Soft Computing Letters 3 (2021) 100025

Contents lists available at ScienceDirect

Soft Computing Letters

journal homepage: www.sciencedirect.com/journal/soft-computing-letters

Fuzzy engineering design semantics elaboration and application

Alain-Jérôme Fougères [a,*], Egon Ostrosi [b]

[a] Laboratory of Informatics and Telecommunications, ECAM Rennes, Campus de Ker Lann - Bruz, Rennes 35091, France
[b] University of Bourgogne Franche-Comté, UTBM, ERCOS/ELLIAD EA4661, Belfort 90010, France

ARTICLE INFO

Keywords:
Fuzzy collaborative design
Fuzzy collaborative system
Natural language processing
Fuzzy requirement engineering
Fuzzy engineering design platform

ABSTRACT

Product design activities are predicated on fuzzy modelling, given that verbalising and interpreting engineering requirements are inherently fuzzy processes. The aim of this paper is to present a method for fuzzy intelligent requirement engineering from natural language to Computer-Aided Design (CAD) models. The field exploring the dynamics of computational processes from fuzzy linguistic modelling to fuzzy design modelling is complex and remains under-explored. No existing research has been identified which focuses specifically on fuzzy requirements engineering from natural language to CAD modelling. This paper seeks to address this by providing a design formalisation system based on five key principles. These principles are used to set out a computing procedure which follows a method broken up into six phases. The results of these six phases are fuzzy semantic graphs, which provide engineering requirements according to reliable design information. The approach is put into practice using the fuzzy agent-based tool developed by the authors, called F-EGEON (Fuzzy Engineering desiGn sEmantics elabOration and applicatioN). The proposed method is illustrated through an application from the automotive industry.

1. Introduction

The concept of possibility plays a central role in human decision-making and underlines much of human ability to reason in approximate terms [1]. Design is a domain of possibility. A major issue in design is the meaning of information. In design, what matters is the ability to create and chain information, and to answer questions relating to information. The proper framework for information analysis is by nature possibilistic rather than probabilistic [1]. Much of the information on which designers base decisions is also possibilistic by nature.

The design process from conceptualization through to representation involves several different forms of knowing. Cognitive models can be presented in words through the form of linguistic modelling, which is based primarily on natural language use. As a result of the subjective nature of natural language use, difficulties may arise in the context of collaborative design, given that there may be subtle variations between what each designer understands a word to mean.

A significant component of design is, in effect, the making of meaning. Although words tend to have established definitions, the boundaries of their meaning often remain indistinct and ill-defined. When asked to explain what a word means, people may not be willing to provide an explanation, or else may offer an account which differs from their actual understanding of the word [2]. What is more, many words have imprecise or changeable meanings. As a result, a meaning that is clear to one designer may be fuzzy to another, even if their experience appears to be shared from the outside. Hard conceptual boundaries in word semantics may be achievable using topological and dynamical concepts, as discussed by French mathematician Thom [3,4]. Because the semantics of words is partially governed by underlying perceptual mechanisms and by cultural conventions [5], the natural language inherently incorporates myriad possible meanings, with the resulting fuzzy boundaries to its form [1,6].

In the context of engineering design, cognitive modelling can take the external form of a drawing, plan, prototype or 3D model of the engineering object in question. Cognitive modelling therefore forms one of the languages which designers use to develop their designs. Language in the design context provides an expression of creativity, and shapes the courses of action available to the designer. Models form many uses to designers, and may constitute an aid for thinking or decision making, a prompt for discussion, or a way of assessing how reliable their proposals are.

The different potential ways of arranging and distributing structures form a key component of engineering design, particularly in the initial phases of the design process. New structures with novel parameters are

* Corresponding author.
E-mail address: alain-jerome.fougeres@ecam-rennes.fr (A.-J. Fougères).

https://doi.org/10.1016/j.socl.2021.100025
Received 31 August 2021; Received in revised form 16 October 2021; Accepted 31 October 2021
Available online 2 November 2021

produced, which may not be recognized by the earlier ones. This process is an important aspect of design, given that it increases the potential for creativity and discovering alternative approaches. Assisting the Computer-Aided Design (CAD) with natural language processing (NLP) means expanding the creativity process from engineering requirements. The following research questions can be then addressed:

- How can the natural language requirements of a product best be formalized, whilst taking fuzziness into account?
- How can we most effectively identify and extract the necessary linguistic information from the product design requirements in order to create formal or diagrammatic models which take into account the possibilistic nature?
- How can the fuzzy language of requirements be formalized into a format appropriate for CAD modelling?

Fuzzy set theory offers a natural foundation for the theory of possibility. Fuzzy sets and their associated membership functions play key roles in developing formal models. Natural language describes uncertain knowledge, which provides a label to specify a fuzzy set. The knowledge is subsequently translated into membership functions, which provide a semantic manifestation of the designer's knowledge. Membership functions provide suitable representations which are then retranslated into language.

From this perspective, fuzzy variables are associated with possibility distributions. In this way, a fuzzy restriction's membership function, interpreted as a possibility distribution function, acts as a possibility distribution function.

The primary aim of this paper is to propose a method for fuzzy intelligent requirements engineering from natural language to CAD models. The paper is organized as follows: section 2 presents a survey of the related work, relevant research questions, and the open issue; section 3 presents a fuzzy requirements engineering method, consisting of six phases; section 4 illustrates this method by presenting a case study on the design of a car door hinge; and section 5 provides the conclusions as well as a discussion of the study's implications for future research.

2. The current research landscape

The research questions this paper presents have tended to be addressed by research into requirement engineering aided by NLP [7]. Over the past thirty years, the research theme centring on software requirement engineering supported by NLP has developed, and involves a variety of different approaches [8–19].

Significantly less research has taken place which focuses on the application of NLP to design specification. Recent works include the following: an approach focusing on improving CAD/CAM's design efficiency and ease of use [20]; the application of NLP techniques to automatically produce parametric property-based requirements from both semi-structured and unstructured specifications [21]; a NLP approach which identifies ambiguous terms from different domains, and ranks them according to their ambiguity score [22]; the use of NLP to automatically implement changes in construction models [23].

Engineering design or requirements assistance can be carried out through semantic analysis and representing texts or verbalisations, making use of existing linguistic and conceptual knowledge. This is equally true of translating technical texts into formal or semi-formal diagrams and employs knowledge of specific areas which have previously been conceptualised. The underlying motive of this is to offer increased assistance for conceptualisation than an automatic design could offer. In the following example, a steadfastly symbolic approach is employed. The approach is guided by the hypothesis of representations with AI formalisms using the analogy of mental representations (for alternative approaches, including deep learning and neural networks, refer to [24–27]). Understanding is associated with constructing a series of formal representations, which rely to varying degrees on a set of statements, which are subsequently translated into computational processes. This relies on the design specifications having the following characteristics:

- Due to the need to be understood by multiple designers, design texts or verbalisations are generally required to be concise and unambiguous. The use of precise language demonstrates the designers' readiness to work collaboratively and be mutually understood, whilst also ensuring that the engineering specifications remain relevant [28].
- The design specifications are in essence descriptions of events, objects, actions, contexts or situations. To be interpreted, linguistic universals can be employed, including the distinction between a verb and a noun, between action and relation, and between objects. To analyse discourse propositionally, a basic typology of three verbal categories is generally suggested: (1) the stative, (2) the factive, and (3) declarative verbs. Regarding characterising objects, five voices can be differentiated: 1) the existential, (2) the situative, (3) the equative, (4) the descriptive, and (5) the subjective. Assistance for effective design initially seeks to identify, and then to represent, the functional, structural, and dynamic aspects within the sentences which express the engineering requirements. In this way, grammars which allow a proposition's category (event, action, state), and a predicate's arguments (object, agent, source etc.) to be determined, are well suited for these processes [29].
- The lexicon and the sentence [30] are the units which are of primary importance for linguistically processing the brief, precise and unambiguous design expressions. This automatic comprehension is therefore facilitated, primarily through computing sentence meanings from words (referred to as compositionality) and by employing a knowledge representation formalism attained from AI, which allows meanings to be partially represented.
- According to Vygotsky [31], the internal language of thought is translated into an external language (spoken or written word) through the process of speaking or writing. Understanding therefore proceeds symmetrically. When an individual is asked to describe an activity they have expertise in, they are in essence being asked to translate their internal thoughts about this activity into an external language. To ensure that the description is understood, they will attempt to translate their thoughts directly into an abbreviated language which employs a simplified form of syntax and exclusively predicative judgements. Understanding is thus conceptualising.

Product design is an activity of fuzzy linguistic modelling and fuzzy conceptual modelling because it is characterized by verbalization/ formulation of engineering requirements. Product design usually starts with the identification of a need, proceeds through a sequence of activities to seek for a solution to the problem and ends with a detailed description of the product or the technical system. There have been many attempts to draw up models to handle the complexity management of design process in systematic steps [32–36]. Both functional modelling and structural modelling have been investigated [37–41]. There are many different interpretations of the notion of the function in engineering design [42,43]. Usually, functions can be defined as the abstracted behaviour [44,45] represented in natural language [33,46]. The point is natural language cannot represent unequivocally the designers' thinking. Uncertainty is thus an integral part of the design process.

During the design process, the designer deals with some distinct forms of uncertainty such as incompleteness, imprecision, randomness, fuzziness, and ambiguity [47]. Therefore, it is unrealistic to model, evaluate and forecast without considering this inherent uncertainty and imprecision [48–50]. Formalization based on fuzzy set theory makes it possible to develop flexible and possibilistic models. Basically, product design means building and maintaining a coherent network from fuzzy requirements to CAD models [51]. Fuzzy functional architecture, fuzzy logical architecture, fuzzy generative architecture that produces a variety of different configurations, and fuzzy physical architecture, are just some of the fundamental elements proposed by the fuzzy network of product design [52]. Formalization based on fuzzy set theory makes it possible to compute a fuzzy set of consensual solutions based on customer requirements and service engineering constraints [47], a fuzzy optimal product configuration [53], and a fuzzy nucleus for product configuration [54]. A fuzzy network of product design enables the construction of platforms for design configurations [55,56]. In this context, multidisciplinary stakeholders, from customers to the designers including different engineering services, must acknowledge the uncertainty and fuzziness in order to ultimately achieve relevant communication and chaining of engineering requirements. These engineering requirements should be computed, represented and memorized in digital documents or diagrams. Consequently, any soft computational procedure facilitating building design requirements contributes to reducing the time taken to realize not only the requirements' elaboration, but also the other design phases. Nonetheless, while there has been much research on the application of requirements engineering, none has focused specifically on fuzzy requirements engineering from natural language to CAD modelling. Thus, it is important to gain a coherent and more complete picture of requirement engineering and its changes by considering the possibilistic nature of design formalized by fuzzy set theory. This paper addresses this issue.

3. Fuzzy modelling of requirements engineering

As the dynamics of computational processes from fuzzy linguistic modelling, fuzzy cognitive modelling to fuzzy design modelling constitute in themselves a difficult and not fully explored domain, we propose five principles that guide a new approach to engineering requirements and allow the development of a design formalization system. These principles, which enrich those proposed in [29,57], are the following:

- Principle 1: Linguistic modelling, cognitive modelling and design modelling are fuzzy. This principle enhances the use of fuzzy set theory to model uncertainty, imprecision and the possibilistic nature of natural language, and to compute design requirements. This principle becomes even stronger when considering developing requirements in a cooperative context where each designer will have a personal perception of the different formulated language elements. The fuzzy logic concepts for fuzzy requirements modelling and computing are presented in the next section.
- Principle 2: Design requirements from natural language should be refined to obtain formal design requirements. It is not easy to directly translate an informal design requirement into a formal one. This difficulty corresponds to the problem of translating a natural language into a formal language. Observing co-design activity corroborates this principle. During the activity, a product design is gradually elaborated by refinement [58].
- Principle 3: The move from the informal requirements to the formal ones needs an intermediate representation. Refining the expression of a design requirement is a process not sufficiently defined to determine the real number of refinement steps necessary for a given requirement. Thus, we proposed an intermediate representation, referred to as the pivot of the formalization. This intermediate representation might be in the form of conceptual graphs [59] or fuzzy conceptual graphs that extend conceptual graphs to fuzzy values [60, 61]
- Principle 4: The formal representation of fuzzy requirements should maximize the content of information. The automatic processes of formal representation produce requirements which are too close to natural language (literal semantics). In addition, it is often necessary to transform them using experiences rules in order to make them relevant and usable to designers. The major cause of the distance between initial requirements and the formal requirements produced is the overuse of high-level thematic relations (agent, object, source, destination, etc.) that might express the meaning of sentences but are unimportant for technical requirements.
- Principle 5: A formal intermediate representation of fuzzy requirements serves as a pivot for translation into different design requirement languages. This principle stems from a common observation among designers: a single language rarely lends itself to modelling the complete design requirement (which is the origin of the multi-requirement paradigm [62]). It is then possible to translate the logical description's content into the target formal design language, provided that the proper translator is available.

3.1. Fuzzy set and fuzzy logic concepts

In set theory, a set has elements that satisfy all of its specific properties. That is to say that elements that do not satisfy all the properties of a set cannot belong to this set. Thus, a subset A of a set X can be described from its characteristic function $x : X \rightarrow \{0, 1\}$ as follows (Eq. (1)):

$$x_A(x) = \begin{cases} 1 \; if x \varepsilon A \\ 0 else \end{cases} \quad (1)$$

However, many subsets expressed in common language cannot be defined by a specific/discriminant property: for example, the subset "Large Opening" of set "Opening", in a car door context. It is then possible to define a characteristic function of membership of a subset of A, denoted μ_A, which associates each element x of X with a real value $\mu_A(x)$ in the interval [0, 1]. This membership function allows, on the one hand, highlighting grades of membership of elements of the set X, and on other hand, the reference to define a fuzzy subset of X. Fuzzy subsets use the same operations as classical subsets: equality, inclusion, union, intersection, complement, etc. Thus, membership function is defined for a fuzzy relation R between two universes of reference X and Y as follows (Eq. (2)):

$$f_R : X \times Y \rightarrow [0, 1] \quad (2)$$

The fuzzy logic knowledge representation [63] is based on fuzzy elementary propositions such as "V is A", defined from a set L of variables (V, X, T_V), where V is a linguistic variable, X is the universe of discourse (set of values), and T_V is a list of characterizations (linguistic values) of V represented by fuzzy subsets of X. For instance, let us consider the opening V, and the list $T_V = \{A_1, A_2, A_3, A_4, A_5\}$, where A_1 = "Very small", A_2 = "Small", A_3 = "Medium", A_4 = "Large", and A_5 = "Very large".

A fuzzy proposition is formed from a fuzzy subset A of T_V or a modified form thereof (for instance, weakening or strengthening). This fuzzy proposition is provided by the membership function of the fuzzy set μ_A and its truth value belongs to any set [0, 1] (i.e., the fuzzy proposition is especially true for any value x of X that $\mu_A(x)$ is high). A truth value equal to 0 or 1 corresponds to a proposition that is absolutely false, or absolutely true respectively (as defined in set theory).

Zadeh [6] has proposed fuzzy logic as a framework for approximate reasoning. This approximate or fuzzy deductive reasoning is an extension of the reasoning in classical logic. Thus, the three basic operators of classical logic, conjunction, disjunction and negation, are also defined for fuzzy logic. Consider the fuzzy propositions P1 = "V is A" and P2 = "W is B", then:

(1) the conjunction of P1 and P2 is a fuzzy proposition whose truth value $\mu_{A \wedge B}$ is obtained by aggregation using a t-norm of truth values of the two propositions (for example using the min operator, $\mu_{A \wedge B}(x) = \min(\mu_A(x), \mu_B(x))$.
(2) the disjunction of P1 and P2 is a fuzzy proposition whose truth value $\mu_{A \vee B}$ is obtained by aggregation using a t-conorm of truth

values of the two propositions (for example using the max operator, $\mu_{A \vee B}(x) = \max(\mu_A(x), \mu_B(x))$.

(3) the negation of P1 is a fuzzy proposition whose truth value is $\neg\mu_A$ (for example using the complement to 1 operator, $\mu_A(x) = 1 - \mu_A(x)$).

Fuzzy implication rules, such as "IF V is A THEN W is B", can be used to represent knowledge about a system. A fuzzy rule defines on $X \times Y$ a relation R, that is noted A→B, between the values taken by V and W. The relation R determines the bonding strength between the premise "V is A" and the conclusion "W is B". Its membership function $\mu_{A \to B}$ corresponds to the truth value of the fuzzy implication between the two propositions (premise and conclusion). Many forms or operators of fuzzy implication have been proposed (generally derived from work on multivalued logic). For instance, Mamdani [64] proposes seeing implication as a conjunctive relation (Eq. (3)):

$$\mu_{A \to B}(x, y) = \min(\mu_A(x), \mu_B(y),) \tag{3}$$

For implementing fuzzy set theory in fuzzy requirements modelling and computing in product design, we proceed in the following three phases:

(1) defining a universe of discourse for a design problem, named U, made up of a set of domains representing linguistic variables (ex: {Access, Rotate, Solidarize, Open, Close, …}).

(2) defining fuzzy subsets of these domains to represent fuzzy linguistic values (ex: {Weak, Average, Strong, …}); and finally.

(3) defining membership functions, named $\mu_i \dots \mu_j$, allowing determination of the degree of membership of an instance of a linguistic variable stated by a designer, to one (or more) linguistic values (ex: let x be an instance of "open" and A the fuzzy subset corresponding to the linguistic value "Strong" then $\mu_A(x) \in [0,1]$).

The introduction of fuzzy sets and membership functions to these fuzzy sets allows fuzzy associations, inferences and reasoning to be made using the concepts of fuzzy logic.

3.2. Fuzzy requirements engineering approach

According to these five principles, we proposed a 5-phase method to model intelligent requirements expressed in natural language and their chaining for CAD models (Fig. 1).

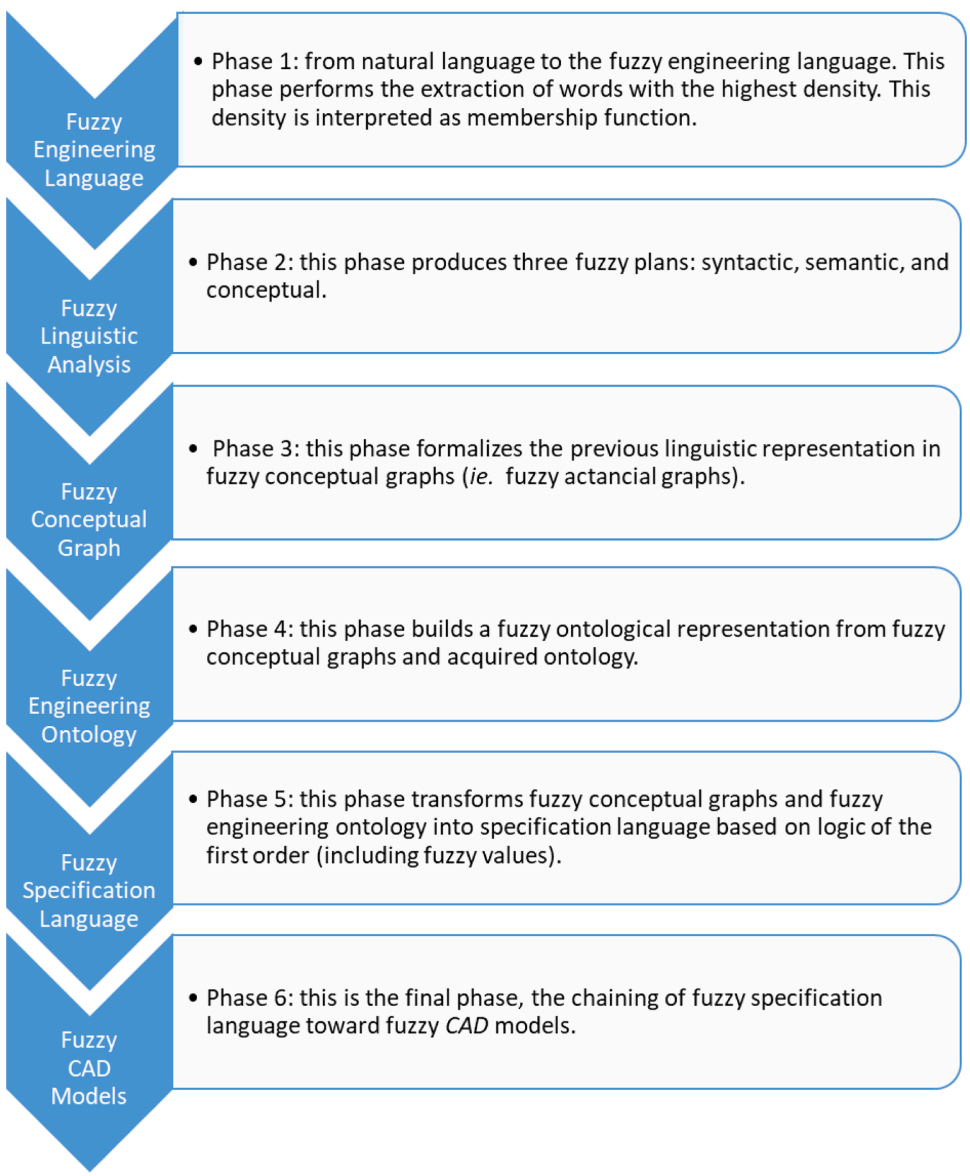

Fig. 1. 6-phases method proposed for building fuzzy engineering specifications.

3.2.1. Phase 1: building fuzzy engineering ontology

The phase of building fuzzy engineering ontology consists of extracting the lexical information contained in the specifications and determining the privileged links between the words [65]. Extracting lexical information can be performed in two ways. Firstly, it may be automatically selected from texts or speech from engineering domains using co-occurrence studies of words, based on lexical proximity analysis and statistical filtering techniques, such as mutual information [66]. Many NLP technics and tools have been proposed to help the building of terminologies and to acquire/learn ontologies from corpora [65,66]. Secondly, it may be semi-automatically extracted, as we presented in [67] and that we propose in the context of this work.

The method proposed for obtaining knowledge entails extracting meanings of previously recognized concepts from a digital dictionary and then describing them in a semantic dictionary, for example, using conceptual graphs (CG or fuzzy CG). The treatment's purpose is to grow the application domain's knowledge base by integrating semantic information from a dictionary and expanding the model's ontology through the definitions of each concept. It is feasible to construct a corresponding fuzzy CG and incorporate it in a canonical base after analyzing the contents of a definition.

3.2.2. Phase 2: formulate fuzzy requirements engineering in natural language

Understanding a text or a speech of design specification involves identifying three elements: (1) the static situations of spatio-temporal location and characterization of objects; (2) the kinematic situations of moves in a space or of state changes attributed to objects; and (3) dynamic situations of moves or changes of states caused by an external force. There are three levels of representation for these elements (Fig. 2): (1) the cognitive, generated from cognitive archetypes; (2) the conceptual, organized into predicates and arguments; and (3) the linguistic, organized from grammatical schemas specific to a language [68]. Thus, to state a description is to integrate cognitive archetypes into predicative conceptual schemes and to encode these predicative structures into specific linguistic systems in the form of grammatical schemas specific to the language used (nouns, verbs, prepositions, etc.) [31].

3.2.3. Phase 3: linguistic analysis of fuzzy engineering specification

The language has three fundamental properties [69]: it is intentional (for example, the intention to transmit information), syntactic (for example, the structure and coherence of a speech act), and enunciative (for example, transmitting information). Thus, the aims of NLP are to propose methods, techniques and tools that cover these three properties. In computational linguistics, five levels for written or spoken language are considered (morphology, lexicon, syntax, semantics, and pragmatics). As we will see in the next section, the semantic level is of primary interest in our representation of specifications. This level concerns identifying the meaning of words and sentences, as well as the relationships they establish between them. There are several approaches to this challenge, for example: lexical semantics, based on primitives and relations between elements, such as Fillmore's case grammars [70–72], Schank's action primitives [73], Desclés's cognitive archetypes [68], Pottier's general semantic [74], STDP (Stanford Typed Dependency Parser) [75], notions of prototypes and mental models; or grammatical semantics, such as Montaguë's semantics, DRT [76]. As for the relations between semantics and automatic treatments, they are amply described in [77].

3.2.4. Phase 4: building fuzzy conceptual graphs

Representing knowledge aims to allow a correct representation through a sufficiently precise and formal notation or representational framework. Considering conceptual graphs model [59] a formalism of the family of semantic networks, elementary objects are concepts or relations, and propositions or facts are represented by a graph connecting concepts and relations by directed arcs. Rules are defined offering the possibility of joining or dissociating CGs (joints and projections). An isomorphism with the first-order predicate logic is established for a basic kernel. Thus, the CGs' formalism is both a conceptual model in its form (great readability), and a system with axiomatic foundations.

Moreover, by using fuzzy conceptual graphs [60,61] it is possible to represent the uncertainty (probability) and especially the imprecision (subjectivity) of the proposals made by the different designers collaborating in a design activity. The transition from linguistic analysis to the representation of extracted knowledge therefore consists in the semantic translation of the syntactic/semantic structure in the form of fuzzy CGs (see the algorithm presented in Fig. 3). For instance, case grammars are used to determine the different thematic roles filled by the constituents of a sentence using information acquired on the order of words, prepositions, verbs, and context. In other words, the parser determines how the noun phrases of a sentence are related to verbs - the semantic role that specifies how an object participates in the description of an action.

3.2.5. Phase 5: transforming the fuzzy specification language

In [59], Sowa defined the operator Φ which makes a formula in the predicate logic of the first order correspond to every CG. Moreover, an interpretation of the CG model extended to fuzzy values has been proposed by Thomopoulos et al. [78]. For instance, the fuzzy CG u representing the sentence "Very strong access to the car" (Eq. (4)), will have as equivalent the formula Φ (u), where LV and DM are respectively a linguistic value and a degree of membership (Eq. (5)):

$$u\text{: [value: [0,1]]} \leftarrow \text{(DM)} \leftarrow [\mathit{strong}\text{:\#1}] \leftarrow \text{(LV)} \leftarrow [\mathit{access}\text{: *}] \rightarrow \text{(LOC)} \rightarrow [\mathit{car}\text{: \#2}] \tag{4}$$

$$\Phi(u)\text{: } \exists x,y \in [0,1],\ \text{access}(x) \wedge \text{LV}(x,\#1) \wedge \text{strong}(\#1) \wedge \text{DM}(y,\#1) \wedge \text{value}(y) \wedge \text{LOC}(x,\#2) \wedge \text{car}(\#2) \tag{5}$$

The isomorphism between CG and predicate logic allows the representation of a design specification in a formal language of specification to have a better readability for designers. For example, in Fig. 4 we illustrate the translation of fuzzy GCs in the Z language (or Object-Z [79]) given the algorithm for this translation. The notation in Z language schemas has the advantage of making the representation in first order logic more explicit.

3.2.6. Chaining fuzzy specification language into fuzzy CAD models

In CAD modelling, a bill of materials (CAD-BOM) represents the files' structure. For an existing product, the language of specification outputs (including functions, formulas, variables, etc.) are sent to the files (part or assembly) in CAD-BOM via communicating agents. For a new product, the first order logic outputs (functions, formulas, and variables) are also used to create a CAD-BOM.

As a further aspect of the CAD modelling, formulas are identified as true or false, and are assigned a variable which is associated with a geometric or topological element of the CAD. For example, check the form's rules, IF condition THEN action, in Knowledgeware Advisor of Computer Aided Three-Dimensional Interactive Application (CATIA) is

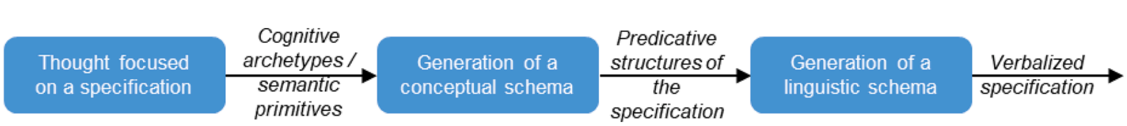

Fig. 2. Scheme describing the verbalization of specification.

Step 1:	*Search the graph u of the verb in the fuzzy GCs included in the canonical base (fuzzy engineering ontology)*
Step 2:	*Construct the graph* u_i *for the subject and try to join it with the relations AGNT, INST, OBJ according to u*
Step 3:	*For each verb argument, join the concept to the graph* u_i*, with the relation defined by the proposition that introduces the argument*
Step 4:	*Join the subgraphs corresponding to the modifiers (preposition, relative pronoun, etc.) to the graph* u_i
Step 5:	*Contextualize the complete graph* u_i *(e.g. negative, if the verb itself is negative)*
Step 6:	*Attach the graphs* $u_{j..k}$ *corresponding to the subordinate clauses*

Fig. 3. Fuzzy CG building from the semantic structure and the fuzzy engineering ontology.

Init:	*Create 1 fuzzy CG* u_i *per sentence analyzed*
Step 1:	*Join the graphs built for a section, Join(*u_1*, ..., *u_n*)* $\rightarrow$ *u*
Step 2:	*Identify external references to the section*
Step 3:	*Identify the parameters, variables and relations of the description, establishing:* *- the list of individual referents (object instance)* *- the list of concepts according to the type of referent* *- the list of relations according to the arity*
Step 4:	*Run the fuzzy CG* $\rightarrow$ *Z algorithm that provides the Scheme U*

Fig. 4. Building a formal description from the fuzzy CGs representing a section of specification.

programmed for evaluation. In this way, during the CAD modelling process, the CAD model is evaluated according to how well it satisfies the specifications.

4. Application and results

In this section we illustrate the method presented in the previous section with an example of a team of designers specifying a car door hinge [80]. This example proposes a cooperative semi-automatic process of diagrammatic representation (close to the UML language) of design description [66]. This process incorporates the six main phases presented in Fig. 1. The designers use the F-EGEON platform for this activity.

4.1. Explanation of the method's six phases

The components of the hinge are shown in Fig. 5. They are listed in the basic ontology, of which Fig. 6 presents a few examples, as well as the canonical definition of the product's basic functions. For example, the type "Door" is defined as a physical object, component of the type "Car", allowing a user to access the interior of a car. As for the "Access" type, it is defined as an action, a product function of a "Car", having a degree of membership to a linguistic value between 0 and 1. Realizing this ontology is a prerequisite for the method, as is fuzzy modelling (Fig. 7).

In the first phase of the method, the designers' specification statement is used to produce lists of components and functions, as illustrated in Fig. 8 (designers identify and verbalize the components of the product they specify).

Following the formulation of specifications, in phase two of the method, linguistic analysis is used to produce syntactic and semantic

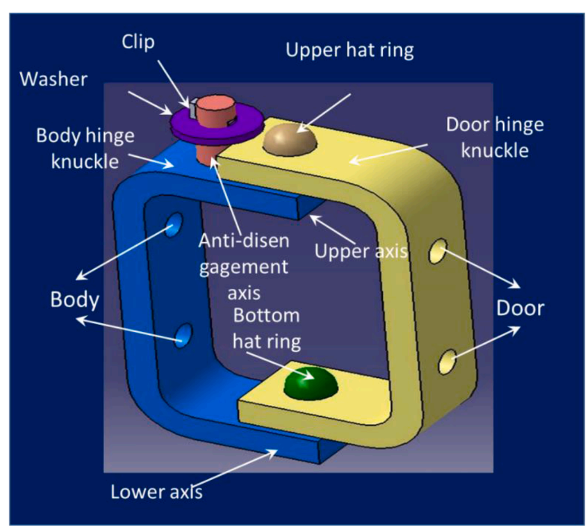

Fig. 5. A car door hinge and its components.

representations using a case grammar that we developed in a previous project [82]. The case grammar is presented in Fig. 9, where s is a sentence, vp is a verbal phrase, np is a noun phrase, v is a verb, n is a noun, p is a preposition, and d is a determiner, and AGNT, ACT, LOC, REL are casual relations. The result of phase two makes it possible to build the fuzzy CGs presented in Fig. 10 (phases three and four of the method). Then, in phase five, the specifications are transformed into formulas using the logic of first order predicates and can be translated if necessary into a specification language such as Z language, using the

TYPE Door IS
(TAX)→[Physical_Object],
(COMP)→[Car],
[User]←(AGNT)←[Access]→(LOC)→[Car].
...
TYPE Access IS
(TAX)→[Action],
(FUNC)→[Car],
(VL)→[FuzzySet, {Weak, Average, Strong}] →(DM)→[Value: [0,1]],
(AGNT)→[Entity, {Human, Animated_Object}],
(LOC)→[Entity].
TYPE Rotate IS
(TAX)→[Move],
(FUNC)→[Car],
(VL)→[FuzzySet, {Weak, Average, Strong}] →(DM)→[Value: [0,1]],
(OBJ)→[Entity, {Physical_Object}],
(REL)→[Entity].
TYPE Solidarize IS
(TAX)→[Action],
(FUNC)→[Car],
(VL)→[FuzzySet, {Weak, Average, Strong}] →(DM)→[Value: [0,1]],
(OBJ)→[Entity, {Physical_Object}],
(REL)→[Entity, {Physical_Object}].
TYPE Open IS
(TAX)→[Action],
(FUNC)→[Car],
(VL)→[FuzzySet, {Weak, Average, Strong}] →(DM)→[Value: [0,1]],
(AGNT)→[Entity, {Human, Animated_Object}],
(OBJ)→[Entity, {Physical_Object}],
(INST)→[Entity].

Fig. 6. Some examples of Fuzzy CGs defining the type of components and product functions.

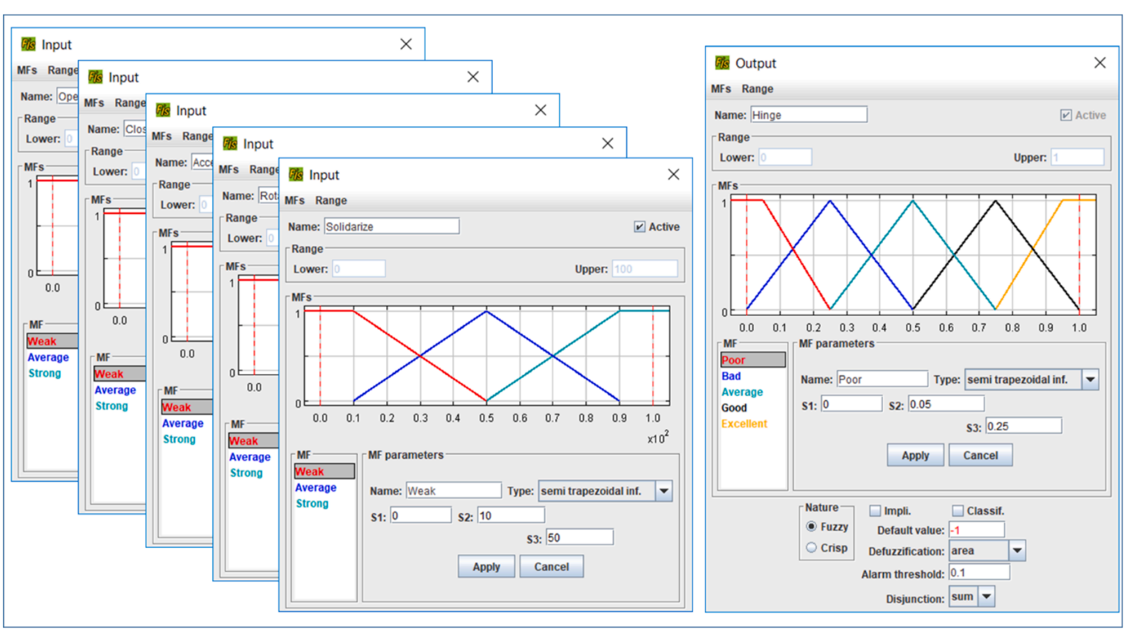

Fig. 7. Fuzzy modelling of the case study realized with FisPro software [81].

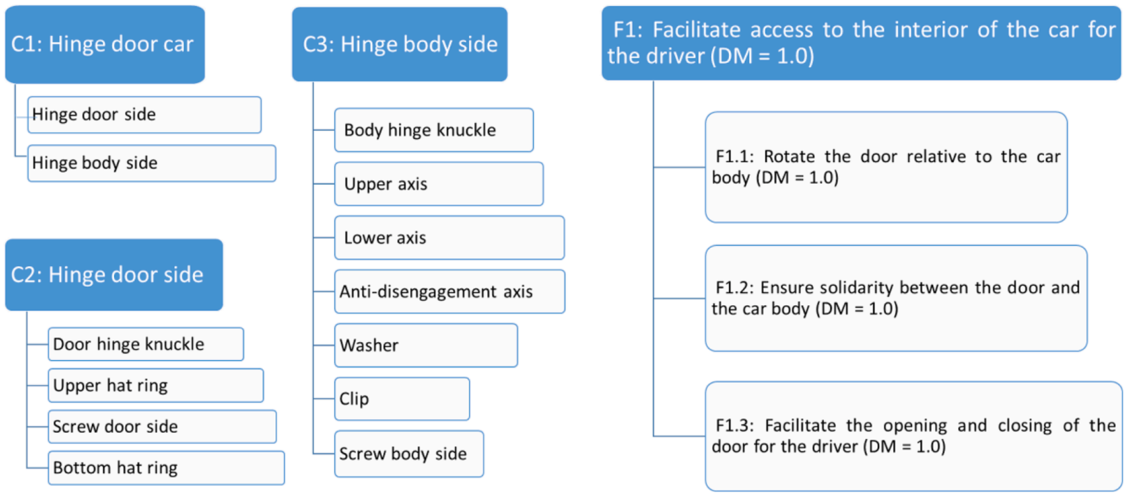

Fig. 8. Examples of lists of components and product functions built during phase 1.

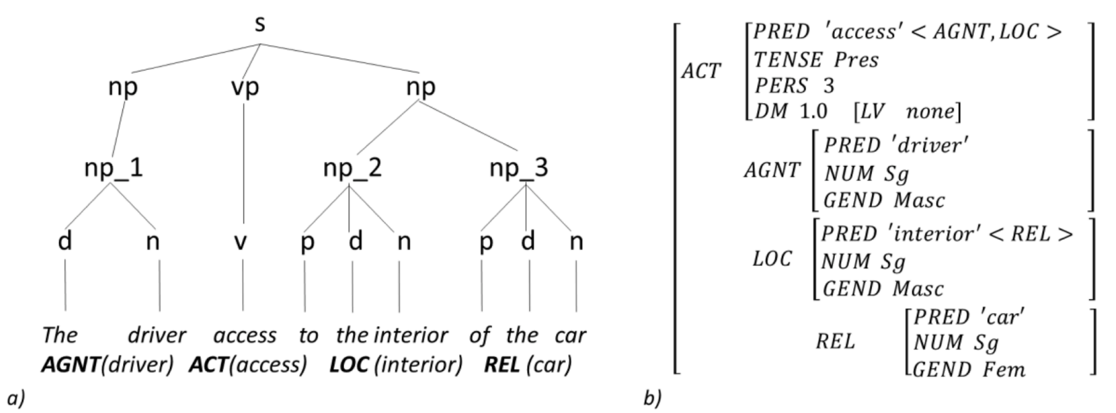

Fig. 9. Simple example of (a) syntactic tree and (b) fuzzy casual relations built, during the phase 2.

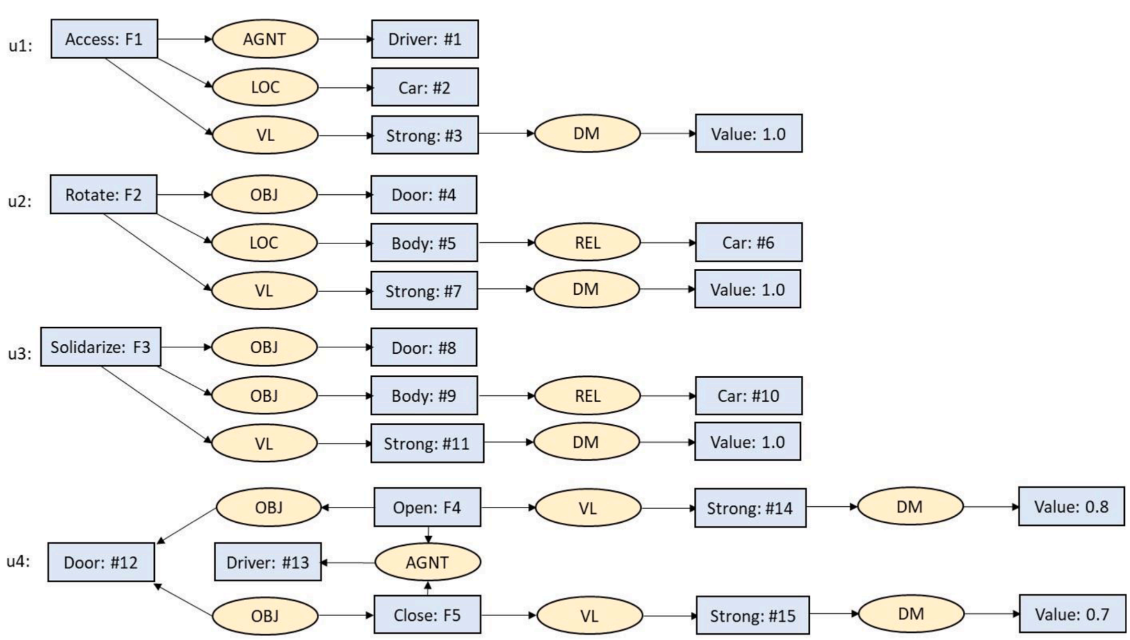

Fig. 10. Four examples of fuzzy CGs built during phases 3 and 4.

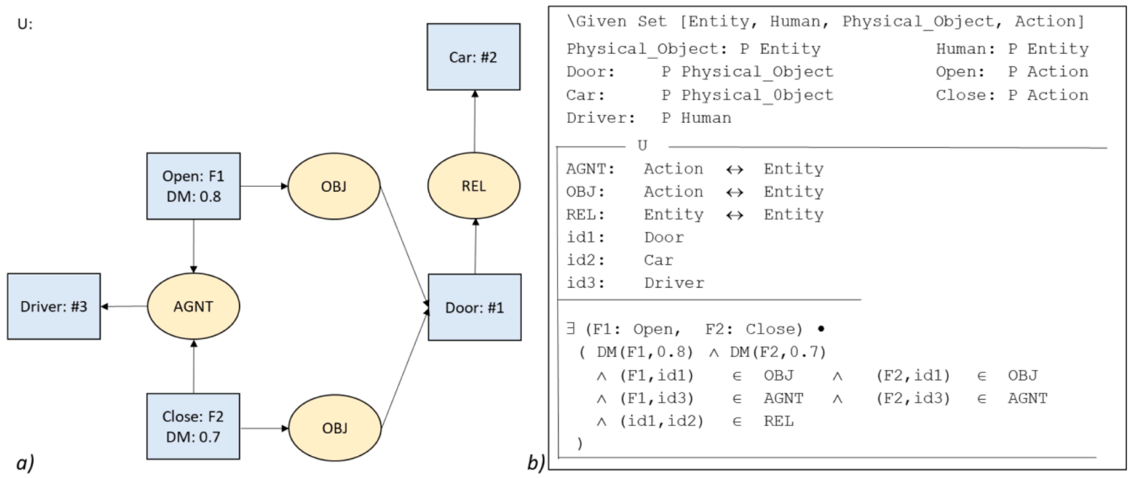

Fig. 11. Given the sentence "The driver must be able to open (MD = 0.8) and close (MD = 0.7) the door of the car", (a) the fuzzy CG U built during phase 4, and (b) the schema Z built during phase 5.

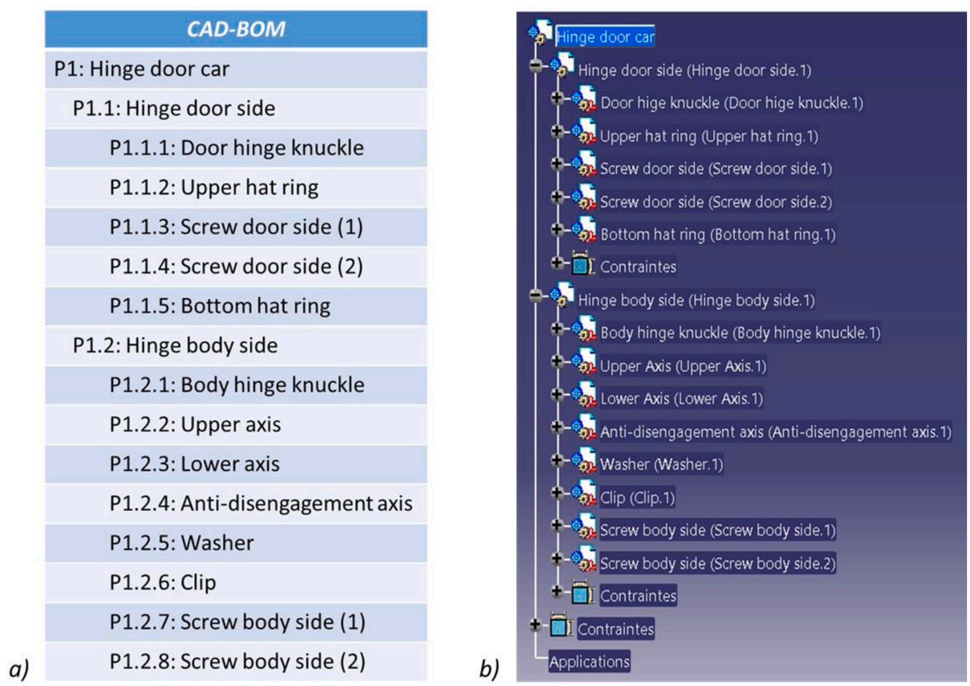

Fig. 12. (a) Example of CAD-BOM, and (b) product structure obtained in Computer Aided Three-Dimensional Interactive Application (CATIA V5 R21), illustrating phase 6.

algorithms we present in [29] (Fig. 11). In the last phase of the method, these representations are chained into CAD models, as we illustrate in Fig. 12.

4.2. F-EGEON platform for assisting requirements engineering

In order to put our intelligent requirements engineering approach into practice, we have developed a fuzzy agent-based tool called F-EGEON (Fuzzy Engineering desiGn sEmantics elabOration and applicatioN). This tool operates according to the first 5 phases of the approach presented in Fig. 1. F-EGEON makes it possible to represent the information transmitted by designers during their engineering design activities on two levels: the first level corresponds to a designer's individual activity and to the visualization of the requirements independently of each other; the second level, where the requirements graphs are merged, corresponds to a collaborative activity between designers and a global visualization of the requirements. These two levels of information and visualization of information can be seen as two modes of use of F-EGEON, which we illustrate below.

A designer can state a set of requirements (verbally or textually) as shown in Fig. 13. The current version of the tool operates at the sentence level. In other words, a requirement corresponds to a sentence formulated by a designer (frame at the bottom of the tool, where the text is entered on the keyboard or from a voice interface, Fig. 13). Then, each of the requirements is represented by a graph (drawing area of the tool, Fig. 13) and the design tree is completed (left frame of the tool, Fig. 13). In this level of representation, the functions are therefore independent. This is the first phase of understanding the function, but also the transformation of the function into a solution.

In a cooperative design activity, several designers can jointly state the requirements of a product (always verbally or textually). In this case, designers can ask the tool to merge the different graphs generated for

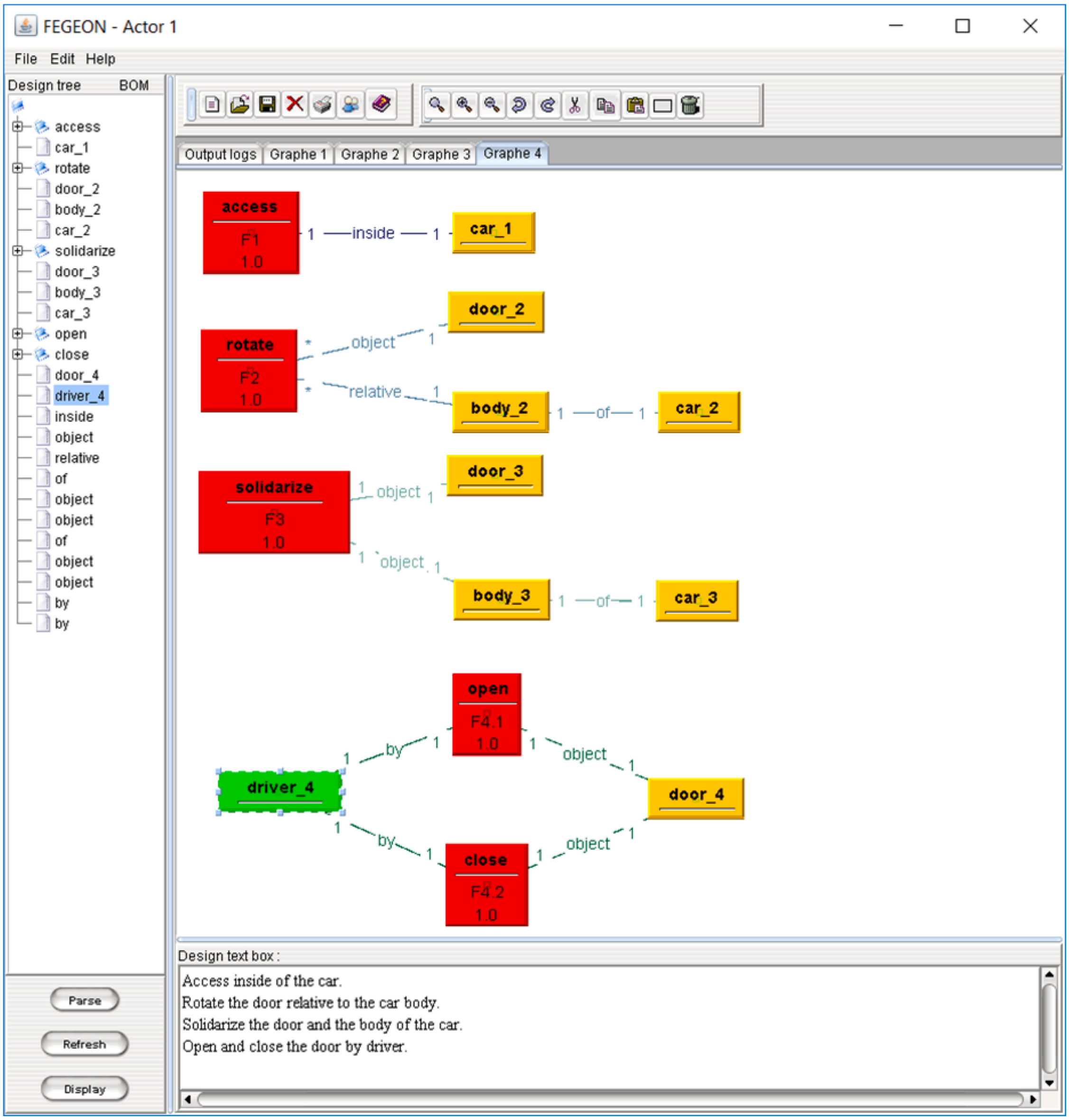

Fig. 13. Semantic representations during an individual use of F-EGEON.

each requirement, as illustrated in Fig. 14 (screen of a designer identified as Actor 3). It should be noted that this graph fusion function can also be activated in the individual working mode described above.

The knowledge network generated in the second level of representation shows the interactions between the functions. This representation is essential so that all the designers can visualize the result of the cooperative work, but also allows each designer to be informed individually of the state of construction of the common graph which can therefore detect conflicts between functions and their parameters. For instance, the parameters of the actions “rotate” and “solidarize” interacting with “body” and “door” can be conflictual. Similarly, the actions “open” and “close” related to “door” and “driver” are semantically conflictual. The network can also assist the designer in explaining the causality and the deductions, as well as showing the interaction between an established function and a newly defined one.

5. Results and discussion

The developed fuzzy agent-based tool F-EGEON tool encapsulates the engineering requirements’ fuzzy semantic network. It operates according to the proposed formal approach.

F-EGEON makes it possible to represent the information transmitted by designers during their engineering design activities on two levels: the first level corresponds to a designer’s individual activity and to the visualization of the requirements independently of each other; the second level, where the requirements graphs are merged, corresponds to a collaborative activity between designers and a global visualization of the requirements. It also assists the designer in explaining causality and deductions.

The time and quality of requirements is improved by intelligent requirements engineering from natural language and through the process of chaining fuzzy requirements engineering from natural language into CAD models. Design proposals are also improved by specialist team members who collaborate to develop requirements.

The method proposed here can be employed as a foundation for design through social processes such as discussions. The method’s central focus is on group discussions in which fuzzy design requirements are articulated. From this perspective, team building and collaborative working are assisted by intelligent requirements engineering derived from natural language, and its chaining into CAD models. The process of using intelligent models in open, exploratory, and dialectical ways by the design team is therefore made possible.

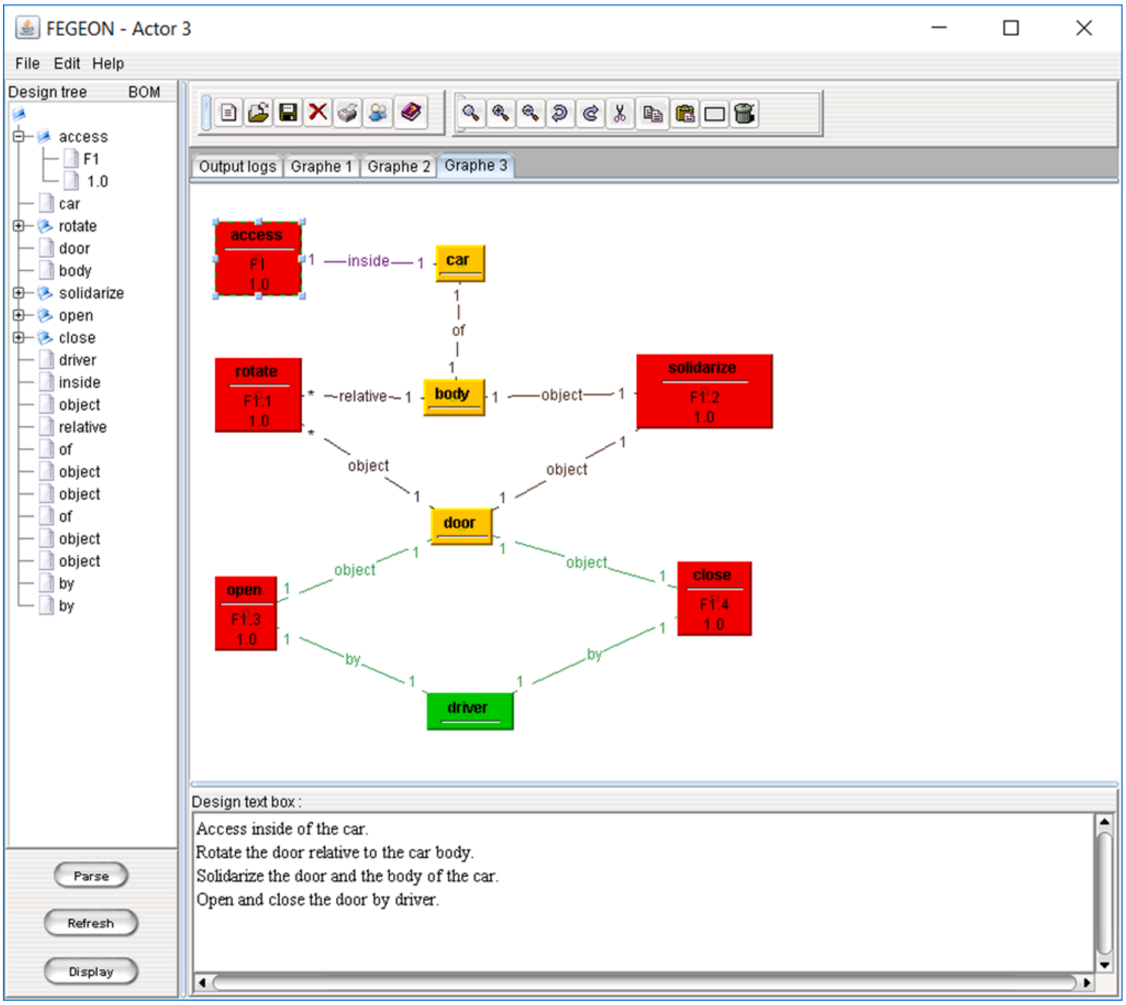

Fig. 14. Semantic representations during a cooperative use of F-EGEON.

6. Conclusion

This research aimed to develop a method for fuzzy requirements engineering by chaining natural language into CAD models. The proposed approach shows that moving from linguistic analysis to representing engineering requirements involves translating the fuzzy syntactic structure of language into a fuzzy semantic formation and representing this in fuzzy conceptual graphs. The isomorphism between fuzzy conceptual graphs, and fuzzy predicate logic enables a fuzzy formal language of requirement to be produced. This is translated into Z language and is subsequently chained into Computer Aided Three-Dimensional Interactive Application (CATIA) models. The approach shows that processing natural language and building engineering languages are core components of requirements engineering.

It can be concluded that the formalisation presented here increases requirements' reliability and relevance in both CAD models and PLM systems more generally. The results indicate that the time, the quality of requirements and design proposals are improved through the process of chaining fuzzy requirements engineering from natural language into CAD models.

The proposed intelligent requirements engineering model indicates open-ended scenarios and invites further contributions. It does not indicate a completed system, or offer a closed AI application. Rather, the systems proposed increase the breadth of scenarios in which AI is applicable for engineering requirements.

Nonetheless, the intelligent requirement engineering by the proposed systems are not intended only to represent the proposals, but also to enable positive actions and behaviour within the design environment. An intelligent requirement engineering system should also facilitate analysis of biased representation and misrepresentation. These limitations and shortcomings can produce a negative effect on design development, as well as negatively impacting the social process of design. Consequently, future research for developing fuzzy engineering design semantics elaboration method and tool should highlight the assertion of requirements and making intelligent recommendations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets Syst. 1 (1) (1978) 3–28.
[2] L. Wittgenstein, The Blue and Brown books, Blackwell, Oxford, 1958.
[3] R. Thom, R. Narasimhan, A. Haefliger, Topologie et linguistique, Eds.. Essays On Topology and Related Topics, Springer, Berlin, 1970, pp. 226–247.
[4] R. Thom, Structural Stability and Morphogenesis, Westview Press, Oxford, 1988.
[5] W. Wildgen, Archetypal dynamics in word semantics: an application of catastrophe theory. Words, Worlds, and Contexts, De Gruyter, Berlin, Boston, 1981, pp. 234–296.
[6] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[7] K. Ryan, The role of natural language in requirements engineering, in: Proceedings of the IEEE International Symposium on Requirements Engineering, San Diego, CA, USA, 1993, pp. 240–242.

[8] T. Loomis, Software design issues for natural language processing, Mach. Transl. 2 (4) (1987) 219–230.
[9] S. Vadera, F. Meziane, From english to formal specifications, Comput. J. 37 (9) (1994) 753–763.
[10] S. Du, On the Use of Natural Language Processing For Automated Conceptual Data Modelling, University of Pittsburgh, 2008. PhD ThesisJune 25th.
[11] A. Arellano, E. Carney, M.A. Austin, Natural language processing of textual requirements, in: Proceedings of the Tenth International Conference on Systems (ICONS 2015), Barcelona, Spain 9, 2015, pp. 93–97.
[12] S. Gulia, T. Choudhury, An efficient automated design to generate UML diagram from natural language specifications. 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), IEEE, 2016, pp. 641–648.
[13] S. Ghosh, D. Elenius, W. Li, P. Lincoln, N. Shankar, W. Steiner, ARSENAL: automatic requirements specification extraction from natural language. NASA Formal Methods Symposium, Springer, 2016, pp. 41–46.
[14] S. Čandrlić, M.A. Katić, M. Pavlić, A system for transformation of sentences from the enriched formalized node of knowledge record into relational database, Expert Syst. Appl. 115 (2019) 442–464.
[15] I.G. Harris, Extracting design information from natural language specifications, in: Proceedings of the 49th Annual Design Automation Conference, 2012, pp. 1256–1257.
[16] F. Nazir, W.H. Butt, M.W. Anwar, M.A.K. Khattak, The applications of natural language processing (NLP) for software requirement engineering - a systematic literature review. International Conference on Information Science and Applications, Springer, 2017, pp. 485–493.
[17] F. Dalpiaz, A. Ferrari, X. Franch, C. Palomares, Natural language processing for requirements engineering: the best is yet to come, IEEE Softw. 35 (5) (2018) 115–119.
[18] F.S. Bäumer, M. Geierhos, NLP in OTF Computing: current approaches and open challenges, in: Proceedings of the International Conference on Information and Software Technologies, Cham, 2018, pp. 559–570. Springer.
[19] D. Falessi, G. Cantone, G. Canfora, Empirical principles and an industrial case study in retrieving equivalent requirements via natural language processing techniques, IEEE Trans. Softw. Eng. 39 (1) (2011) 18–44.
[20] X.Y. Kou, S.K. Xue, S.T. Tan, Knowledge-guided inference for voice-enabled CAD, Comput. Aided Des. 42 (6) (2010) 545–557.
[21] R. Pinquié, P. Véron, F. Segonds, N. Croué, Natural language processing of requirements for model-based product design with ENOVIA/CATIA V6, in: Proceedings of the IFIP International Conference on Product Lifecycle Management, Cham, 2015, pp. 205–215. Springer.
[22] A. Ferrari, A. Esuli, An NLP approach for cross-domain ambiguity detection in requirements engineering, Autom. Softw. Eng. 26 (3) (2019) 559–598.
[23] H. Dawood, J. Siddle, N. Dawood, Integrating IFC and NLP for automating change request validations, J. Inf. Technol. Constr. 24 (2019) 540–552.
[24] Y. Zhang, B. Wallace, A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification, 2015 arXiv: 1510.03820v4.
[25] L. Deng, Y. Liu, Deep Learning in Natural Language Processing, Eds., Springer, 2018.
[26] T. Young, D. Hazarika, S. Poeria, E. Cambria, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75.
[27] A. Torfi, R.A. Shirvani, Y. Keneshloo, N. Tavaf, E.A. Fox, Natural Language Processing Advancements By Deep learning: A survey, 2020 arXiv:1510.03820v4.
[28] D. Sperber, D. Wilson, Relevance Theory. Handbook of Pragmatics, Blackwell, 2002.
[29] A.J. Fougères, Formal specifications building from specifications written in natural language, in: Proceedings of the European Conference Human Centered Processes, (HCP'99), Brest, France, 1999, pp. 225–232.
[30] N. Indurkhya, F.J. Damerau, Handbook of Natural Language Processing, Eds., 2, Chapman and Hall/CRC, 2010.
[31] L.S. Vygotski, Mind in Society: The development of Higher Psychological Processes, Harvard University Press, Cambridge MA, 1980.
[32] M.J. French, Engineering Design: the Conceptual Stage, Heinemann Educational Books Ltd, London, 1971.
[33] G. Pahl, W. Beitz, Engineering Design: A Systematic Approach, The Design Council, London, 1984.
[34] V. Hubka, W.E. Eder, Theory of Technical Systems, Springer Verlag, New York, 1988.
[35] N. Suh, Principles of Design, Oxford University Press, New York, 1990.
[36] D.G. Ullman, The Mechanical Design Process, McGraw-Hill, New York, 1992.
[37] F.J. Erens, K. Verhulst, Architectures of product families, Comput. Ind. 33 (2–3) (1997) 165–178.
[38] K. Ulrich, S.D. Eppinger, Product Design Development, McGraw-Hill, New York, 1995.
[39] J. Jiao, M. Tseng, Fundamentals of product family architecture, Integr. Manuf. Syst. 11 (7) (2000) 469–483.
[40] K.N. Otto, K.L. Wood, Product design techniques. Reverse Engineering and New Product Development, Prentice Hall, Upper Saddle River, NJ, 2001.
[41] R.V. Welch, J.R. Dixon, (1992). Representing function, behavior and structure during conceptual design. In: Taylor D.L. and Stauffer L.A. (eds.) Design Theory and Methodology, American Society of Mechanical Engineers, Vol. 42, pp. 11–18.
[42] A. Chakrabarti, L. Blessing, Representing functionality in design, Artif. Intell. Eng. Des. Anal. Manuf. 10 (1996) 251–253.
[43] M.S. Erden, H. Komoto, T.J. VanBeek, V. D'Amelio, E. Echavarria, T. Tomiyama, A review of function modeling: approaches and applications, Artif. Intell. Eng. Des. Anal. Manuf. 22 (2008) 147–169.
[44] B. Chandrasekaran, J.R. Josephson, Function in device representation, Eng. Comput. 16 (3–4) (2000) 162–177.
[45] B. Chandrasekaran, Representing function: relating functional representation and functional modeling research streams, Artif. Intell. Eng. Des. Anal. Manuf. 19 (2005) 65–74.
[46] J. Hirtz, R.B. Stone, D.A. McAdams, S. Szykman, K.L. Wood, A functional basis for engineering design: reconciling and evolving previous efforts, in: Res. Eng. Des., 13, 2002, pp. 65–82.
[47] E.R. Deciu, E. Ostrosi, M. Ferney, M. Gheorghe, Configurable product design using multiple fuzzy models, J. Eng. Des. 16 (2) (2005) 209–233.
[48] T. Chen, M.J.J. Wang, Forecasting methods using fuzzy concepts, Fuzzy Sets Syst. 105 (1999) 339–352.
[49] T.C.T. Chen, Y.C. Wang, C.W. Lin, A fuzzy collaborative forecasting approach considering experts' unequal levels of authority, Appl. Soft Comput. 94 (2020), 106455, https://doi.org/10.1016/j.asoc.2020.106455.
[50] D. Huang, T. Chen, M.J.J. Wang, A fuzzy set approach for event tree analysis, Fuzzy Sets Syst. 118 (1) (2001) 153–165.
[51] H. Issa, E. Ostrosi, M. Lenczner, R. Habib, Fuzzy holons for intelligent multi-scale design in cloud-based design for configurations, J. Intell. Manuf. 28 (2017) 1219–1247.
[52] E. Ostrosi, A.J. Fougères, Z.F. Zhang, J. Stjepandić, Intelligent modular design with holonic fuzzy agents, Adv. Manuf. 9 (1) (2021) 81–103.
[53] E. Ostrosi, A.J. Fougères, Optimization of product configuration assisted by fuzzy agents, Int. J. Interact. Des. Manuf. 5 (1) (2011) 29–44 (IJIDeM).
[54] A.J. Fougères, E. Ostrosi, Fuzzy agent-based approach for consensual design synthesis in product configuration, Integr. Comput. Aided Eng. 20 (3) (2013) 259–274.
[55] E. Ostrosi, A.J. Fougères, M. Ferney, Fuzzy agents for product configuration in collaborative and distributed design process, Appl. Soft Comput. 12 (8) (2012) 2091–2105.
[56] E. Ostrosi, A.J. Fougères, M. Ferney, D. Klein, A fuzzy configuration multi-agent approach for product family modelling in conceptual design, J. Intell. Manuf. 23 (6) (2012) 2565–2586.
[57] A.J. Fougères, E. Ostrosi, Intelligent requirements engineering from natural language and their chaining toward CAD models, in: Digital Proceedings of TMCE2020, TMCE 2020 Repository, 2020, pp. 403–416, at tmce.io.tudelft.nl/proceedings.
[58] D. Choulier, A.J. Fougères, E. Ostrosi, Developing multiagent systems for design activity analysis, Comput. Aided Des. 59 (2015) 201–213.
[59] J.F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley Publishing Company, Reading, MA, 1984.
[60] V. Wuwongse, M. Manzano, Fuzzy conceptual graphs. International Conference on Conceptual Structures, Springer, 1993, pp. 430–449.
[61] P. Buche, J. Dibie-Barthélemy, L. Ibanescu, Ontology mapping using fuzzy conceptual graphs and rules, in: Proceedings of the ICCS Supplement 1724, 2008.
[62] M. Amrani, D. Blouin, R. Heinrich, A. Rensink, H. Vangheluwe, A. Wortmann, Towards a formal specification of multi-paradigm modelling. ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, 2019, pp. 419–424.
[63] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning —I, Inf. Sci. 8 (3) (1975) 199–249 (Ny).
[64] B. Bouchon-Meunier, D. Dubois, L. Godo, H. Prade, Fuzzy sets and possibility theory in approximate and plausible reasoning. Fuzzy Sets in Approximate Reasoning and Information Systems, Springer, Boston, MA, 1999, pp. 15–190.
[65] D. Fensel, Ontologies. Ontologies, Springer, Berlin, Heidelberg, 2001, pp. 11–18.
[66] C.C. Aggarwal, C. Zhai, Mining Text Data, Eds., Springer Science & business Media, 2012.
[67] A.J. Fougères, Faciliter Les Interactions Coopératives Entre Utilisateurs De Systèmes coopératifs. Vers un Système De Médiation Coopérative, Éditions Universitaires Européennes, 2011, pp. 160–168.
[68] A. Culioli, Cognition and Representation in Linguistic Theory, 112, John Benjamins Publishing, 1995.
[69] N. Chomsky, Language and Mind, Cambridge University Press, 2006.
[70] C.J. Fillmore, E. Bach, R. Harms, The Case For Case, Ed., Universals in Linguistic Theory, New York, 1967.
[71] R. Van Trijp, The Evolution of Case Grammar, Language Science Press, 2016.
[72] M. Butt, Theories of Case, Cambridge University Press, 2006.
[73] R.C. Schank, Conceptual dependency: a theory of natural language understanding, Cogn. Psychol. 3 (4) (1972) 552–631.
[74] B. Pottier, Sémantique Générale, Presses Universitaires de France, Paris, 1992, p. 240p.
[75] M.C. De Marneffe, B.B. MacCartney, C.D. Manning, Generating typed dependency parses from phrase structure parses, in: Proceedings of Int. Conf. on Language Ressources and Evaluation (LREC), 6, 2006, pp. 449–454.
[76] H. Kamp, A theory of truth and semantic representation of language. Meaning and the Dynamics of Interpretation, Brill, 1981, pp. 329–369.
[77] G. Sabah, Natural language understanding, where are we going? Where could we go? Comput. J. 54 (9) (2011) 1505–1513.
[78] R. Thomopoulos, P. Bosc, P. Buche, O. Haemmerle, Logical interpretations of fuzzy conceptual graphs, in: Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS, 2003, pp. 173–178. IEEE.
[79] G. Smith, The Object-Z Specification Language, 1, Springer Science & Business Media, 2012.
[80] E. Frey, E. Ostrosi, S. Gomes, L. Roucoules, Tool based supporting collaborative design from specification development to CAD modelling, in: Proceedings of the

TMCE 2010 symposium (Tools and Methods of Competitive Engineering, Italy, 2010, pp. 941–950.
[81] S. Guillaume, B. Charnomordic, Learning interpretable fuzzy inference systems with FisPro, Inf. Sci. 181 (20) (2011) 4409–4427 (Ny).
[82] A.J. Fougères, Assistance to agent-based μ-tools development for a co-operative design platform, in: Proceedings of Virtual Concept 2006, Springer, 2006, pp. 1–4, hal-00572837.