Context-Aware Cognitive Design Assistant: Implementation and study of design rules recommendations

Armand Huet[a]; Frédéric Segonds[a]; Romain Pinquie[b]; Philippe Veron[c]; Jerôme Guegan[e]; Antoine Mallet[d]

[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
[e] LaPEA, Université de Paris and Univ Gustave Eiffel, Boulogne-Billancourt, France.

ABSTRACT

Design rules are an essential interface to facilitate the information exchange between designers and experts. Despite many innovations in Knowledge-Based Engineering and Knowledge Management, unstructured design rules documents are still widely used in the manufacturing industry. Due to the complexity of the design process, these documents often contain hundreds of design rules, applicable in varying design contexts. Searching for the right rules according to a design context is demanding in time and cognitive resources. In this paper, we propose a Context-Aware Cognitive Design Assistant (CACDA) to capture the design context and perform tasks such as the recommendation of design rules, the verification of design solutions, or the automation of design routines. Contrary to existing works in model quality testing, the CACDA uses a proactive approach of design rules application and helps designers to provide error-free designs on first attempt. In this paper, we present the design rules recommendation system of the CACDA, its capabilities and its implementation. Then, to measure the impact of design rules recommendations on the design process, we compare our approach with the use of traditional design rules documents. Results show that the use of the CACDA's design rules recommendations lower the perceived difficulty of design rules retrieval from 75 to 43.5 on a scale of 100. On average, participants that used the demonstrator successfully applied 8.6 design rules on the 25 applicable design rules of our set. Participants that used unstructured documentation correctly applied 4.3 design rules. The global cognitive weight of the design activity as well as the design rules retrieval performances appear to be unchanged. These results demonstrate the usability of the demonstrator and show a positive impact on the design process and on the quality of CAD models. Future works will be focused on the overcome of the main limitations of our current experiments, with a panel of professional designers, a lager design rules set and the implementation of several lacking features of the CACDA into the demonstrator.

Keywords: Design rule; Computer-Aided Design; Knowledge Graph; Knowledge Management; Context-Awareness; Cognitive assistant, recommendation system, CACDA

1. Introduction

1.1 Context

In the manufacturing industry, an ideal mechanical part should bring an innovative and efficient answer to a technical issue while having high performances in every step of the part's life cycle (easy to design and reuse, cheap to produce, transport and maintain, etc.). A designer cannot meet all these expectancies and arbitrate efficiently between them when necessary. This is why companies sum up their collective knowledge in prescriptive design rules that, in a specific design context, will guide designers toward better designs.

Design rules are essential to the industry as they limit the number of flawed designs and push toward better design habits. It is a common knowledge that the design phase of a project freezes the majority of its cost [1], [2]. Therefore, any design mistake occurring during this phase will have negative consequences on costs and/or performances later in the lifecycle. Consequently, manufacturing companies have strong incentives to build and maintain design rules documentation.

1.2 Problem

In the manufacturing industry, a tendency of growing design complexity and the deployment of new and innovative technologies, lead to an ever-increasing number of design rules to consider when building a part. In fact, many challenges of the industry of the future will be primarily handled by the creation of new design rules [3].

Moreover, design rules are often stored in unstructured or semi-structured documentation in natural language. While easy to develop, this support is unsuitable to efficiently retrieve design rules. Despite companies moving toward model-based definition [4], designers still have to search the design rules adapted to their design context into PDF documents of hundreds of pages. From example, we studied several proprietary design manuals currently used by an aerospace company. These design rules documents represent a total of about 1900 pages that designers need to go through to identify applicable design rules. Moreover, several validation cycles can be necessary to insure the absence of design errors in a model. The time and effort dedicated to design rules retrieval and validation is lost for design creativity and CAD modeling, thus diminishing the productivity of the detailed design phase.

1.3 Proposal

Existing design assistants in the industry or in academic research are not adapted to manage large quantities of design rules in natural language [5]. They cannot process all the types of design rules written in industrial documents. Moreover, most of them adopt a CAD centric reactive approach to design rules application where the machine checks rules once the design is done. To replace unstructured design rules documentation, there is a need for a user centric solution that proactively process and distill relevant design rules. This is why we propose a Context-Aware Cognitive Design Assistant (CACDA) that will be able to perform on the fly design rules recommendations based on the user's design context. In a previous work, we presented the knowledge graph developed to structure – in a computable format – the information required to perform the CACDA's functions [6]. In this paper, we present in detail the implementation of the CACDA prototype, especially the design rules recommendation system. We demonstrate the usability of this demonstrator and investigate its impact on the design process of a major company of the aerospace industry.

In the first part of the paper, the literature review, we briefly remind the definition of our design assistant, especially its main capabilities, and the knowledge graph that structures the design rules in a computable format.

In the second part of the paper, we concentrate on the adaptation of an existing recommendation system that runs reasoning operations on the knowledge graph to distill the right design rules according to the current design context.

In a third part of the paper, we measure the impact of our approach on the design process. We present design experimentations realized with the CACDA demonstrator where 14 participants had to model a CAD part while searching applicable design rules. The CAD part is inspired from an existing aerospace part and the set of design rules comes from the design manuals of a large aerospace company. We compare our approach, supported by the CACDA, with the use of traditional design rules documentation.

2. Literature review

2.1 Presentation of the Context-Aware Cognitive Design Assistant (CACDA)

Our proposal to improve the retrieval and application of design rules is to develop a Context-Aware Cognitive Design Assistant.

2.1.1 Cognitive assistant

A cognitive assistant, also called a knowledge-based agent or an expert system, is a software that tries to enhance human-machine capabilities and performances in complex tasks [7]. They simplify users' interactions with knowledge bases, allowing them to focus on crucial or creative tasks. Design assistants that focus on design rules for the manufacturing industry are classified as 3D Model Quality Testing tools by González-Lluch *et* al. in their literature review [8]. These tools have a CAD centric approach. They try to detect design flaws in 3D models so that designers can correct them. The "traditional" approach is procedural and consists in representing design rules by exploration algorithms that scan the 3D model. Commercial software currently used in the industry adopt this approach [9], [10] and research teams work to improve these exploration algorithms [11], [12]. Recent scientific works explore a new approach of 3D quality testing based on a semantic representation of design rules [13]–[16]. In these approaches, design rules are often manually written in code like SWRL or pseudo-code like SADL. However, significant progress have been made in the automated capture of design knowledge into computable semantic relations [17]–[20]. These technologies lead to more flexible CAD quality software that integrate varying design rules. In recent works, Yang *et* al. use these technics to develop a new software for model quality analysis [21], [22].

Despite recent progresses, most solutions rely on formal geometrical rules and require expert knowledge to edit or modify the definition of design rules. Moreover, the CAD centric approach implies constant validation cycles and corrections from the designer. Conversely, a designer centric approach would focus on the designer's need and ensure he/she is in the best conditions to avoid design errors and make the best design possible at the first attempt. In fact, if many sources can be found on the digital performances of model quality testing tools, only few case studies measure the impact of these tools on the design process. There is a need for a design assistant oriented towards designers guidance and instruction [23] to replace unstructured design rules documentation in the industry.

To meet this need, Kim *et* al. present the structure of a Virtual Design Assistant based on deep learning technologies [24]. The goal of this AI platform is to model unstructured design language and extract design requirements in order to provide design suggestion to designers. In the same effort to develop a proactive approach of design rules application, we propose a Context-Aware Cognitive Design Assistant. The CACDA performs personalized services to a designer based on his/her design context. One of its services is to perform on the fly design rules recommendations based on a knowledge graph featuring design rules linked to structured contextual information. This user-centric approach is necessary to integrate seamlessly design rules recommendations and design guidance to the design process. By facilitating the retrieval of relevant design rules in near real time and supporting design workflow, our goal is to guide designers toward error-free designs and to replace unstructured design rules documentation in manufacturing companies.

2.1.2 Context-awareness

Context-awareness is the capability for a software to sense and react to contextual information [25]. Contextual information has a broad definition that can apply to any relevant information concerning the user and/or the software [26]. Context-awareness is frequently used for information retrieval to establish user-centric recommendations [27], [28]. Moreover, context histories can be used to improve recommendations and provide analysis based on users past activities [29]–[31]. Several authors deployed context-aware systems in factories [32]–[34]. We propose to develop a design assistant that links design context to design rules knowledge in order to perform design rules recommendations on the fly. To the best of our knowledge, this approach does not exist in the literature. The use of context-histories analysis could lead to the identification and automation of routine design operations to improve the design process further more. A technology of context prediction [35] could be used to anticipate these routine designs and question the user about their automation.

2.1.3 Context-Aware Cognitive Design Assistant (CACDA)

As presented in previous work [6], CACDA is defined as follow:

"*A context-aware cognitive design assistant is a seamless, ubiquitous and intelligent computer program that senses relevant information that can be used to characterize 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*"

The core service of the assistant is to perform on the fly design rules recommendation. However, other services can be expected like design routine detection and help in design validation. In this paper, we focus on the design rules application aspect of the CACDA, structured in three main functions, presented in Figure 1. The first function is to capture design rules from enterprise documentation and model them in a knowledge graph. The second function is to perform design rules recommendations and help the designer to identify design rules that apply to his/her design context. The third function is to guide the designer in the respect of identified design rules. In this paper, we focus on the implementation and study of services one and two, which are the main pillars of the CACDA.

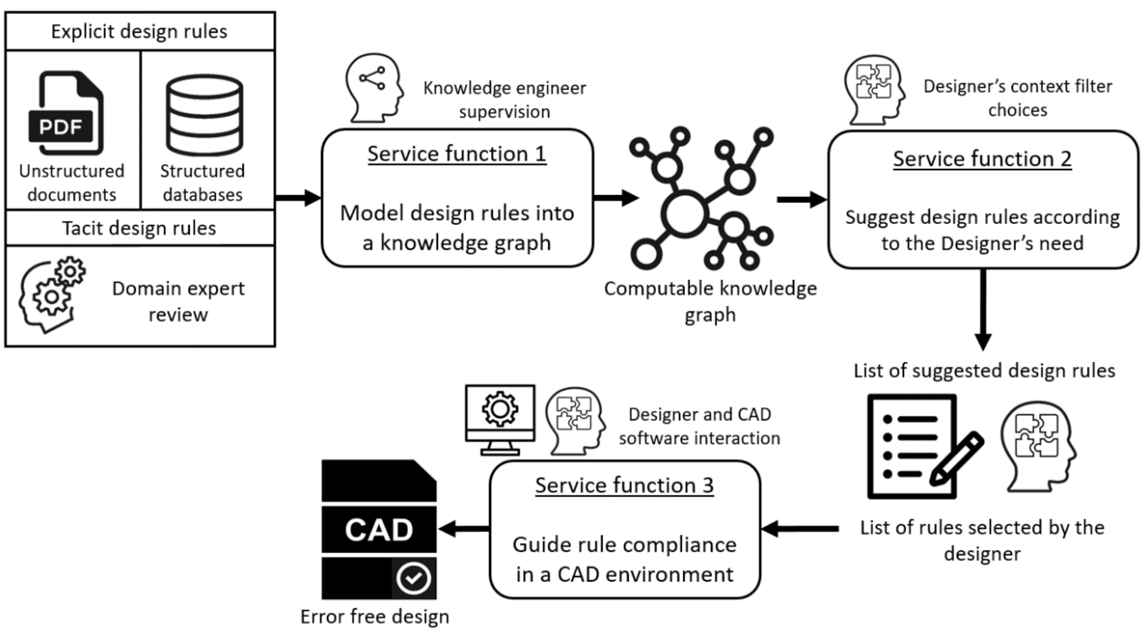


Figure 1 : Functional process of the Context-Aware Cognitive Design Assistant's core service

2.2 Knowledge graph

In its core, a knowledge graph is a set of data structured in a graph. This representation is essential in many application fields [36] including recommendation engines [37] and context-aware systems [38], [39]. A crucial step in the development of CACDA is therefore to build a graph data model adapted to its functionalities.

2.2.1 Graph databases

Graph databases are NoSQL databases that use a graph representation of data. Due to the growing need for knowledge graphs in many domains, several industrial graph database services emerged recently. A comparison between the most frequently used graph databases can be found in [40]. Graph databases like NEO4J have flexible structures that allow an easy manipulation of data structure. In order to use a graph database as knowledge base for a specific usage, a property graph data model is needed. The property graph data model, presented in [41], imposes a graph structure adapted to specific applications. Therefore, a specific graph data model is essential to develop the CACDA.

2.2.2 Data model of the CACDA

The CACDA's property graph data model must be adapted to CACDA's services functions as described in Figure 1. In order to perform design rules recommendations based on contextual information, both must be stored in the data model. We build on the graph data model presented in [6], [42]. Design context is structured in four different dimensions or sub-contexts. The semantic sub-context models semantic information of the design context, including design rules knowledge and domain specific words and concepts. The engineering sub-context models technical knowledge of the manufacturing company, manufacturing process and tools for example. The social sub-context models social relations in design teams. The IT sub-context models the digital environment of the designer, especially the CAD model on which he/she is working. For example, Figure 2 presents the graph data model of the semantic sub-context.

A consolidation process serves to link all four sub-contexts resulting in a unique

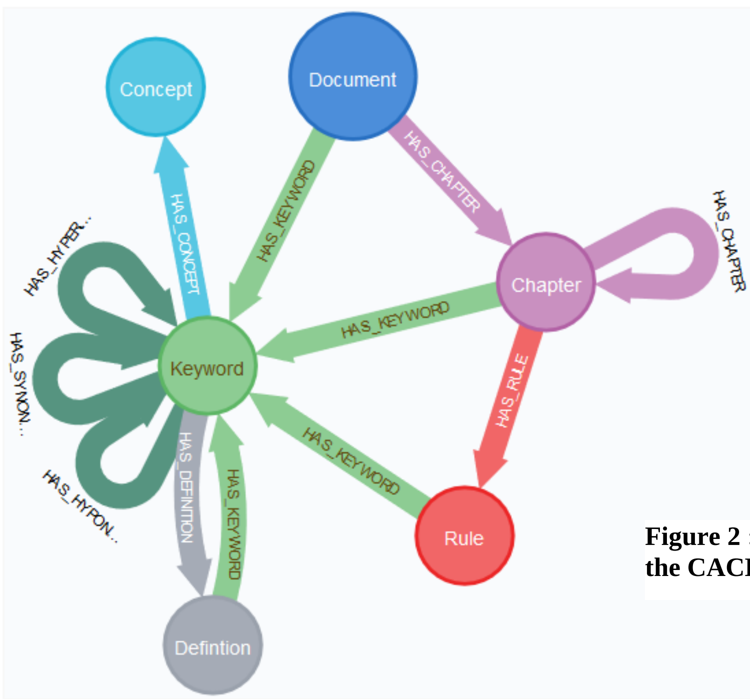


Figure 2 : Semantic sub-context of the CACDA graph data model

knowledge graph. It is therefore possible to link design rules with contextual information and perform design rules recommendations based on the design context.

2.3 Recommendation engine

Object recommendation in graph databases is a vast and dynamic field of research. In this chapter, we will review some of the most frequent approaches to object recommendation. The goal of a recommendation algorithm is to extract from a database, information that is relevant to a user. Recommendation systems are classified in two main categories, collaborative filtering and content-based filtering. However, as research in this domain is very dynamic, authors regularly propose new approaches [43], [44]. The goal of the review is to identify the most relevant approach to develop CACDA's demonstrator.

2.3.1 Collaborative filtering

The main idea of collaborative filtering is that users who rate items similarly or have similar behaviors will probably share similar interests. Therefore, collaborative filtering algorithms use users' ratings and activity to recommend content. This strategy is manly used to perform recommendations based on users' preferences in various domains, such as social medias [45] and movies recommendation [46]. Collaborative filtering algorithms often rely on a bi-partite weighted graph composed of two labels, items and users [47]. Weighted relations in this graph correspond to users' ratings of items. Algorithms use this data to compute similarity scores between users and items. When a user asks for a recommendation, collaborative recommendation systems will recommend items that are the most similar to him/her. Authors explored a large variety of approaches to perform this task. For example, Huang *et* al. propose a context-aware collaborative filtering system [48]. Su and Khoshgoftaar propose a survey of existing collaborative filtering technics in [49].

2.3.2 Content based filtering

Content based recommendation systems recommend items to a user based on items' representation and a profile of the user's interest [50]. The traditional content-based approach represents user interest by a list of tags (concepts, subjects or genres) a user is interested in. This list can be deduced by a user's profile or by the user's history. Then, as every item in the database is linked with one or more tags, the system can compute a similarity score between the user and an item based on the tags they share. A standard method for similarity measures is cosine similarity [51] but many others exist.

2.3.3 Path analysis

The recent deployment of new technologies like natural language processing and context-awareness led to new types of heterogeneous graphs. A standard approach to perform recommendations in a heterogeneous network is based on path analysis. Authors use path analysis to compute similarity measurements between nodes in a graph [16], [52]. For example, authors use path analysis methods in semantic graphs to compute semantic similarities between texts, sentences or words [53]–[55]. In a recommender system based on this approach, a "user set" of nodes represents the user interest [56]. The system selects nodes that are the most similar to the user set. However, similarities between nodes are complex to compute – they require the exploration of every paths between the two nodes considered – and many are required to perform recommendations. Therefore, this approach is not adapted to real time recommendations in a graph database that is continuously evolving in real-time.

Bogers proposes ContextWalk, a path analysis method based on a random walk that performs recommendations on a context graph [57]. In a graph of N nodes, the author considers a vector space of N dimensions where each dimension represents a node. A position probability vector V is defined by Equation 1, where $V(i)$ is the value of V in the dimension i and represents the chance for the position to be on the node i.

$$V(i) \in [0, 1] \ ; \ \sum_{i=1}^{N} V(i) = 1$$

Equation 1 : Position vector

The author also considers a matrix X named transition probability matrix, where $X(i, j)$ is the chance, if positioned on the node i, to move to the node j in the next step. The behavior of the transition probability matrix is described in Equation 2, where V_n is the position probability vector of the graph after n steps.

$$V_{n+1} = V_n X$$

Equation 2 : Step in the graph with the transition probability matrix

The random walk begins with the initial vector V_0 that represents the user interest. The user and/or the system can select any set of nodes to initiate the random walk. During the walk, all sub-contexts are explored simultaneously. After n steps, the system reads recommendation results in the vector V_n. In fact, V_n associates every node in the graph with a probability score. The system simply recommends the k most probable nodes in a specific category. This category can be a specific label, like movies in the example of the author, but it can also be an entire sub-context in order to suggest contextual elements of potential interest.

2.4 Summary of the state of the art

In the first part of this chapter, we presented the Context-Aware Cognitive Design Assistant and defined its core functionalities:

Cognitive assistants are software aimed at enhancing human-machine capabilities in complex tasks. Existing cognitive assistants for design rules application concentrate on the identification of design errors in a CAD part model. This CAD centric approach lead to a reactive application of design rules where designers need to correct their mistakes afterward. There is a need for cognitive assistants with proactive approach of design rules application. Such tools would proactively guide designers in their search and application of design rules knowledge to improve the design process. This is why we consider this approach as designer centric. Through our review of literature, we highlight the lack of papers on proactive, designer centric cognitive assistants as presented in Table 1. This is why we propose a new cognitive assistant, the CACDA.

Table 1 : synthesis of related works

CAD centric approach **Reactive application of design rules**	Number of papers : 9 [8, 11, 12, 13, 14, 15, 16, 20, 21]

Designer centric approach **Proactive application of design rules**	Number of papers : 1 [6, 23]
CAD centric approach = focus on the detection of errors in CAD design (geometry and topology) **Designer centric approach** = focus on designers' need for information and guidance during the design process **Reactive application of design rules** = Detection and correction of design errors after they occur **Proactive application of design rules** = Preventing design errors from occurring	

The CACDA is a context-aware software. It can model contextual information and use it to provide services to the designers. Context-awareness is frequently used to develop user-centric information retrieval systems. We propose to use a context-aware system that performs used centric design rules recommendations as the foundation element of the CACDA.

In the second part of the literature review, we presented the propriety graph data model used by the CACDA to model design rules and contextual information into a knowledge graph. The design context is divided into four different sub-contexts: semantic, engineering, social and IT. These sub-contexts are all represented and interconnected in the knowledge graph in a way that allows the CACDA to perform its expected functionalities.

The last part of our literature analysis is focused on recommender systems. We reviewed most frequent recommendation approaches, which are collaborative and content-based filtering. The most adapted strategies to perform context-aware recommendations on heterogeneous multi-dimensional domains are based on path exploration algorithms. These algorithms rank items that are the most linked to a set of initial nodes representing the user interest. We reused the *ContextWalk* [57] algorithm for the CACDA design rules recommender system.

3. Proposal

This chapter presents the demonstrator of the CACDA's design rules recommender system. The goal of this demonstrator is to perform design experimentations that demonstrate how context-awareness helps to satisfy design rules. It is also a demonstration of the usability of our approach, our software architecture and user interface. In order to be representative of a manufacturing industry use-case, it has to provide two services (see chapter 2.1.3):

- "Model design rules into a knowledge graph",
- "Suggest design rules according to the designer's need".

The third service cannot be developed without obtaining sound results with the first two services. The demonstrator will model CACDA's knowledge graph into a graph database and use it to perform design rules recommendations to a designer. In this paper, our demonstrator captures two sub-contexts: the semantic sub-context and the engineering sub-context. The social sub-context and the IT sub-context are currently under prototyping.

First, we will focus on the recommendation algorithm and its interaction with the knowledge graph. Then we present the software architecture and the implementation of the demonstrator.

3.1 Recommendation algorithm

In order to develop a recommender system for our assistant, we chose to adapt the ContextWalk proposed by Bogers [57]. In fact, a random walk over the knowledge graph has several advantages:

- As highlighted by the author, the system is easy to adapt to different domains and data-models.
- The system is adapted to heterogeneous knowledge graphs as the algorithm explores every sub-contexts simultaneously to perform recommendations. Every node explored during the random walk is ranked. Therefore, the system can recommend design rules but also coherent contextual elements that can be used as contextual filters.
- The implementation of new sub-contexts or new contextual elements is very easy. In our case, it allows us to keep the same recommendation system while adding new sub-contexts and contextual information.
- The initial probability vector can feature any weighted set of nodes. In our use-case, the user can select keywords but also any contextual filters like semantic concepts and technical elements that will influence recommendation results.

In the ContextWalk system as presented by Bogers, contextual elements do not have relations between them and each sub-context contains only one label. Our sub-contexts are by comparison complex networks with multiple labels. When building our probability transition matrix we need to create a list of nodes ordered by sub-context and then by label. The system then associates a node to its index in the list. Labels and sub-contexts are associated with arrays of consecutive indexes in the list. These indexes are used to build the transition probability matrix and to read results vectors, as presented in Figure 3. To recommend design rules to the user, the system selects the most probable nodes of the area associated with the label *Rule*. To select contextual filters from the engineering sub-context, the system reads the area associated with this sub-context.

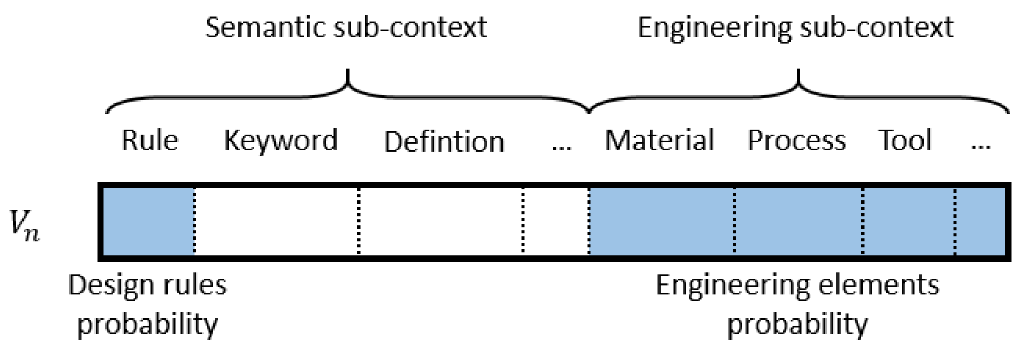


Figure 3 : labels structure into the position vector

Once every node is indexed, our algorithm builds an adjacency matrix A where $A(i,j)$ is equal to one if a relation exists between node i and node j or is equal to its weight if this relation is weighted. $A(i,j)$ is equal to zero if no relation exists between nodes i and j. To obtain our transition probability matrix we need to row normalize this matrix so that the sum of each row is equal to one. Bogers introduces a self-transition probability α equal to 0.7 in order to slow down the graph exploration. We keep the same value for α and insure an equal weight between sub-contexts during the row normalization.

3.2 Software architecture

The demonstrator is a Python-based Dash[1] web application. Its first function is to model design rules and contextual information into a graph database. We use NEO4J[2] as a graph database. Our main code in Python reads and writes in the graph through the library Py2NEO[3]. The overall software structure of the demonstrator is presented in Figure 4.

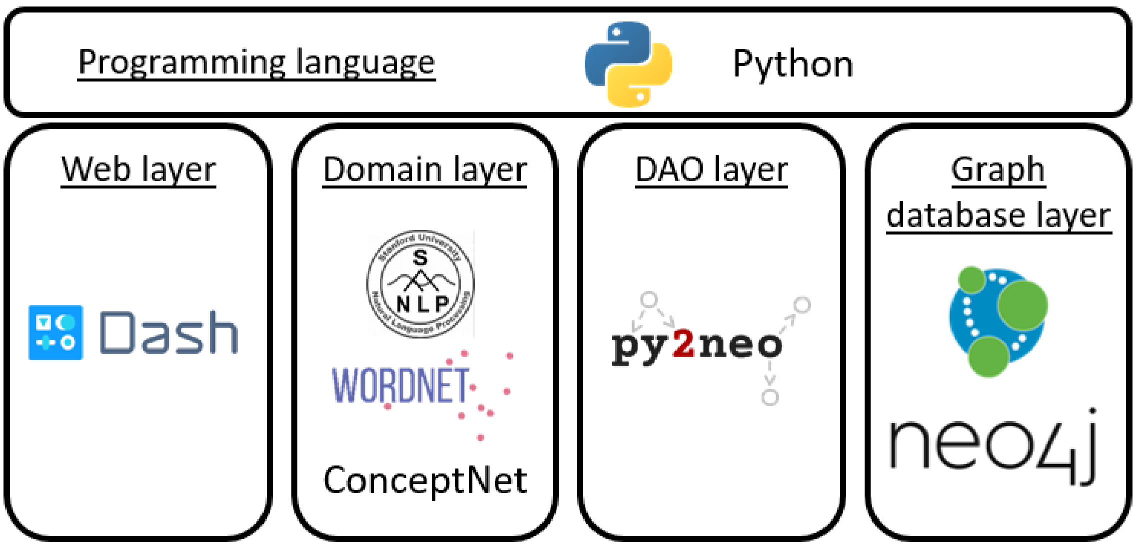


Figure 4 : Software structure of the CACDA demonstrator

At the beginning of the implementation, the design rules set used in the experiment is stored into a semi-structured spreadsheet document. To build the semantic sub-context, the demonstrator relies on the natural language processing toolkit Stanford

---

[1] https://plotly.com/dash/
[2] https://neo4j.com/
[3] https://py2neo.org

CoreNLP[4], the thesaurus WordNet[5], and the ontology ConceptNet[6] to extract, disambiguate and enrich the semantic sub-context. Figure 5 shows an example of indirect links between two keywords resulting from semantic enrichment. More details on this process are given in [6].

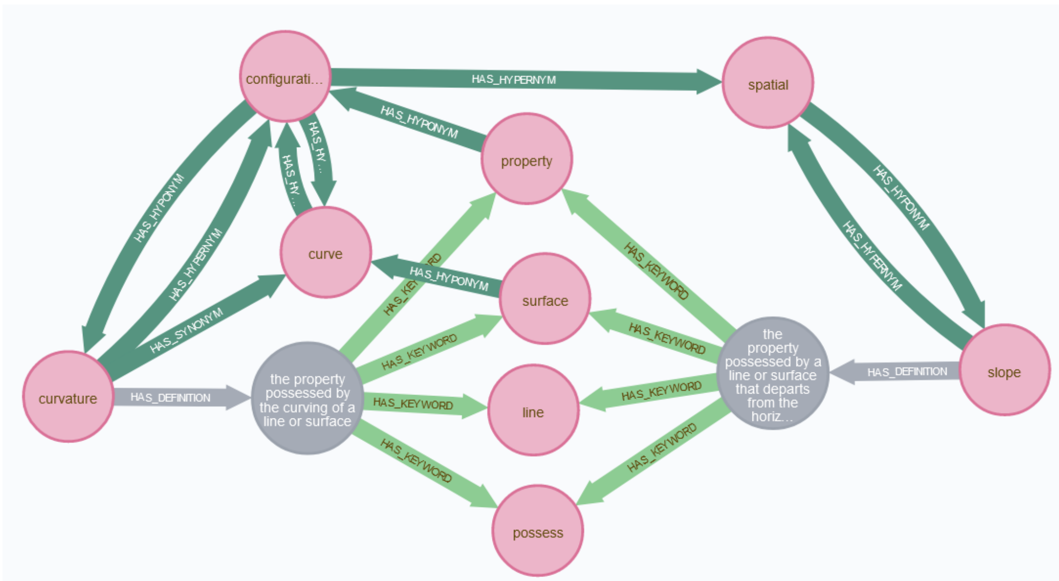


Figure 5 : Example of indirect links between keywords “curvature” and “slope”

A hand-made dictionary of technical terms extracted from glossaries defined in our industrial data set is used to identify domain-specific vocabulary and linked them together in order to build the technical sub-context. After the end of the graph database writing process, the system builds the probability transition matrix as described in chapter 3.1. Figure 6 represents this process, structured in 5 steps:

Step 1: The set of design rules is extracted from a semi-structured spreadsheet that contains the main statement of design rules, their source document and the chapter

[4] https://stanfordnlp.github.io/CoreNLP/
[5] https://wordnet.princeton.edu/
[6] https://conceptnet.io/

they belong to. The Python program captures this information and writes associated Rule, Document and Chapter nodes.

Step 2: Each node that contains a text attribute is linked with all keywords extracted from this text. Keywords linked to Rule nodes are used for semantic enrichment by linking them to definitions, synonyms and related concepts that are selected after a disambiguation process.

Step 3: In our demonstrator, the disambiguation process aims at selecting the most appropriate meaning for a word before semantic enrichment. This process is essential, as the meaning of a technical word is very likely to be different from its common sense. Thus, we consider each WordNet synset of an input word as a potential meaning of this word. Each synset is associated with a list of words extracted form WordNet information on this synset (Words of the definition, synonyms, etc…). The system computes a cosine similarity measure between this list and a large list of words extracted from our knowledge graph. To build this list, the system selects the 400 keywords of the graph that have the shortest paths to the input word. Finally, it selects the most similar synset for the semantic enrichment of the input word. For example, let us consider the input word "turning". A first synset has the definition: "act of changing in practice or custom". This synset is associated with a list containing words like "act", "change", "reversal", "variation" or "entail". A second synset of "turning" has the definition: "the activity of shaping something on a lathe". The list of words for this synset contains words like "shape", "lathe", "formation", "fabrication", "manufacture", "material" and "creation". As more of these words are susceptible to be present in our graph database, the system will select the second synset and use its information for the semantic enrichment of the keyword “turning”.

Step 4: The system has a library of technical terms used in the aerospace industry. The system will add an appropriate technical label, like “MATERIAL”, to any keyword of the graph database that is featured in this library. Technical keywords are then manually linked together.

Step 5: The system builds the transition probability matrix from the resulting graph.

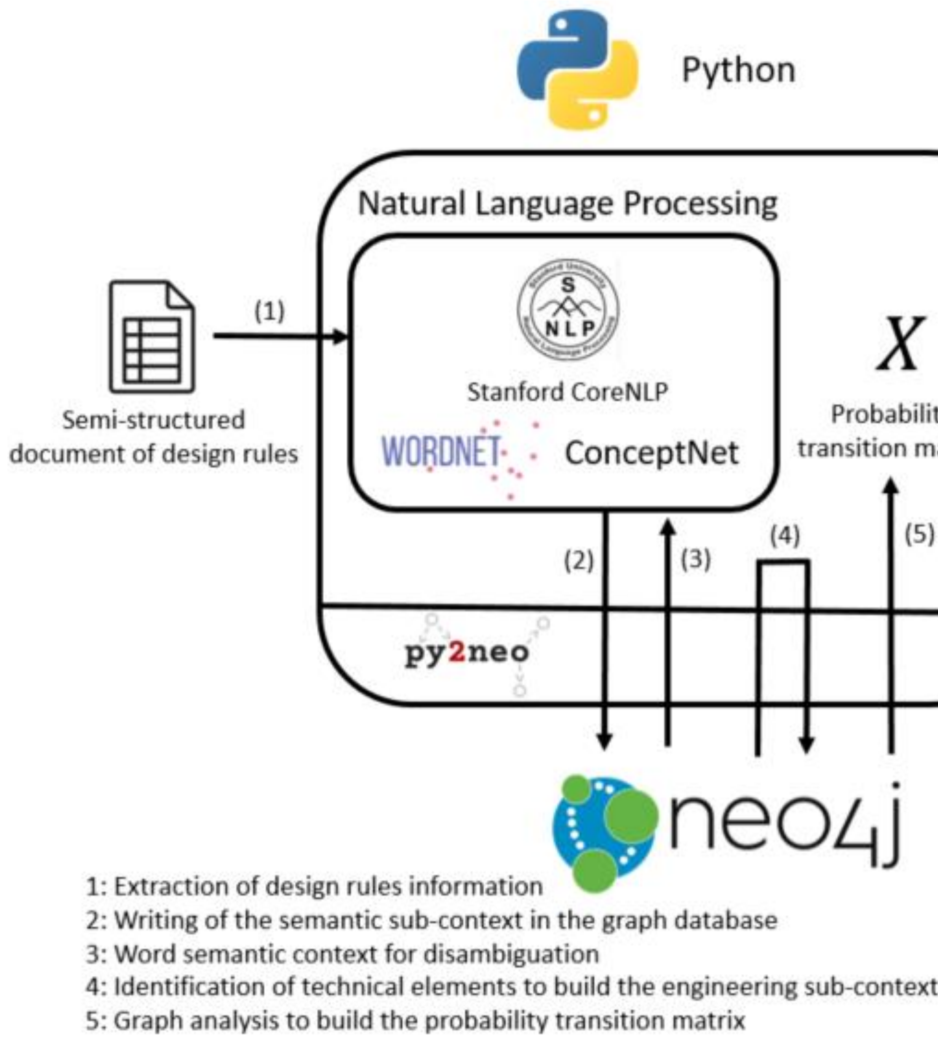


Figure 6 : Software architecture

The second function of the demonstrator is to "suggest design rules according to the designer's need". To realize this function, the demonstrator has to interact with the designer. It does so through a web user interface generated by a Python-based Dash web application (Figure 7).

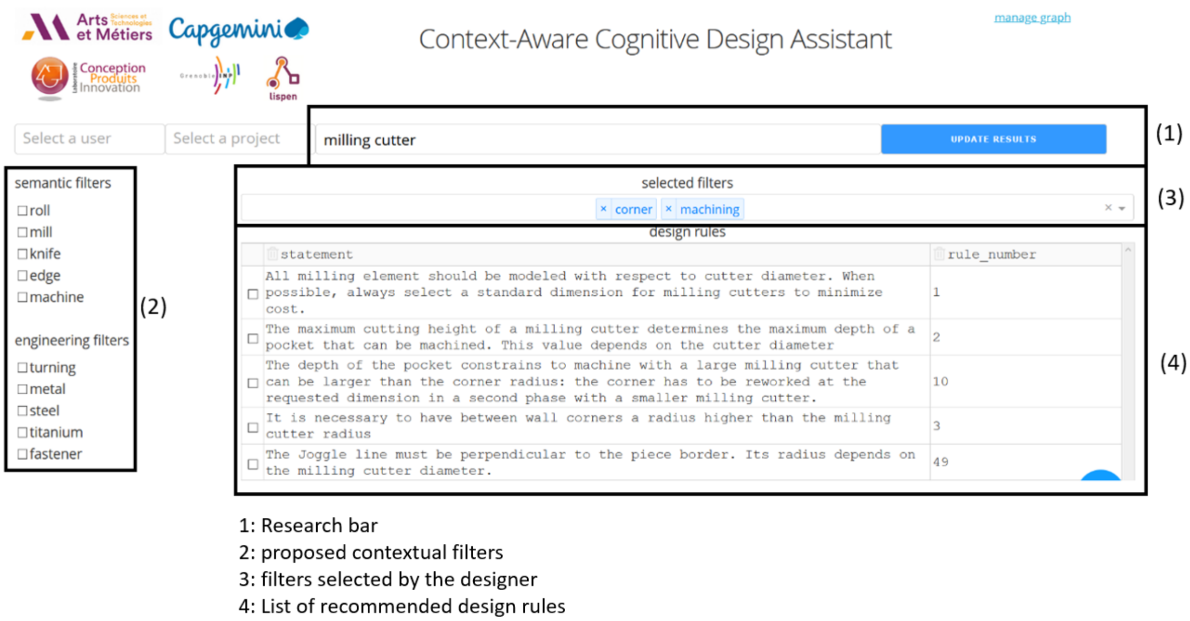


Figure 7 : Dash web interface

The designer can enter a set of keywords (1) to initiate a recommendation. Each recommendation returns a list of design rules (4) and a list of contextual filters named facets in the information retrieval domain (2). The designer can select these filters (3) to influence future recommendations. For experimental purposes, when the designer selects a design rule to consult, the demonstrator opens it in a PDF document. Figure 8 presents the full process of design rules recommendation, structured in 6 steps:

Step 1: In the user interface, the designer can write keywords in the research bar and select contextual filters. When the designer clicks on update results, the system uses this information to build a list of input nodes for the recommendation process.

Step 2: From the list of input nodes, the system builds V_0 and uses it to initiate the random walk.

Step 3: The system performs random steps and returns a probability position for each word structured into the vector V_7.

Step 4: The system reads V_7 to present the list of recommended design rules as well as the updated list of contextual filters, in the user interface.

Step 5: In the results display area of the user interface, the designer can select one or more design rules and click on an open button. The system opens selected design rules in PDF pop-ups.

Step 6: The designer reads selected design rules and uses the information to design the part.

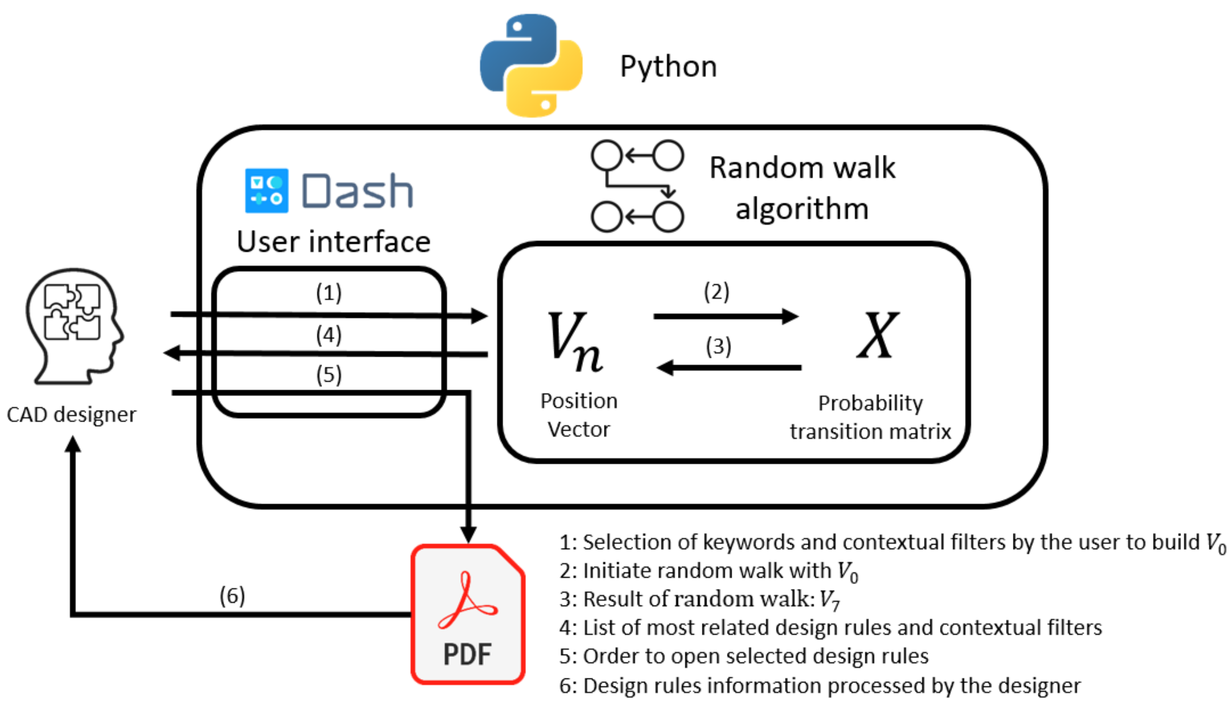


Figure 8 : Design rules recommendation process

3.3 Future implementation with a CAD system

Several features are still lacking in the demonstrator and will be added in future developments. We still have to model two more sub-contexts into our knowledge graph. We will use experimental data, presented in the following chapter, in order to build a social sub-context from real human interactions with the design rules set and the demonstrator.

We plan to capture the IT sub-context in real time by coding a data link with a CAD environment. Most CAD software have knowledgeware APIs that can be used to capture geometrical and logical features. CAD systems also have exchange protocols to communicate with external programs. Of course, the capture of the IT sub-context in near real time requires to update the probability transition matrix. The theoretical process of updating the matrix is not a challenge. For each new node added to the graph, a dimension is added to the adjacency matrix to feature every relation between the new node and the rest of the graph. Once all new nodes and relations have been added, the row normalization of the matrix is performed again to obtain the updated probability transition matrix.

We also plan to detect design events into the CAD environment like the usage of specific modeling tools for example. The demonstrator will use these events to select initial nodes and directly influence design rules recommendations in real time. Figure 9 presents this evolution into the logical schema of the design rules recommendation process. Two new steps are added to the recommendation process:

Step 7: The system performs a scan of the CAD environment and updates the probability transition matrix accordingly.

Step 8: When the system detects an IT event, it automatically adds related IT nodes to the list of user selected initial nodes and starts a new research.

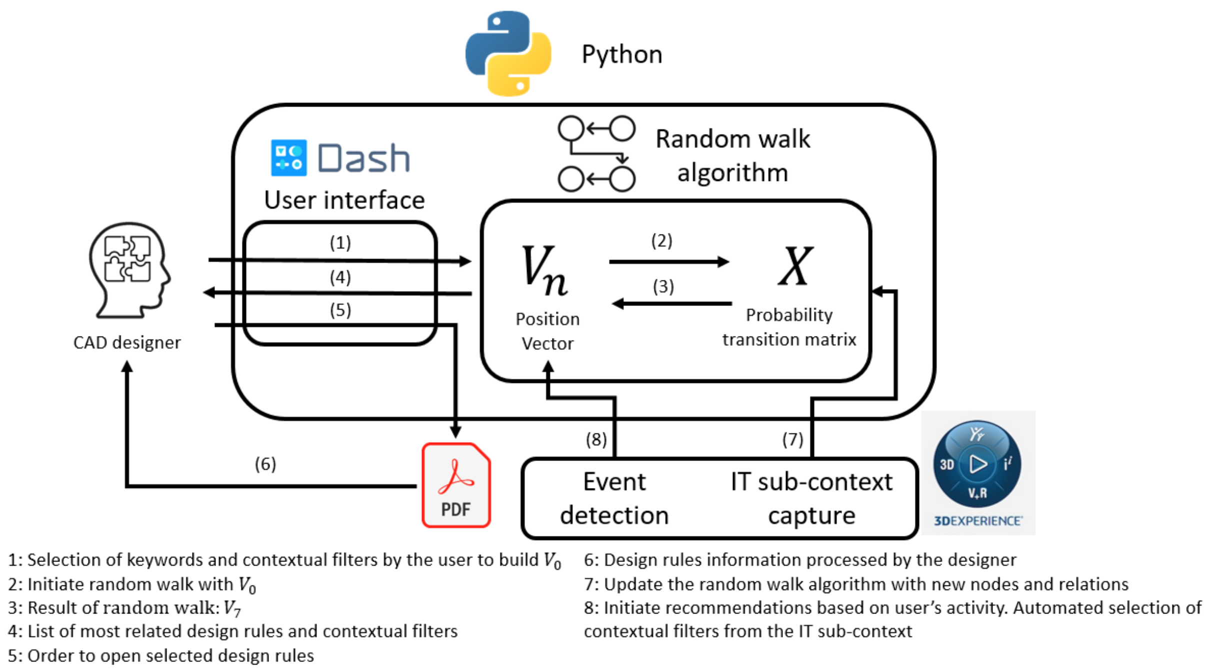


Figure 9 : Demonstrator future developments

4. A manufacturing industry use-case for CACDA

The goal of this experiment is to demonstrate the usability of the CACDA demonstrator in an industrial context and to measure its impact on the design process. We have asked two groups of participants to realize the same design activity constrained by the same set of design rules. As our goal is to replace unstructured design rules documentation, participants of the control group have access to a large PDF document that contains our set of design rules. The test group can only access design rules of the set via the CACDA demonstrator. We used the design rules to populate the knowledge graph of the assistant and create the semantic and engineering sub-contexts. Comparison in design performances between the two groups will highlight potential benefits and drawbacks of our approach. It will also enable us to identify any usability issue with the current state of the demonstrator. Unstructured design rules documentations present an issue when designers have to model a part while searching applicable design rules. To obtain meaningful results when comparing the two groups, we need to place them in a situation similar to this design context. This is why we propose an industrial use-case that models a design context in the aerospace industry.

4.1 Presentation of the use-case

4.1.1 Design rules set

The design rules set is entirely extracted from proprietary design manuals developed in the aerospace industry. It features 102 design rules. In this set, 25 design rules are directly applicable on the test part. The set of design rules is representative of our input design manuals in which a wide range of design situations are considered. This is why a majority of design rules are not relevant to the design situation of our protocol. Thus, designers have to search for the applicable design rules contained in the given dataset.

The PDF document of the design rules set is structured in different chapters to facilitate research, as an industrial documentation would be. To insure that each group has access to the same information, when a participant selects a design rule in the user interface of the CACDA demonstrator, a window containing the selected rules in a form of a PDF pops up.

4.1.2 Design task

Participants are placed in a design situation as close to an industrial context as possible. They have to realize the detailed design of an airplane mechanical part. The part is made of aluminum alloy and has to be manufactured with a 3-axis milling machine. All participants begin their design on the same CAD model of the pre-machining part presented in Figure 11. For experimental purposes, we consider that no sizing calculation is necessary. Participants must respect minimal thicknesses for pockets' bottom and walls that insure acceptable mechanical proprieties. Figure 10 shows the minimum wall thickness as presented on the initial CAD model.

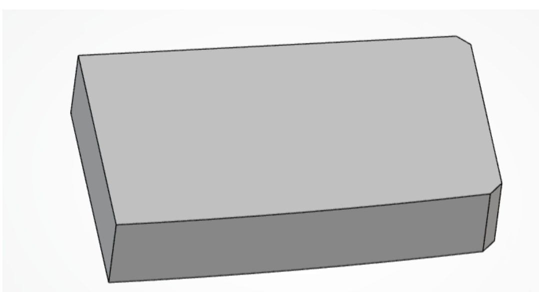


Figure 11 : CAD model of the pre_milling part

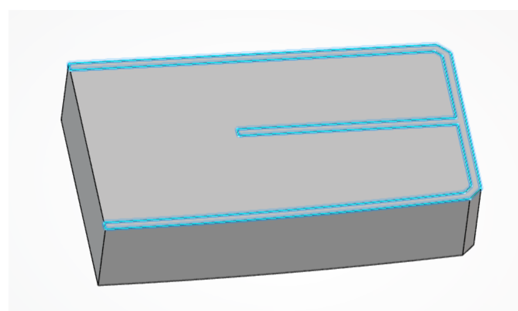


Figure 10 : Sketch of minimal wall thickness on the initial part

Participants shall achieve four design goals in their CAD modeling task:

- They shall respect the applicable design rules.
- They shall minimize the volume of their final part.
- They shall design the run-out of the central stiffener inside of the main pocket. All other stiffeners are at maximum height.

- They shall design the fastened joint of the part with an Aluminum plate of known dimensions. Once again, no sizing calculation is required, participants have to minimize the fasteners size and maximize their number.

These goals are presented to participants on a printed-paper. Participants have no further instructions than these four design goals. All other meaningful information needed for their task is accessible in the design rule set. Participants are allowed to ask questions about design goals to ensure that the understanding of the use-case is not a differentiating factor between participants. No question about the design strategy, the interpretation of a design rule or the set of design rules is answered. To simplify results analysis, we minimized the number of possible design outcomes that respect both design goals and design rules. The Following paragraph presents expected results for each design step:

The first step is to design the main pocket (Figure 12).

The second step is to design the central stiffener run-out (Figure 13).

The minimum thickness for pocket bottom is 4 mm shorter in the top right end on the part. We expect participants to design a sub-pocket in order to minimize the volume of their part (Figure 14).

The final step is to design the fastened joint (Figure 15)

Participants have no knowledge of these expected design steps. They can apply any design strategy they want as long as they respect design goals and design rules.

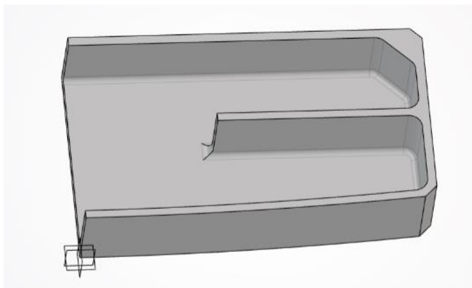
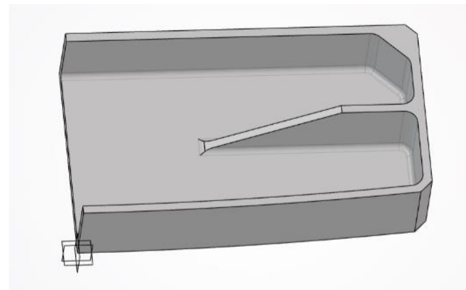
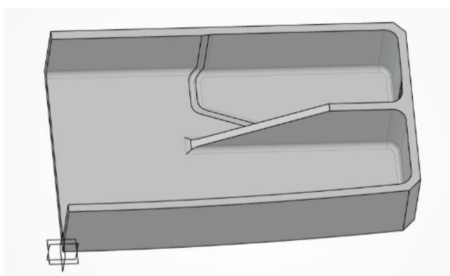
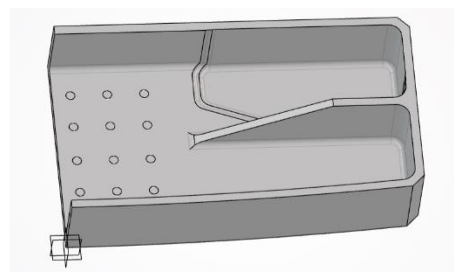
Expected design steps of the experimental part	
 **Figure 12 : Final part _ main pocket**	 **Figure 13 : Final part _ stiffener run-out**
 **Figure 14 : Final part _ sub-pocket**	 **Figure 15 : Final part _ fastened joint**

Table 2 : Design steps of the expected part

4.1.3 Measures

We identified four different parameters to measure the performance of the design process. Each parameter is composed of one or more measures. For each measure, we performed an independent samples t-test to highlight significant differences in the average scores of each group.

Design rules retrieval. The research of applicable design rules is an essential part of this experiment. A design rule is applicable if its information is necessary to realize an error free design of the test part. Participants have to register all design rules they intend to use in their design. Participants have all freedom to add or remove design rules from their list at any moment. In a participant's list, we consider applicable design rules as true positives and other design rules of the set as false positives. We count as false negatives, applicable design rules of the set that have not been selected by participants. These results are then used to calculate precision and recall value. We then calculate the F-factor to get a score that balances precision and recall. Detailed formulas of these parameters are given in Table 3.

Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
F-factor	2 x (precision x recall) / (precision + recall)
Number of selected design rules	TP + FP

Table 3 : formulas of design rules retrieval parameters (TP = true positive, FP = false positive, FN = false negative)

Design rules application. The score of design rules application is measured on participants' final CAD part. For each applicable design rules of the set, we increase the score of a participant by one if he/she applied it correctly. Partially applied design rules only count as 0.5 point. We decided to count correctly applied design rules and not design errors because it would advantage unfinished designs. Indeed, a participant who didn't have the time to design the fastened joint will not make any error on design rules associated with this feature. We also measure the percentage of design rules correctly identified by participants but not correctly applied. The final measure of this parameter is the final volume of the part.

Time measures. The use of the demonstrator may influence the duration of the experiment and more specifically the time dedicated to design rules retrieval. It is not possible in our use-case to differentiate the time used to find a design rule from the time used to understand and decide to use this design rule. In fact, our participants perform both activities simultaneously. This is why we decided, for each participant, to measure the total duration of the experiment as well as the time that was not dedicated to CAD modeling. We did not count any time spent on other activities than CAD modeling and design rules retrieval and understanding. In order to allow time differences between participants, we did not impose a maximum duration and participants were free to stop whenever they want. However, many participants still had time constraints and imposed themselves a maximum duration for their task.

Mental workload and perceived difficulty are measured at the end of the experiment. When a participant decides to stop his/her task, he/she has to complete a questionnaire. This final questionnaire includes a NASA-TLX rating scale [59]. The cognitive weight is composed of six different dimensions. Participants rate the impact of each dimension in their task, in a scale from zero to one hundred. NASA-TLX is considered as an appropriate method to measure designer workload and the difficulty of a design task [60], [61]. Finally, we asked participants to rate their perceived difficulty of the design retrieval activity, the CAD modeling activity and of the overall task. We use the same gradients for perceived difficulty as for the NASA-TLX measures.

4.2 Protocol

Fourteen engineering students in their last studying year participated in our experiment. These participants have novice to intermediate CAD modeling knowledge. They all assisted at least at 40 hours of lessons on CAD modeling during their engineering learning program. A majority of them already used a small set of ten or less design rules often imposed by a teacher during educational design projects. We conclude that participants had no experience in dealing with an industrial size set of design rules. They also had no knowledge of the set of design rules from the aerospace industry used in the experiment. We consider that participants are equipped with design skills but very small knowledge about design rules. They stand as inexperienced designers in a manufacturing company.

As presented in

Table **4**, our test and control groups are balanced in terms of age, gender and level of skill in CAD modeling.

	Number of participants	Average age	Level of skill in CAD modeling			Gender	
			Novice	Confirmed	Expert	Men	Women
Control group	7	23,4	4	3	0	5	2
Test group	7	23,4	5	2	0	5	2

Table 4 : Panel of participants

4.3 Experimental results

Table 5 presents average results of each group for design rules retrieval. These results do not show any significant difference between the two groups. We notice that average values are in line with our hypothesis, with higher scores for recall, F-factor and on the number of selected design rules in the test group. We also notice that results are more homogeneous in the test group with much lower standard variations.

	Precision	Recall	F-factor	Number of selected design rules
Control group	0.49 (SD = 0.26)	0.18 (SD = 0.13)	0.25 (SD = 0.16)	8.6 (SD = 5.3)
Test group	0.53 (SD = 0.13)	0.26 (SD = 0.04)	0.34 (SD = 0.05)	12 (SD = 3.7)
P-value	0.757	0.155	0.152	0.187

Table 5 : Design rules retrieval performances

Results of the design rules application parameter are presented in Table 6. We observe that the design rules application score in the test group is the double of the control group. This difference is significant with a P-value inferior to 0.05. However, there is no significant difference on the error percentage on correctly identified design rules, despite a large difference in average values. These is also no significant difference measured in the volume of the final part. In addition, for this parameter also, we notice that the variability is much higher in the control group for every measure.

	Design rules application	Error percentage on correctly identified design rules	Volume of the final part (cm^3)
Control group	4.29 (SD = 2.8)	62.37 (SD = 35.48)	465 (SD = 96)
Test group	8.57	24.69	454

	(SD = 1.3)	(SD = 8.56)	(SD = 37.1)
P-value	0.003	0.155	0.782

Table 6 : Design rules application performances

Table 7 presents time measures. There is no significant difference between the two groups for this parameter.

	Total time of the experiment	Time of design rules research
Control group	2h 12min (SD = 15.59 min)	52min 18s (SD = 14.88 min)
Test group	2h 19min (SD = 18.18 min)	48min 30s (SD = 20.54 min)
P-value	0.402	0.699

Table 7 : Time measures

Table 8 presents results on cognitive weight and perceived difficulty. The cognitive weight dimensions are detailed in Figure 16 where a point is a participant's score and a cross is the mean value for the dimension. We observe that the use of our demonstrator has no impact on the cognitive weight of participants. In the detailed representation of cognitive weight results, we can notice that mental and physical demands as well as the performance score are similar for the two groups. For temporal demand, we can clearly differentiate in both groups, participants with a low score (between 20 and 30) from participants with a high score (above 70). There are more participants with a high score in temporal demand in the test group. We can also notice that effort and frustration levels are slightly higher in the control group. In general, there is a large variability between participants on cognitive weight measures.

	Cognitive weight	Research difficulty	Modeling difficulty	Total difficulty
Control group	51.90 (SD = 11.96)	75 (SD = 12.90)	60.70 (SD = 19.90)	65.70 (SD = 12.40)
Test group	52.38 (SD = 13.11)	43.55 (SD = 19.75)	43.55 (SD = 23.40)	50 (SD = 13.80)
P-value	0.945	0.004	0.165	0.079

Table 8 : Cognitive weight and perceived difficulty measures

Figure 16 : detailed cognitive weight

We observe significant differences between the two groups on perceived difficulty. The difficulty of design rules research is more than 30 points higher without our demonstrator. With a P-value below 0.05, this result is statistically significant. Modeling difficulty does not shows any significant difference between the two groups. The total perceived difficulty shows a tendency toward a lower perceived difficulty in the test group. The average value is 15 points lower in the test group and the P-value is just above the 0.05 threshold.

4.4 Discussion

4.4.1 Results interpretation

We observe a clear improvement of design rules application in the test group. Despite design rules retrieval been slightly better in the test group with a better F-factor and more design rules selected, the difference is not significant enough to explain design rules application results. Several interpretations are possible:

- With the CACDA demonstrator, design rules research was easier for the test group. However, both groups spend the same amount of time in design rules retrieval and have equivalent cognitive weight for the overall design task. A better understanding of selected design rules or a more efficient design rules application in the CAD model may explain the test group better results in design rules application. Using the demonstrator, participants of the test group experienced less difficulty in design rules pre-selection, therefore sparing cognitive resources on this task in comparison with the control group. They were able to use these cognitive resources on design rules understanding and CAD modeling, therefore achieving better performances in design rules application.
- We can also explain this result by the difference in research types between groups. In fact, participants of the control group were able to perform a visual overview of the design rules PDF and select schemas that seemed to fit their design context. This kind of visual research is not possible with our demonstrator that selects design rules through their meaning and their links with contextual elements. The visual selection of design

rules may help in pure information retrieval but may cause a lack of understanding of those design rules.

We observe that the variability of results between participants is very high. In fact, participants can use many different approaches to design rules retrieval and CAD modeling. For example, several participants tried to get a full understanding of the design rules set before modeling, they achieved better results in design rules retrieval but sometimes lacked the time to complete their design. As standard deviations are lower in the test group for both design rules retrieval and application, we deduce that our demonstrator helps to reduce the impact of individual strategies. However, this is not enough to ensure a clear improvement in design rules retrieval.

Several measures do not lead to interpretable results. Time measurements show that participants used the full time at their disposal for the experiment and tried to balance their time between CAD modeling and design rules retrieval. The measure of the final volume of the part is similar in both groups and design errors can indistinctively lead to a decrease or an increase of the volume. Two design errors may even compensate their impact on the part's volume.

Overall, we conclude that our CACDA demonstrator has a positive impact on the design process. Design rules research is easier with our demonstrator with a perceived difficulty of design rules retrieval more than 30 points lower with the CACDA demonstrator on a scale of 100. Our results also show a tendency toward a lower difficulty of the overall design process with the demonstrator. Designers have better design rules application using the demonstrator, with a design rules application score twice higher in the test group than in the control group. This study demonstrates the usability of our CACDA demonstrator in an industrial context. We did not found any drawback caused by its usage, despite its novelty for participants. Several improvements are still necessary to reduce the impact of individual strategies, improve design rules recommendation and reduce the cognitive weight of the design task.

4.4.2 Identification of limitations and biases

The first limitation of our experiment is due to the low number of participants. Only strong impacts of our approach can be identified in this study. Moreover, despite their knowledge in design, our participants are not professional designers, which are our targeted end-users. These limitations are common in design experimentations with long and complex tasks to realize [32], [61]–[63]. Moreover only a small number of participants is required to highlight usability results [64].

Participants were asked to perform both an information retrieval and a design task over a period of 2 to 4 hours and with no prior knowledge of the documentation. In the industry, a designer may have a week or more to gradually become familiar with a design rules set. The fact that industrial sized design rules documentations are much larger than in our use-case compensates this limitation. Some participants of our study tried to acquire an overall understanding of the design rules set before beginning the CAD modeling. Such a strategy would not be possible with an industrial dataset. Moreover, we think that the use of the CACDA design rules recommendation system would be more impactful on a use-case with more design rules, as it would scale better than unstructured documentation. We need to demonstrate this point in future studies with a larger set of design rules.

Finally, our demonstrator is not yet a fully functional design software and several of the CACDA's features are still under development. A new test with a demonstrator featuring social and IT sub-contexts and taking advantage of contextual information in real time will be necessary to demonstrate the impact of the CACDA on the design process. The CACDA theoretically builds the social sub-context by recording users' interaction with the system and the set of design rules. We can therefore manually create a consistent social sub-context. Our participants stand as previous users of the CACDA whose interaction data has been captured and modeled in the knowledge graph.

5. Conclusion and future work

In this paper, we present a demonstrator of a Context-Aware Cognitive Design Assistant. The demonstrator performs design rules recommendations by reasoning on a knowledge graph that stores computable design rules and contextual knowledge. The demonstrator includes two sub-contexts out of the four planned in the CACDA's data model. To perform recommendations, the demonstrator relies on a random walk over the knowledge graph. This method is efficient and supports the exploration of multiple sub-contexts. As the CACDA recommendation system explores all sub-contexts simultaneously, it can recommend design rules as well as contextual elements of interest that the designer can use as contextual filters for future recommendations. We present the software architecture of the demonstrator and detail its implementation.

The second part of the paper is dedicated to design experimentations realized with the CACDA demonstrator. We placed fourteen participants in a design situation where they had to model an aeronautical part in a CAD environment while searching and applying design rules. We compared the use of our design assistant with traditional design rules manuals used in the industry. Our results show that designers successfully applied on average 8.6 design rules with the CACDA demonstrator, when designers using a PDF documentation applied only 4.3 design rules. The perceived difficulty of design rules retrieval is also significantly lower with our approach with a difficulty score of 75 out of 100 in the control and of 43.5 in the test group. This study also demonstrates the usability of the CACDA demonstrator in a design context, as it has been successfully used with no downside compared to PDF documentation.

In future works, we plan to develop a data link with a CAD environment in order to capture IT contextual information (CAD software, workbench, part, features, etc.) to influence the recommendation of design rules in near real time. Our goal is to improve design rules recommendations as well as to reduce the impact of designers' individual research strategies on the design process. We will also use our experimental results to implement a social sub-context and improve recommendations further more. Indeed by capturing the activities of designers during past experiments, we will be able to recommend design rules to designers with similar profile. Experiments with this future version of the demonstrator will feature a panel of professional designers and an improved use-case with larger set of design rules.

6. References

[1] D. WALDEN, G. ROEDLER, K. FORSBERG, D. HAMELIN, and T. SHORTELL, *Systems engineering handnook: a guide for system life cycle processes and activities*. Hoboken: John Wiley & Sons, 2015.

[2] F. Segonds, G. Cohen, P. Véron, and J. Peyceré, “PLM and early stages collaboration in interactive design, a case study in the glass industry,” *Int. J. Interact. Des. Manuf.*, vol. 10, no. 2, pp. 95–104, 2016, doi: 10.1007/s12008-014-0217-4.

[3] R. B. Handfield, S. A. Melnyk, R. J. Calantone, and S. Curkovic, “Integrating environmental concerns into the design process: The gap between theory and practice,” *IEEE Trans. Eng. Manag.*, vol. 48, no. 2, pp. 189–208, 2001, doi: 10.1109/17.922478.

[4] K. Goher, E. Shehab, and A. Al-Ashaab, “Model-Based Definition and Enterprise: State-of-the-art and future trends,” *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, 2020, doi: 10.1177/0954405420971087.

[5] A. Huet *et al.*, “Design rules application in manufacturing industries: a state of the art survey and proposal of a context-aware approach,” in *proceedings - International Joint Conference on Mechanics, Design Engineering and Advanced Manufacturing*, 2020, p. 6.

[6] A. Huet, R. Pinquié, P. Véron, A. Mallet, and F. Segonds, “CACDA : A knowledge graph for a context-aware cognitive design assistant,” *Comput. Ind.*, vol. 125, 2021, doi: 10.1016/j.compind.2020.103377.

[7] G. Tecuci, D. Marcu, M. Boicu, and D. A. Schum, *Knowledge engineering: building cognitive assistant for evidence-based reasoning*, Cambridge. 2016.

[8] C. González-Lluch, P. Company, M. Contero, J. D. Camba, and R. Plumed, “A survey on 3D CAD model quality assurance and testing tools,” *CAD Comput. Aided Des.*, vol. 83, pp. 64–79, 2017, doi: 10.1016/j.cad.2016.10.003.

[9] “DFMPro.” [Online]. Available: https://dfmpro.geometricglobal.com/.

[10] “Siemens NX Checkmate.” [Online]. Available: https://www.plm.automation.siemens.com/en_us/Images/2504_tcm1023-11882.pdf.

[11] B. Huang, C. Xu, R. Huang, and S. Zhang, “An automatic 3D CAD model errors detection method of aircraft structural part for NC machining,” *J. Comput. Des. Eng.*, vol. 2, no. 4, pp. 253–260, 2015, doi: 10.1016/j.jcde.2015.06.008.

[12] A. Elkaseer, T. Mueller, A. Charles, and S. Scholz, “Digital Detection and Correction of Errors in As-built Parts: a Step Towards Automated Quality Control of Additive Manufacturing,” pp. 389–392, 2018, doi: 10.3850/978-981-11-2728-1_58.

[13] A. Rangarajan, P. Radhakrishnan, A. Moitra, A. Crapo, and D. Robinson, “Manufacturability Analysis and Design Feedback System Developed Using Semantic Framework,” pp. 1–11, 2013.

[14] M. Dinar and D. W. Rosen, “A design for additive manufacturing ontology,” *J. Comput. Inf. Sci. Eng.*, vol. 17, no. 2, pp. 1–9, 2017, doi: 10.1115/1.4035787.

[15] Q. Wang and X. Yu, “Ontology based automatic feature recognition framework,” *Computers in Industry*, vol. 65, no. 7. pp. 1041–1052, 2014, doi:

10.1016/j.compind.2014.04.004.

[16] J. Leal, V. Rodrigues, and R. Queirós, "Computing semantic relatedness using DBPedia," *Symp. Lang. Appl. Technol. 1st*, no. January 2014, pp. 133–147, 2012, doi: 10.4230/OASIcs.SLATE.2012.i.

[17] A. Boufrida and Z. Boufaida, "Rule extraction from scientific texts: Evaluation in the specialty of gynecology," *J. King Saud Univ. - Comput. Inf. Sci.*, 2020, doi: 10.1016/j.jksuci.2020.05.008.

[18] X. Ye and Y. Lu, "Automatic extraction of engineering rules from unstructured text: A natural language processing approach," *J. Comput. Inf. Sci. Eng.*, vol. 20, no. 3, pp. 1–4, 2020, doi: 10.1115/1.4046333.

[19] S. Kang *et al.*, "Ontology-Based Ambiguity Resolution of Manufacturing Text for Formal Rule Extraction," *J. Comput. Inf. Sci. Eng.*, vol. 19, no. 2, pp. 1–9, 2019, doi: 10.1115/1.4042104.

[20] J. Luttmer, D. Ehring, R. Pluhnau, and A. Nagarajah, "Representation and Application of Digital Standards Using Knowledge Graphs," *Proc. Des. Soc.*, vol. 1, no. AUGUST, pp. 2551–2560, 2021, doi: 10.1017/pds.2021.516.

[21] W. Yang, C. Fu, X. Yan, and Z. Chen, "A knowledge-based system for quality analysis in model-based design," *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1579–1606, 2020, doi: 10.1007/s10845-020-01535-8.

[22] W. Yang, Q. Zhao, X. Yan, and Z. Chen, "A system framework of model quality analysis for product model in collaborative manufacturing," *Int. J. Adv. Manuf. Technol.*, 2021, doi: https://doi.org/10.1007/s00170-021-07622-1.

[23] C. González-Lluch, P. Company, M. Contero, J. D. Camba, and J. Colom, “A case study on the use of model quality testing tools for the assessment of MCAD models and drawings,” *Int. J. Eng. Educ.*, vol. 33, no. 5, pp. 1643–1653, 2017.

[24] S. G. Kim, S. M. Yoon, M. Yang, J. Choi, H. Akay, and E. Burnell, “AI for design: Virtual design assistant,” *CIRP Ann.*, vol. 68, no. 1, pp. 141–144, 2019, doi: 10.1016/j.cirp.2019.03.024.

[25] S. van Engelenburg, M. Janssen, and B. Klievink, “Designing context-aware systems: A method for understanding and analysing context in practice,” *J. Log. Algebr. Methods Program.*, vol. 103, pp. 79–104, Feb. 2019, doi: 10.1016/j.jlamp.2018.11.003.

[26] A. K. Dey, “Understanding and using context,” *Pers. ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, 2001, doi: 10.1016/j.healthplace.2012.01.006.

[27] M. Melucci, “A basis for information retrieval in context,” *ACM Trans. Inf. Syst.*, vol. 26, no. 3, 2008, doi: 10.1145/1361684.1361687.

[28] I. Ruthven, “Information retrieval in context,” in *Advanced Topics in Information Retrieval. The Information Retrieval Series*, Springer., M. Melucci and R. Baeza-Yates, Eds. Berlin, Heidelberg., 2011.

[29] D. Dupont, J. L. V. Barbosa, and B. M. Alves, “CHSPAM: a multi-domain model for sequential pattern discovery and monitoring in contexts histories,” *Pattern Anal. Appl.*, vol. 23, no. 2, pp. 725–734, 2020, doi: 10.1007/s10044-019-00829-9.

[30] J. H. Rosa, J. L. V. Barbosa, M. Kich, and L. Brito, “A Multi-Temporal Context-aware System for Competences Management,” *Int. J. Artif. Intell. Educ.*, vol. 25, no. 4,

pp. 455–492, 2015, doi: 10.1007/s40593-015-0047-y.

[31] A. S. Filippetto, R. Lima, and J. L. V. Barbosa, "A risk prediction model for software project management based on similarity analysis of context histories," *Inf. Softw. Technol.*, vol. 131, no. December 2020, 2021, doi: 10.1016/j.infsof.2020.106497.

[32] M. Peruzzini and M. Pellicciari, "A framework to design a human-centred adaptive manufacturing system for aging workers," *Adv. Eng. Informatics*, vol. 33, pp. 330–349, 2017, doi: 10.1016/j.aei.2017.02.003.

[33] J. Zhu, S. K. Ong, and A. Y. C. Nee, "A context-aware augmented reality system to assist the maintenance operators," *Int. J. Interact. Des. Manuf.*, vol. 8, no. 4, pp. 293–304, 2014, doi: 10.1007/s12008-013-0199-7.

[34] K. Alexopoulos, S. Makris, V. Xanthakis, K. Sipsas, and G. Chryssolouris, "A concept for context-aware computing in manufacturing: the white goods case," *Int. J. Comput. Integr. Manuf.*, vol. 29, no. 8, pp. 839–849, 2016, doi: 10.1080/0951192X.2015.1130257.

[35] J. H. Da Rosa, J. L. V. Barbosa, and G. D. Ribeiro, "ORACON: An adaptive model for context prediction," *Expert Syst. Appl.*, vol. 45, pp. 56–70, 2016, doi: 10.1016/j.eswa.2015.09.016.

[36] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," *arXiv*, pp. 1–27, 2020, doi: 10.1109/TNNLS.2021.3070843.

[37] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L. K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," *RecSys 2018 - 12th ACM Conf.*

Recomm. Syst., pp. 297–305, 2018, doi: 10.1145/3240323.3240361.

[38] A. Kofod-Petersen and A. Aamodt, “Case-based situation assessment in a mobile context-aware system,” *AIMS ’05 Artif. Intell. Mob. Syst. 2005*, pp. 41–49, 2003.

[39] V. Bajaj, R. B. Panda, C. Dabas, and P. Kaur, “Graph Database for Recipe Recommendations,” *2018 7th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2018*, pp. 276–281, 2018, doi: 10.1109/ICRITO.2018.8748827.

[40] R. Angles, “A Comparison of Current Graph Database Models,” in *IEEE 28th International Conference on Data Engineering Workshops*, pp. 171–177, doi: 10.1109/ICDEW.2012.31.

[41] R. Angles, “The property graph database model,” in *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*, 2018, vol. 2100, no. Section 2.

[42] R. Pinquié, P. Véron, F. Segonds, and T. Zynda, “A Property Graph Data Model for a Context-Aware Design Assistant,” in *IFIP International Conference on Product Lifecycle Management*, 2019, pp. 181–190, doi: 10.1007/978-3-030-42250-9_17.

[43] K. Arora, “Recommendation Systems: a Review Report,” *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 7, pp. 503–507, 2017, doi: 10.26483/ijarcs.v8i7.4308.

[44] D. Asanov, *Algorithms and Methods in Recommender Systems*. Berlin, 2011.

[45] X. Cai *et al.*, “Learning collaborative filtering and its application to people to people recommendation in social networks,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 743–748, 2010, doi: 10.1109/ICDM.2010.159.

[46] V. Subramaniyaswamy, R. Logesh, M. Chandrashekhar, A. Challa, and V.

Vijayakumar, “A personalised movie recommendation system based on collaborative filtering,” *Int. J. High Perform. Comput. Netw.*, vol. 10, no. 1–2, pp. 54–63, 2017, doi: 10.1504/IJHPCN.2017.083199.

[47] S. Sawant, “Collaborative Filtering using Weighted BiPartite Graph Projection - A Recommendation System for Yelp,” *Proc. CS224W Soc. Inf. Netw. Anal. Conf.*, 2013.

[48] Z. Huang, X. Lu, and H. Duan, “Context-aware recommendation using rough set model and collaborative filtering,” *Artif. Intell. Rev.*, vol. 35, no. 1, pp. 85–99, 2011, doi: 10.1007/s10462-010-9185-7.

[49] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Adv. Artif. Intell.*, vol. 2009, no. Section 3, pp. 1–19, 2009, doi: 10.1155/2009/421425.

[50] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web. Lecture Notes in Computer Science*, Springer., vol. 4321 LNCS, N. W. Brusilovsky P., Kobsa A., Ed. Berlin, Heidelberg., 2007, pp. 325–341.

[51] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic Cosine Similarity,” *Semant. Sch.*, vol. 2, no. 4, pp. 4–5, 2012.

[52] M. Dehmer, “Structural analysis of complex networks,” *Struct. Anal. Complex Networks*, pp. 1–486, 2011, doi: 10.1007/978-0-8176-4789-6.

[53] J. W. G. Putra and T. Tokunaga, “Evaluating text coherence based on semantic similarity graph,” *Proc. TextGraphs@ACL 2017 11th Work. Graph-Based Methods Nat. Lang. Process.*, pp. 76–85, 2020, doi: 10.18653/v1/w17-2410.

[54] M. A. Alvarez and S. J. Lim, "A graph modeling of semantic similarity between words," *ICSC 2007 Int. Conf. Semant. Comput.*, pp. 355–362, 2007, doi: 10.1109/ICSC.2007.23.

[55] I. Gurevych, "Using the structure of a conceptual network in computing semantic relatedness," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3651 LNAI, pp. 767–778, 2005, doi: 10.1007/11562214_67.

[56] L. Strobin and A. Niewiadomski, "Recommendations and object discovery in graph databases using path semantic analysis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8467 LNAI, no. PART 1, pp. 793–804, 2014, doi: 10.1007/978-3-319-07173-2_68.

[57] T. Bogers, "Movie Recommendation using Random Walks over the Contextual Graph," *Proc. 2nd Work. Context. Recomm. Syst. ACM*, no. 2010, 2010.

[58] E. Hjelseth and N. Nisbet, "Overview of concepts for model checking," *Proc. CIB W78 2010 27th Int. Conf. –Cairo, Egypt*, no. September, pp. 16–18, 2010.

[59] S. G. Hart, "NASA-task load index (NASA-TLX); 20 years later," *Proc. Hum. Factors Ergon. Soc.*, pp. 904–908, 2006, doi: 10.1177/154193120605000909.

[60] C. Nikulin, G. Lopez, E. Piñonez, L. Gonzalez, and P. Zapata, "NASA-TLX for predictability and measurability of instructional design models: case study in design methods," *Educ. Technol. Res. Dev.*, vol. 67, no. 2, pp. 467–493, 2019, doi: 10.1007/s11423-019-09657-4.

[61] S. J. Grigg, S. K. Garrett, and L. C. Benson, "Using the NASA-TLX to assess first year

engineering problem difficulty," *62nd IIE Annu. Conf. Expo 2012*, pp. 2793–2802, 2012.

[62] S. Khan and M. J. Awan, "A generative design technique for exploring shape variations," *Adv. Eng. Informatics*, vol. 38, no. March, pp. 712–724, 2018, doi: 10.1016/j.aei.2018.10.005.

[63] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski, "Design study of linesets, a novel set visualization technique," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2259–2267, 2011, doi: 10.1109/TVCG.2011.186.

[64] R. Alroobaea and P. J. Mayhew, "How many participants are really enough for usability studies?," *Proc. 2014 Sci. Inf. Conf. SAI 2014*, pp. 48–56, 2014, doi: 10.1109/SAI.2014.6918171.