marcu, Dorin, Boicu, Mihai, Tecuci, Gheorghe, Schum, David A., Cambridge 2016]. Knowledge Engineering: Building Cognitive Assistants for Evidence-Based Reasoning.
Mattos, T.D.C., Santoro, F.M., Revoredo, K., Nunes, V.T., 2014]. A formal representation for context-aware business processes. Comput. Ind. 65 (8), 1193–1214, http://dx.doi.org/10.1016/j.compind.2014.07.005.
Merrouni, Z.A., Frikh, B., Ouhbi, B., 2019]. Toward contextual information retrieval: a review and trends. Procedia Comput. Sci. 148, 191–200, http://dx.doi.org/10.1016/j.procs.2019.01.036.
Miller, G.A., 1995]. WordNet: a lexical database for english. Commun. ACM 38 (11), 39–41, http://dx.doi.org/10.1145/219717.219748.
Moitra, A., Palla, R., 2016]. Automated capture and execution of manufacturability rules using inductive logic programming. Proc. 30th Conf. Artif. Intell. (AAAI 2016), 4028–4034.
Pinquié, R., Véron, P., Segonds, F., Croué, N., 2018]. A requirement mining framework to support complex sub-systems suppliers. Procedia CIRP 70, 410–415, http://dx.doi.org/10.1016/j.procir.2018.03.228.
Pinquié, R., Véron, P., Segonds, F., Zynda, T., 2020]. A Property Graph Data Model for a Context-Aware Design Assistant.
Rangarajan, A., Radhakrishnan, P., Moitra, A., Crapo, A., Robinson, D., 2013]. Manufacturability analysis and design feedback system developed using semantic framework. Proc. ASME Des. Eng. Tech. Conf. 4, 1–11, http://dx.doi.org/10.1115/DETC2013-12028.
Rouse, W.B., 2020]. AI as systems engineering augmented intelligence for systems engineers. Insight 23 (1), 52–54, http://dx.doi.org/10.1002/inst.12286.
Rowson, H., Bricogne, M., Roucoules, L., Durupt, A., Eynard, B., 2018]. Knowledge capture and reuse through expert' s activity monitoring in engineering design. In: PLM Conf., Torino, Italy, pp. 621–630.
Roy, S.B., Wang, H., Nambiar, U., Das, G., Mohania, M., 2009]. DynaCet: building dynamic faceted search systems over databases. Proc. - Int. Conf. Data Eng., 1463–1466, http://dx.doi.org/10.1109/ICDE.2009.117.
Sanya, I.O., Shehab, E.M., 2014]. An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry. Int. J. Prod. Res. 52 (20), 6192–6215, http://dx.doi.org/10.1080/00207543.2014.919422.
Shi, P., Chen, F., Han, L., Childs, J., 2017]. A data-driven text mining and semantic network analysis for design information retrieval. J. Mech. Des. 139 (11), http://dx.doi.org/10.1115/1.4037649.
Shrestha, P., 2011]. Corpus-based methods for short text similarity. Rencontre des Étudiants Chercheurs en Informatique pour le Traitement automatique des Langues.
Skarka, W., 2007]. Application of MOKA methodology in generative model creation using CATIA. Eng. Appl. Artif. Intell. 20 (5), 677–690, http://dx.doi.org/10.1016/j.engappai.2006.11.019.
Speer, R., Chin, J., Havasi, C., no. Singh 2002 2016]. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge., pp. 4444–4451.
Strobin, L., Niewiadomski, A., 2014]. Recommendations and object discovery in graph databases using path semantic analysis. International Conference on Artificial Intelligence and Soft Computing, 793–804, http://dx.doi.org/10.1007/978-3-319-07173-2_68.
Tamine-Lechani, L., Boughanem, M., Daoud, M., 2010]. Evaluation of contextual information retrieval effectiveness: overview of issues and research. Knowl. Inf. Syst. 24 (1), 1–34, http://dx.doi.org/10.1007/s10115-009-0231-1.
Tessier, S., Wang, Y., 2013]. Ontology-based feature mapping and verification between CAD systems. Adv. Eng. Informatics 27 (1), 76–92, http://dx.doi.org/10.1016/j.aei.2012.11.008.
van Engelenburg, S., Janssen, M., Klievink, B., 2019]. Designing context-aware systems: a method for understanding and analysing context in practice. J. Log. Algebr. Methods Program. 103, 79–104, http://dx.doi.org/10.1016/j.jlamp.2018.11.003.
Voorhees, E.M., 1994]. Query expansion using lexical-semantic relations. Proceedings of ACM SIGIR Intenational Conference on Research an Development in Information Retrieval, 61–69.
Wang, Q., Yu, X., 2014]. Ontology based automatic feature recognition framework. Comput. Ind. 65 (7), 1041–1052, http://dx.doi.org/10.1016/j.compind.2014.04.004.
Yoshikawa, H., 1993]. Systematization of design knowledge. CIRP Ann. Manuf. Technol. 42 (1), 131–134, http://dx.doi.org/10.1016/S0007-8506(07)62409-3.