Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Advanced Engineering Informatics 21 (2007) 153–168
ADVANCED ENGINEERING INFORMATICS
www.elsevier.com/locate/aei

Towards the design of intelligent CAD systems: An ontological approach

Gianluca Colombo, Alessandro Mosca, Fabio Sartori *

University of Milano – Bicocca, Department of Computer Science, Systems and Communication, via Bicocca degli Arcimboldi 8, 20126 Milan, Italy

Received 18 October 2006; accepted 12 November 2006

Abstract

Although many sophisticated CAD integrated environments have been developed and are currently adopted by enterprises in the design of mechanical parts and components, such kinds of tools should be extended in order to reach higher levels of performance. To this aim, Artificial Intelligence techniques are particularly suitable to provide CAD tools with a sort of "intelligence" typical of human experts. In particular a complex mechanical object to be designed exploiting CAD systems can be considered as an aggregation of simpler components that have to be put together in order to satisfy precise design rules owned by expert designers core knowledge. This paper presents an ontological approach to the problem of representing relationships among step-by-step more complex parts, in order to obtain a final product that fully meets initial requirements.

1. Introduction

The development of effective software systems to support people involved in the design and manufacturing of complex products has become a very important research field. This fact is demonstrated by the growing number of Conferences and events dedicated to this topic, which presents very complex research issues. This is particularly true in the domain of mechanical industry, where people is generally involved in difficult configuration problems [1] aimed at obtaining a final product meeting marketing requirements that is also more appealing than competitors' ones in terms of price, quality and so on.

A complex mechanical product is made of hundreds and, sometimes, thousands of components, from the simplest ones (e.g. screws, screw bolts, nails and so on) to most composite one. Typically, people involved in the design of such objects exploit sophisticated computer aided design (CAD) tools, like CATIA. Unfortunately, although CAD tools are particularly suitable to check how the single components are put together from the geometrical point of view (e.g. is the screw large enough for that hole? Is that screw bolt right for the chosen screw?) they do not support experts in monitoring the project from the functional perspective (e.g. is that part correctly designed in order to properly accomplish this function?). While the former point is relatively simple to be taken into account, the latter is much more complicated to consider, since it depends on the designers' experience and knowledge.

For this reason, the configuration of complex products is very difficult from the knowledge representation point of view, and building decision support systems for designers in the mechanical industry field is not simple. In this field, three main types of information and knowledge can be recognized: geometric information (geometric representation of the model of the product in most cases the CAD model), information/knowledge about the documents used during the design process (standards, manuals, and recommendations) and information/knowledge about inference rules and external program applications (calculations, simulations and so on) [2].

* Corresponding author. Tel.: +39 02 64487857; fax: +39 02 64487839.
E-mail addresses: gianluca.colombo@disco.unimib.it (G. Colombo), ale.m@disco.unimib.it (A. Mosca), sartori@disco.unimib.it (F. Sartori).

1474-0346/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.aei.2006.11.003

Starting from the end of the 1980s many tools have been marketed as *knowledge-Based Engineering* (KBE)-Tools, for example ICAD (Knowledge Technologies International) or PACElab (PACE). These tools provide a software environment for automating repetitive engineering tasks [3]. Knowledge-Based System applications have in fact a big potential to reduce cost and time for repetitive engineering tasks but require a relevant effort to collect and formalize the required knowledge in a knowledge representation scheme. In this field, called *Engineering Design* in the rest of the paper, one of the most known examples of application to the industrial planning of complex objects has been proposed by Gero and Maher [4]. They defined a conceptual and computational approach starting from the definition of design as "a *goal-oriented, constrained, decision-making, exploration and learning activity which operates within a context which depends on the designer's perception of the context*" [5]. Their approach defines specific knowledge representation schemes (the so-called *prototypes*) for the definition of the conceptualization and ideation process generally followed by a draftsman and proposes the Case-based design paradigm to reuse previous design solutions to solve similar design problems [6].

In this paper, we want to show how the adoption of a functional approach to the representation of design objects which fits with a mereological theory based on Husserl's philosophy [7] can be profitably exploited to create a framework able to capture all the aspects involved in designing a complex mechanical object from the functional point of view. A brief formalization of Husserl's mereological position is proposed and discussed from the engineering design standpoint in Section 4.3. In particular, the adoption of a mereological theory allows to express in a complete way the relationships among the different parts a complex object is made of, according to the tacit knowledge of an expert designer. The following section provides a conceptual description of the main aspects of engineering design, as a composite process that ends with the production of a mechanical object. The conceptual model of our framework will be then described in Section 4, through the use of a simple example (i.e. the configuration of a bicycle). Conclusions and future works will end the paper.

2. Engineering design aspects: a conceptual view

Engineering design can be viewed as *an articulate process composed of phases, where each phase represents a combinatorial action on the parts the composite object is constituted of*.

To realize an object meeting the desired market requirements, engineering designers have to deal at the same time with different kinds of knowledge coming from different epistemological sources: "static" knowledge about objects or, Ontological knowledge [8] (which is often represented in a declarative form), and "dynamic" knowledge about processes (which is often expressed in "procedural terms").

In this paper, we present an ontological approach to the development of engineering design support systems. More precisely, the goal of this work is to derive useful indications on how to exploit specific domain ontologies in order to support the acquisition and representation of dynamic knowledge involved in design activities.

A number of references in literature [9–11,4,12,13] indicates that the competence of engineering designers is related to their ability in considering functional constraints over the parts of the objects they are designing. According to our viewpoint, this expert designers' competence gives the ability to navigate ontological and procedural knowledge, always considering different kinds of relationships among each part of the desired composite object.

The central role of heuristics in performing design tasks mainly resides in this capability to shift through different epistemological dimensions, represented by the static and the procedural sides of knowledge. We look at design heuristics as a set of competencies, growing from experience, which bridge the gap between static and dynamic knowledge and makes designers able to articulate the design process referring to functional constraints.

Therefore, the development of a Knowledge-Based System supporting engineering design activities must take into account the formal representation of both these knowledge sides [12,13]. The conceptual framework we are proposing is aimed at offering to knowledge engineering discipline a theoretical structure for acquiring and representing such a knowledge.

In the following section, we will take into account the role performed by mereology in filling the gap between static and dynamic knowledge representation.

3. The role of function and mereology in design

As observed in [4], a product design process begins with *functional* or *conceptual design*, followed by *basic design* and *detailed design*. Among these steps, functional design plays the central role in ensuring design quality and product innovativeness.

While traditional mechanical computer-aided design, based on geometric modelling, targets detailed design support (e.g. low level technical activities on the model of the designed object), future CAD should support the entire design process, including functional design. Future CAD technology should thus represent and reason about functional requirements of design objects, a facility that traditional CAD systems do not provide.

The traditional engineering design research community has widely accepted this conceptual design methodology [4]: first, a designer determines the entire function of a design object by analyzing the specifications of the product to be built. He/she then divides the function recursively into sub-functions, a process that produces a functional organization. In correspondence to each sub-function, the designer uses a catalogue to look up the most appropriate elements (a component or a set of components) that are able to perform the functional requirement. Finally, the designer composes a design solution from those selected elements.

Here, function plays a crucial role, because the results of the design depend entirely on the decomposition of the function and on the designer's capability to build the appropriate object realizing that function [14,15,12]. As a result, a designer obtains a micro–macro hierarchy of functions that are projected on the aggregate of parts the composite objects is constituted of.

Thus, when designers speak about the "function" held by an object or by one of its components, they can speak about it because they have sufficient knowledge for associating functions to a suitable object structure: this kind of knowledge is called *ontological*.

Functions are knowledge abstractions by which engineering designers conceptualize the object with specific reference to the goals for which it is designed. On the basis of what we have discussed in the previous section, ontological knowledge is put in action by designers for describing design entities in terms of *Part–Whole* relations induced by function decomposition [4].

Therefore, capturing mereological relations may on one hand enhance our representation of the engineers' cognitive structures activated during the problem solving activity, and on the other it can facilitate the development of more effective Knowledge-Based Systems supporting it.

The nature of the compositional relations, however, can widely vary. Understanding these relations allows engineers to reason about the artifacts and to make clearer the sets of ontological and procedural constraints necessary to obtain a final object that meets the market requirements. Not being able to reason about the relationships that hold between different parts and the wholes they belong to can be detrimental to effective product models and design processes.

4. A conceptual framework for capturing engineering knowledge

In this section, a conceptual framework for the acquisition and representation of engineering design knowledge will be presented. A Functional approach will be followed through a mereological description of design entities with the aim of integrating static and dynamic knowledge treatment.

In Section 4.1, the epistemological levels involved in design activity will be taken into account with reference to well known studies and methodological approaches. The Function Behavior–Structure (FBS) model will be discussed in order to present the knowledge levels involved in design activities. Under this perspective our position about functions definition and role in engineering design contexts will be clarified.

In the two following sections, a philosophical inquiry on mereology and a conceptual framework for the representation of engineering design static knowledge will be informally presented. Section 5.2 introduces a set of conceptual relations that are useful in defining engineering design ontological knowledge. In the last section a procedural knowledge model represented by SA-Nets will be introduced.

4.1. Epistemological levels in engineering design

As well stressed in literature, in the context of engineering design an artefact can be considered with respect to the different knowledge sources involved in the design activity. These sources have been clustered in four types as follows [16]:

- *Structural knowledge:* knowledge about the components which comprise the design object and their relations.
- *Behavioral knowledge:* knowledge about the behavior of the design object, i.e. about ways the device responds to changes in its environment and/or in its own state; this type of knowledge describes components in terms of the physical quantities that characterize their state and the physical laws that rule their operation.
- *Teleological knowledge:* knowledge about the purpose and the way the design object is intended to be used; this type of knowledge describes the goals assigned to the artefact and enables designers to translate market requests expressed in some sets of performances associated to the intended use of the artefact also into specific expected behavior of the artefact.
- *Functional knowledge:* as indicated in Gero [5], it is unthinkable that any designer would work directly on the structure of an artefact to be designed without first considering behaviors and goals of the artefact itself. Therefore, structural knowledge is always guided by heuristics regarding:
  (1) The behavior expected from the artefact to decide on its structure.
  (2) The goal given to such behavior.

Functional knowledge is exactly defined as knowledge about how the behavior of the design object is used to accomplish its intended use.

As noted in [17], these knowledge levels can be organized in two broad epistemological categories, depending on their "objective" or "subjective" nature. Design objects considered under the perspective of their structures and behaviors are objective entities in the sense that the structure is given by the physical existence of the object, while the behavior can be (objectively) determined based on physical principles [17].

On the other side design objects considered under the perspective of Teleology and Functions are subjective entities, in the sense that the first one (the teleological knowledge) reflects the intention of a human (the designer or the user) in using the object, while the second one (the functional knowledge) is an abstraction of the object behavior [13].

This epistemological distinction enables to understand why the acquisition and representation of functional knowledge is recognized as a general task in the development of a KBS to support engineering design activity [15]. Without this analysis, it would be impossible to define the heuristics which guide the design activity of an expert designer.

From the knowledge engineering point of view, this topic becomes more and more relevant because design activity is essentially a non-bookish competence which derives from subjective experience. The problem in grasping these competencies lies in their own volatility and informality, what Nonaka [18] has called "tacit" and "implicit" knowledge.

In the perspective of engineering design, focusing on the acquisition and representation of functional knowledge enables knowledge engineering to enter the subjective side of knowledge. This side is the specific domain knowledge adopted by experts to produce structural knowledge about the design object that is sufficient to allow the manufacturing or construction of the object [17].

As mentioned above, during design activity for solving planning and configuration problems [19] functional knowledge is understood as a necessary bridge between behavioral and teleological knowledge [16]. Two distinct but correlated interpretations of functional knowledge can be underlined in this definition.

The first one considers functions as abstractions of behaviors that are adopted by designers in the conceptual phases of design activity. In this framework, the development of functional ontologies in connection with engineering design KB-support systems is aimed at realizing conceptual and computational models for the integrated treatment of behavioral and structural knowledge [20]. Here functional knowledge is helpful in retrieving the behavior of some existing design which can provide a required function [4], or in developing frameworks for cooperative engineering [20].

The second interpretation considers functions as explanations of the intended use of the artefact instead of abstractions of behaviors: this interpretation is followed when design activity is considered under the viewpoint of creative design, where the main goal of a KBS is to induce new structural knowledge in response to functional requirements [21].

Following the first interpretation behavioral knowledge on design object is required to realize the mapping between functions and structures, and thus to develop KBSs that are able to suggest the suitable design object structure in response to specific functional request. In the second case teleological knowledge is required for supporting the decisions on the object structure when a structure is not known but it must rather be defined for the first time on the basis of the expected behaviors derived from the intended use of the object to be designed (see Gero's proposal of *Situated Function Behavior–Structure Modelling* [21]). However, in both cases the assumption of the framework regards the fact that without knowledge of the real behavior of the design object, a function is deprived of the aspects concerning reasoning and operativity [16].

In this paper, the objective of a domain functional description is neither the former nor the latter, but rather the definition of the design procedures followed by designers through a functional representation of the object. This functional map will be dependent on the conceptual representation of the object according to functional dependencies among the parts it consists of. Therefore, a functional description is not considered as a general framework for the integrated treatment of behavioral and structural knowledge as in [20,22], or teleological, behavioral and structural knowledge as in [21].

Our proposal concerns a conceptual framework for the acquisition and representation of ontological knowledge, independently of the definition of knowledge for assigning specific values to the behavioral variables in question. This is because the aim is to define a conceptual description of the object that is suitable for the representation of the procedural design patterns (dynamic knowledge model) adopted by designers. Here the role of functions mainly regards the description of the design objects in accordance with the expert designer conceptualization rather than the definition of a conceptual framework for the treatment of behaviors where physical and causal knowledge must be taken into account [15].

4.2. Function behaviors and structures: the engineering design conceptual space

In order to introduce our conceptual framework, we will now outline the different design knowledge sources with reference to the FBS model, a well known conceptual model [5]. In this way, it will be possible to clarify the presented functional approach to the description of ontological knowledge in the engineering design context and the role it plays in the description of the dynamic knowledge model.

In the FBS model a design process schema is suggested as a way to understand the typical life cycle of a decision-making process in design. All the different knowledge sources described above are involved in this conceptual model as shown in Fig. 1. In this model, all the types of knowledge must be considered in close interaction, and the design activity cannot be understood out of this interaction.

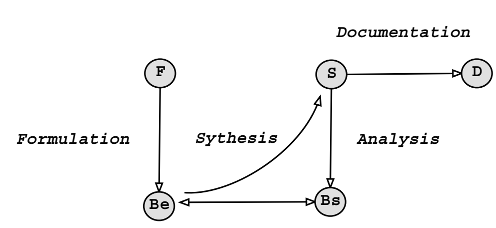

Fig. 1. FBS Model.

According to this model, three different variables define the design conceptual state space: *function variables* (F), *expected behavior variables* (Be), and *structure variables* (S). All the design activity is aimed at translating functional requirements into structures that are able to realize them. When a designer starts his/her activity, he/she considers the artefact that must be designed in terms of the functions the artefact must realize. This means to consider the artefact under the viewpoint of its intended use (see Fig. 2). Teleological knowledge is needed to translate functional requirements into expected behavior and behavioral knowledge is needed to translate the object as in the expected behavior perspective, into a specific structure able to perform that behavior. Structural knowledge is then indispensable to decide the correct configuration of the artefact in terms of topological requirements. The formulation process imposes teleological requirements on the expected behavior. The synthesis process imposes behavioral requirements on structure, and the analysis process imposes structural requirements on the expected behavior [17].

Our approach does not focus on behaviors, but on the representation of the specific design patterns followed by designers in defining the structural variables of composite objects. These patterns are in fact valuable design strategies implicitly established by designers with the aim of reducing the risk of feedbacks during the design activity (Reformulation in the terms of Gero) [21], as shown in Fig. 3. Design patterns can be regarded as heuristics of dynamic knowledge nature that are deeply connected with the designers competence in solving specific design problems.

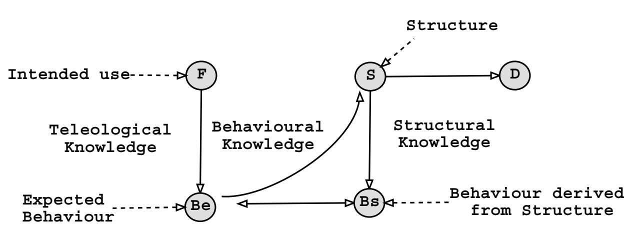

Fig. 2. FBS Model under the viewpoint of the engineering design knowledge.

From this standpoint, dynamic knowledge becomes the core knowledge that knowledge engineering must acquire and represent to develop a KBS system to support design with the aim of reducing feedbacks during the design process.

The main difference between our approach and the others introduced above consists in the different objective we associate to the functional description of the domain entities. Our framework considers the functional representation of the design object – that is the conceptual description of the design object with respect to the functional dependencies among its parts – as the starting point for the definition of a dynamic knowledge model where behaviors and goals of the object are considered as behavioral and teleological constraints on the design decision-making process.

The development of functional ontologies is thus in accordance with the aim of acquiring, representing an treating engineering design dynamic knowledge.

4.3. Philosophical background

Due to the significance of Mereology in coping with the design of complex objects, the aim of this section is to suggest a formalization of some Husserlian observations on the notions of *whole* and *part*.

In the third of his logical investigations [7], Husserl introduces the notion of whole, by means of deep theoretical discussion on the *foundation relation*; the foundation relation is a relation linking autonomous objects (or "contents"). From a logical perspective, the foundation relation can be considered as a primitive predicate symbol holding among objects of the universe.

The following logical formalization has been devised especially to support the definition of a number of significant mereological connections linking objects, whose satisfiability involves the identification of different kinds of "more comprehensive objects", also called "wholes", through the phases of a design process.

We are not interested in introducing a primitive notion of "part-of" and in finding specific *a priori* closure principles stating that the mereological domain of objects linked

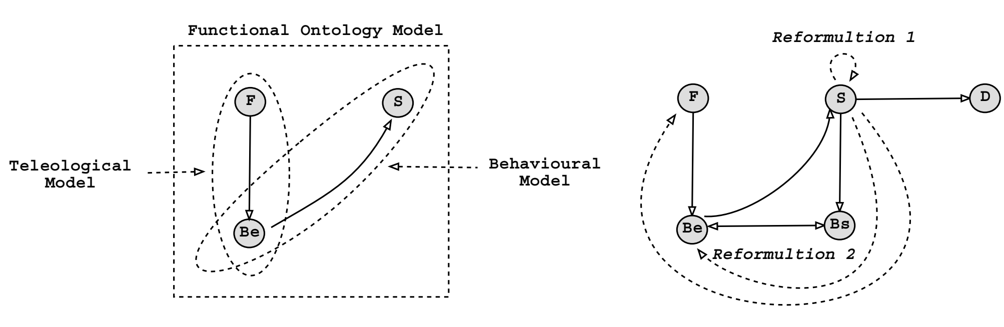

Fig. 3. As shown in Gero, there are three different kinds of Reformulation processes. Reformulation Type One implies a modification of the structural values decided for an object. Reformulation Type Two implies a modification of the behavioral requirements on structure. Reformulation Type Three implies a modification of the teleological requirements on expected behavior (design object intended use).

by that relation must be closed under certain operations (for example, the mereological *sum*, *difference*, and *complement*), which will be object of future investigation.

In fact, unlike many axiomatizations of the part-of relations that are grounded on the primitive relation of part-of ($\preceq$), or on the proper-part-of relation ($\preceq_P$), the present logical formalization starts from the introduction of a *foundation relation*.

As mentioned in [23] "The basis of a axiomatic theory of the part relation is a set D of (otherwise unspecified) "objects" – the fundamental domain or universe of the discourse".

Let $a, b, c, \ldots$ be variables ranging over the universe D. In terms of the primitive predicate FR (the foundation relation), with $FR \subseteq D \times D$, we will define the fundamental relation of "Unitary Foundation" (U) embracing objects.

Given a pair of objects $a, b \in D$, we say that there exists a foundation relation between them, and we simply write $FR(a,b)$, if a is founded on b, or $FR(b,a)$ otherwise. Since the foundation relation is *not symmetric*, Husserl distinguishes two different kinds of foundation on the basis of the nature of the relation linking objects: "unilateral" foundation (when it is the case that $FR(a,b)$ *exor* $FR(b,a)$ holds), and "bilateral" foundation (when it is the case that $FR(a,b)$ *and* $FR(b,a)$ holds).

According to the Husserl accounts, the foundation relation induces a double partition on the universe D, such that $D = FR \cup \text{not}FR$, where FR is the set of objects in D for which a unilateral or bilateral foundation relation holds, while notFR is the set of objects in D for which no foundation relation holds. Alternatively, we can state that the universe is completely determined by the union of FR and notFR. Before starting with the definitions of immediate and mediate foundation relation, we introduce the symbol $F(a,b)$, which will be useful in the sequel, as a generic abbreviation in order to bring together unilateral and bilateral foundation relations between objects in D, i.e. $F(a,b) \leftrightarrow (FR(a,b) \vee FR(b,a))$ (where, the connective "$\vee$" have to be considered as inclusive disjunction).

Definition 1 (*Mediate and Immediate Foundation Relations*). Given a pair of objects in D being in FR relation, a foundation relation can be "mediate" ($FR^M \subseteq D \times D$) or "immediate" ($FR^{\overline{M}} \subseteq D \times D$), depending on the existence of a third object between the two related objects. Let a, b and $c \in D$, the following implication establishes the meaning of the immediate foundation relation:

$$\forall a, b. \quad FR(a,b) \rightarrow (\neg \exists c. FR(a,c) \wedge FR(c,b)) \leftrightarrow FR^{\overline{M}}(a,b) \tag{1}$$

The mediate foundation relation FR^M can now be defined as the transitive closure of the immediate one, i.e.

$$\forall a, b. \quad FR(a,b) \rightarrow (\exists c. FR^{\overline{M}}(a,c) \wedge FR^{\overline{M}}(c,b)) \leftrightarrow FR^M(a,b) \tag{2}$$

Given the definitions above, it can be said also that

$$\forall a, b. \quad FR(a,b) \leftrightarrow FR^{\overline{M}}(a,b) \oplus FR^M(a,b) \tag{3}$$

where, "$\oplus$" stands for the "exor" connective.

The following is the central definition in the Husserlian foundation that leads to the notion of whole. Intuitively, a *unitary foundation relation* is said to embrace an aggregate of objects in D *if and only if*, whenever an object a is picked out of the aggregate, a foundation relation exists between this object and each of the remaining objects in the aggregate. In this sense, the unitary foundation $U \subseteq D^n \times D^n$ is interpreted as a grid structure of binary foundation relations. Formally,

Definition 2 (*Unitary Foundation Relation*). An n-tuple (or, an "aggregate") of objects $a_0, \ldots, a_{n-1} \in D$ is said to be embraced by a relation of unitary foundation if and only if, given an object a_i, with $0 \leqslant i \leqslant n-1$, it is always the case that $F(a_i, a_j)$ for all j, with $0 \leqslant j \leqslant n-1$ and $i \neq j$. Therefore, given an n-tuple $a_0, \ldots, a_{n-1} \in D$, with $0 \leqslant i, j \leqslant n-1$, where $i \neq j$:

$$U(a_0, \ldots, a_{n-1}) \leftarrow \forall a_i \exists a_j . F(a_i, a_j) \tag{4}$$

Remember that, for all $a, b \in D$, $F(a,b) \leftrightarrow FR(a,b) \vee FR(b,a)$, and $FR(a,b) \leftrightarrow FR^{\overline{M}}(a,b) \oplus FR^M(a,b)$; therefore, with reference to the above definition of the unitary foundation relation, it can be said that an aggregate of object is embraced by a unitary foundation if and only if, for each object a in the aggregate, there exists a mediate or an immediate (unilateral or bilateral) foundation relation linking a with each other of the objects in the aggregate.

In the case of certain wholes, it seems necessary to presuppose *a priori* the existence of what Husserl calls *moments of unity*. The sensible forms of unity connects some species of objects, for example: in perceiving two distinct but consequent sounds A and B, it is not only perceived the sound A, and the sound B, but also the fact the sound B *follows* the sound A. Moments of unity are not to be considered as new contents, or new objects of the same species as the objects of D. For this reason, we omit the introduction of a new set of elements of the universe, and we represent the "presence" of a moment of unity between objects of D, just as a binary relation $MU \subseteq D \times D$ holding among them. As a constraint, consider that no moment-of-unity can exists between objects that are not linked by any foundation relation (mediate or immediate, unilateral or bilateral).

Taking into account the Husserlian introduction of the relations of "independence" and "not-independence" among objects (or, in other words, the "connection" among independent objects and the "co-penetration" among the not-independent ones), it is not so difficult to exploit the presence of moments of unity in order to distinguish between objects that are "fractions" and objects that are "moments" of some whole.

Definition 3 (*Fractions and Moments*). If there exists a unitary foundation and a moment-of-unity relation ($MU \subseteq D \times D$) holding between two objects, then the two objects are said "fraction" (*Fract*(a) and *Fract*(b)); otherwise, the objects are said "moments" (*Mom*(a) and *Mom*(b)). For all $a, b \in D$, this can be represented in a very intuitive way, as

$$\forall a, b. \quad U(a,b) \land MU(a,b) \rightarrow Fract(a) \land Fract(b) \tag{5}$$

$$\forall a, b. \quad U(a,b) \land \neg MU(a,b) \rightarrow Mom(a) \land Mom(b) \tag{6}$$

where *Fract* and *Mom* are unary predicate symbols holding for objects in D, and MU is an irreflexive, and symmetric binary relation.

The generalization to the n-ary case, i.e. the case in which it is true that $U(a_0, \ldots, a_{n-1})$, is straightforward. Fractions and moments are "relative" properties of the objects. To be a fraction for an object b depends on the existence of at least another object a (distinct from b), such that unitary foundation and a moment-of-unity relations hold between a and b. Moments of unity have been represented by means of a binary predicate symbol MU, according to the idea that moments are "components of perception" that not necessarily are of the same species as a and b (Husserl refers to moments with the expression "sensible forms of unity").

Definition 4 (*Part-of Relation*). If there exists a unitary foundation linking $a_0, \ldots, a_{n-1}$ objects of the universe D, then there exists an object $c \in D$ that is a whole, and for all $0 \leqslant i \leqslant n-1$, a_i is said to be *part* of c (i.e. $\preceq(a_i, c)$).

$$U(a_0, \ldots, a_{n-1}) \rightarrow \exists c \forall a_i. \bigwedge_{i=0}^{n-1} \preceq (a_i, c) \tag{7}$$

Intuitively, the above definition of part-of relation can be read in the following way: the satisfiability of a unitary foundation among a n-tuple of domain entities $a_0, \ldots, a_{n-1}$, implies the existence of a whole c, whose parts are exactly $a_0, \ldots, a_{n-1}$. In other terms, the existence of whole depends on the "intentional" institution of a foundation relation among a collection of entities (a set, not a *multiset*, i.e. it does not admit repetitions of identical elements).

To discard transitivity of the part-of relation it is possible to make use of explicit predicate modifiers [24], or to introduce some new notation representing the predicate modified. For what concern the present proposal, the $\preceq_{\overline{M}}$ and the $\preceq_M$ relations stand for the "immediate" and "mediated" part-of relations, respectively. Obviously, both these relation are intransitive in a very natural sense.

In the third of his Investigations, Husserl makes a fundamental distinction among the objects that are identified as wholes; very briefly, it is possible to consider "first-species" and "second species" wholes. Such a distinction is made of an observation on the relationships linking objects that are parts of the whole and, after all, on the properties characterizing these objects (*Fract* and *Mom*). We introduce the distinction among objects being whole by means of two unary predicate symbols $^{\mathrm{I}}W$ and $^{\mathrm{II}}W$.

Definition 5 (*First-Species and Second-Species Wholes*). A generic whole is said to be a *first-species whole* if and only if all the objects (that are embraced by the unitary foundation associated with this whole) are moments. A generic whole is said to be a *second-species whole* if and only if all the objects (that are embraced by the unitary foundation associated with this whole) are fractions. In a formal way, this can be represented with the following expressions – given a n-tuple of objects $a_0, \ldots, a_{n-1} \in D$, an object $c \in D$, and an index i, with $0 \leqslant i \leqslant n-1$:

$$^{\mathrm{I}}W(c) \leftarrow \bigwedge_{i=0}^{n-1} \preceq (a_i, c) \land \bigwedge_{i=0}^{n-1} Mom(a_i) \tag{8}$$

$$^{\mathrm{II}}W(c) \leftarrow \bigwedge_{i=0}^{n-1} \preceq (a_i, c) \land \bigwedge_{i=0}^{n-1} Fract(a_i) \tag{9}$$

First-species wholes are characterized by the strength of the connections among their parts, and by the non-existence of moments of unity among these parts (e.g. the surface and the color of a given object). Second-species wholes are instead characterized by a weaker form of these connections, and by the existence of moments of unity (e.g. two notes in a symphony).

The non-existence of moments of unity among parts, that have been recognized as independent, relies on the specific perception one may have of the relationships among parts. According to the above observations on fractions and moments, this definition implies that a whole can be consistently considered at the same time a first-species whole, with respect a subset of its parts, and a second-species whole, with respect to another subset (not necessarily disjoint from the former).

This essentially depends on the kind of connections among its parts. In Husserlian terms, taking a whole as first or second-species depends on the subject intentionality, that is indeed a non-monotonic predication, i.e. the satisfiability of these predicates may pass through true value changes in correspondence with an increase of the subject's knowledge base.

4.4. Mereological integration of the engineering design conceptual space

In order to introduce our conceptual framework, let us start with a stereotypical example. Let us imagine the task of designing a bike. If a bike has, for instance, an outdoor destination, this means that it must give certain requirements, e.g. "to run withstanding considerable stress".

These requirements, initially expressed in terms of bike performances, are translated by the designer into a set of functions the bike has to perform through an observable and testable behavior: in this sense the bike is conceptualized as a functional system whose all parts work together in performing the desired bike behavior [16].

These parts are thus identified as functional parts. According to the FBS model, the general function that the bike must accomplish (i.e. to run under specific conditions) is expressed through teleological knowledge in a global behavior that the bike must fulfil.

This behavioral requirement implies a set of commitments on the parts of the bike that are associated to the requested use. These commitments can be considered as Functional dependencies among the parts (e.g. the Functional dependency among wheels, top frame and shock absorber). The decisions taken on structures during the bike Synthesis Phase will therefore depend on considerations about the functional dependencies among the bike parts. Design entities that participate to a functional dependency constitute a Functional Whole. We call such a whole a "Functional System". Our example manages wheels, top frame and shock absorber as entities with no parts even if, in different domains, they may also consist of parts. This means that the pinion, spokes, tires and all the elements which are in a wheel depend on the same conceptual design process we described with reference to the bike.

Therefore, since the wheel is considered as a functional whole, it imposes a number of teleological constraints on the expected behaviors of its parts, and every component of the bike will be designed according to the teleological and behavioral requirements guided by the functional whole they belong to. In this way it is possible to describe the conceptual structure of the bike with respect to the hierarchy of functional dependencies the designer recognizes on it. Moreover, performances are specific qualifications of the intended use of a function. Of course several of performances can be requested to a bike. For each requested performance a function must be associated to the bike.

When a designer must decide the structure of a pinion, he/she will consider both the behavioral commitments arising from the wheel and the teleological commitments arising from the intended use of the whole bike. Moreover, the structure of each part of the bike is naturally limited by structural bonds. With reference to our example, when a wheel is going to be designed, all its functional parts should be decided on the basis of behavioral and structural commitments.

This implies a decision about a specific design sequence among these parts. In fact it is possible that the choice about some tire structural variables come into conflict with some others structural variables (e.g. Spokes) because of adjacency and dimensional values or, more in general, of topological aspects (e.g. the adjacency between Spokes and Reflectors).

To manage the explosion of functional and structural constraints is one of the hardest tasks in developing a KB-System supporting design activities. Our aim is thus to stress the relevance of a mereological description of the design object (expressed in terms of functional dependencies) as an intermediate phase between Formulation and Synthesis processes. Knowledge engineering has to deal with design patterns which belong to teleological commitments and patterns which belong to behavioral and structural commitments, and it has to find out formalisms that are suitable to the represent them. Our model considers functional knowledge as a bridge among performances, behaviors and structures that are indispensable dimensions along which the design process takes place.

5. Towards the dynamic knowledge representation: functional dependencies

As shown in [17] a function can be performed by more than one object structure, and the behavior (of some structure) can provide more than one function: multiple functional roles performed by a design object part are often responsible for reformulation processes [21]. In our example, the structure of the spokes performs a *Support Function* with respect to the wheel, and a *Sustaining Function* for reflectors.

The design decisions on spoke structure must therefore take into account functional requirements arising both from the wheel and from the reflector. From one hand the structural decisions on reflectors are motivated by the reflector expected behavior, that has been determined on the basis of the intended bike use, and by the spoke's structure. On the other hand, structural decisions on spokes could be influenced by reflector too. According to the conceptual descriptions discussed above let us now introduce some conceptual relations among the introduced functional parts (see Fig. 4).

Has-a relation connects one structural element to its specific function. This relation expresses the intended use of an element whenever teleological requirements on expected behavior must be taken into account. The *Need-a* relation connects functions: it expresses the functional dependencies among elements where behavioral conditions on structures must be taken into account. Thus, the *Need-a* relation establishes that the correct structure of the elements involved in the functional dependency cannot be stated without taking into account behavioral requirements coming from the intended use of the respective Functional System. Connected to the consideration about the multiple functional roles performed by design object parts, we introduce the *Perform-a* relation, in order to explain, for example, that spokes offer a sustain to the reflector positioning.

Up to now, we have taken into account only those situations where the *Has-a* relation was a one to one relation. In the context of engineering design this is a simplification: a function could be provided with more than one structure/behaviour [17]. In our example it is possible that a Lighting function is performed by two different objects. When this is the situation, the same function is connected to distinct entities as in the case of reflector and headlight. Thus, it is possible that one of this objects does not need a sustaining function to satisfy its goals. For this reason we introduce in our conceptual framework the *Want-a* relation for expressing a functional request that has to be satisfied by the structure of some other design object – as in the case of Reflector and Sustaining Function it wants – (see Fig. 5).

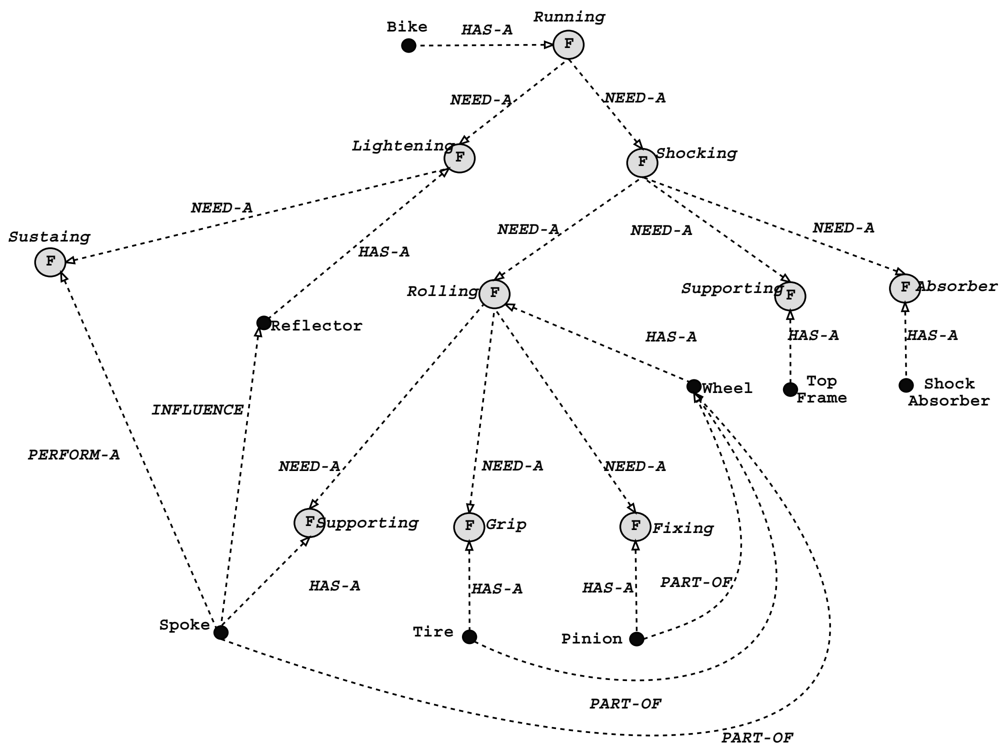

Fig. 4. Mereological conceptual relations.

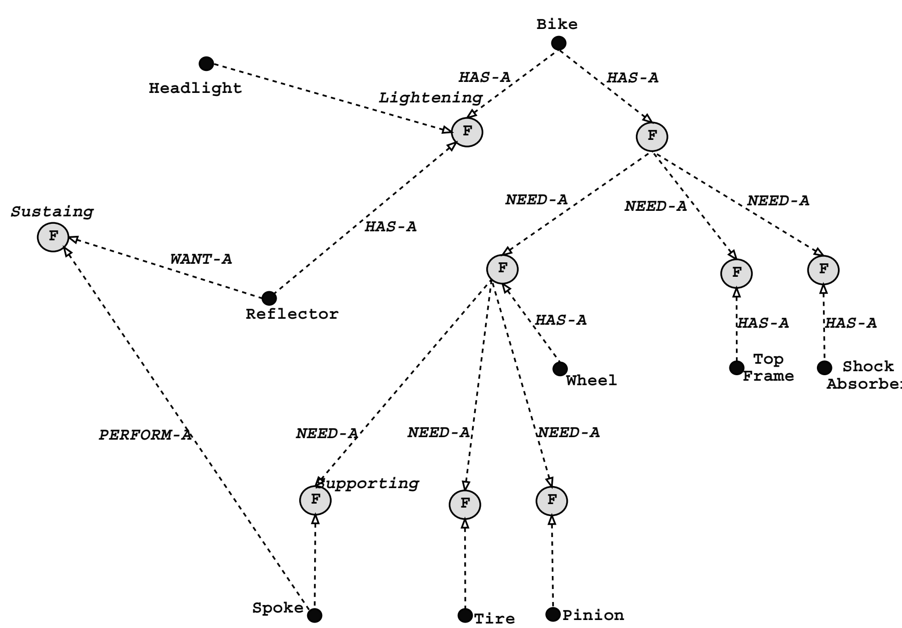

Fig. 5. The *Perform-a* relation.

Similar considerations (see Fig. 6) are on the basis of the introduction of the *Influence* relation whenever more than one functional element is in a *Perform-a* relation with one function. Thus, *Influence* relation defines a mutual functional bond on the structure of the connected parts.

Design elements linked by *Has-a* and/or *Perform-a* relations are said Functional Systems (see Fig. 7).

5.1. Functional ontology model under a phenomenological viewpoint

The functional ontology model aims at representing a twofold conceptual view on the same design object, where each element can be considered with reference to its function viewpoint (if teleological and behavioral requirements

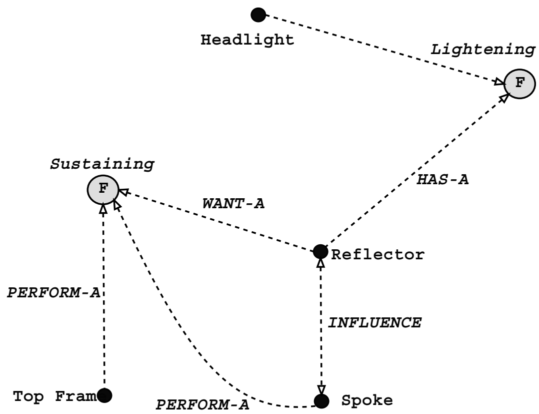

Fig. 6. Introducing the *Influence* conceptual relation.

must be taken into account) or with reference to its structure (if structural requirements must be considered during the design activity). As a consequence, behavioral requirements on structure and teleological requirements on expected behaviors must be taken into account. Thus, structural knowledge on design object is put in action with the aim of meeting topological constraints on the design object parts (see Fig. 7).

In the next section, we will describe how this conceptualization could be profitability adopted to acquire and represent dynamic knowledge: we will briefly present the mereological meaning we ascribed to the introduced relations with respect to FBS model and with reference to the formal definitions in Section 4.3. The *Has-a* and *Perform-a* relations express the role of teleological knowledge in conceptualizing the design object in terms of expected behaviors. These relations link a Structure to the Function they realize.

According to Husserl, a teleological linkage or an *intentional act* define a relation of foundation without a moment-of-unity, where each of the correlated elements is unthinkable without taking into account the other one (in the case for example of the surface-color relation: it is not possible to think at a color without looking at the surface it will cover, and viceversa). Functional Systems are thus defined as First-Species Wholes and their parts are respectively a Functional Moment and a Structural Moment. Functional Systems are grounded on teleological knowledge.

On the other hand, the *Need-a* relation aims at modelling each functional moment with respect to its functional dependencies. As a result of this functional modelling – that is grounded on functional knowledge – a mereology of functions can be described. This mereological representation is necessary to decide the correct working of the design object where behavioral knowledge (physical and casual knowledge) is needed (Fig. 7). This functional schema allows to make explicit, for each Functional Moment, the set of functions that are recognized by designers as ways to achieve the goal represented by that Functional Moment. Each of these functions, differently from all the elements in a "Functional System", is always thinkable independently (e.g. it is always possible to think at a grip function without thinking at a supporting function, as shown in Fig. 4).

For this reason they are considered in our phenomenological perspective as Functional Fractions of the Functional Moments: according to formal definitions, Functional Fractions define Second-Species Wholes. Behavioral knowledge founds the kind of Second-Species Wholes. We refer to them as Behavioral Systems. Functional Moments without Functional Fractions are said Functional Elements. Moreover, recursively each function is teleologically connected to its proper structure through a *Has-a* or a *Perform-a* relation (see Fig. 7).

On this basis, it is possible to recognize different kinds of Wholes which populate a functional ontology according to the different knowledge sources:

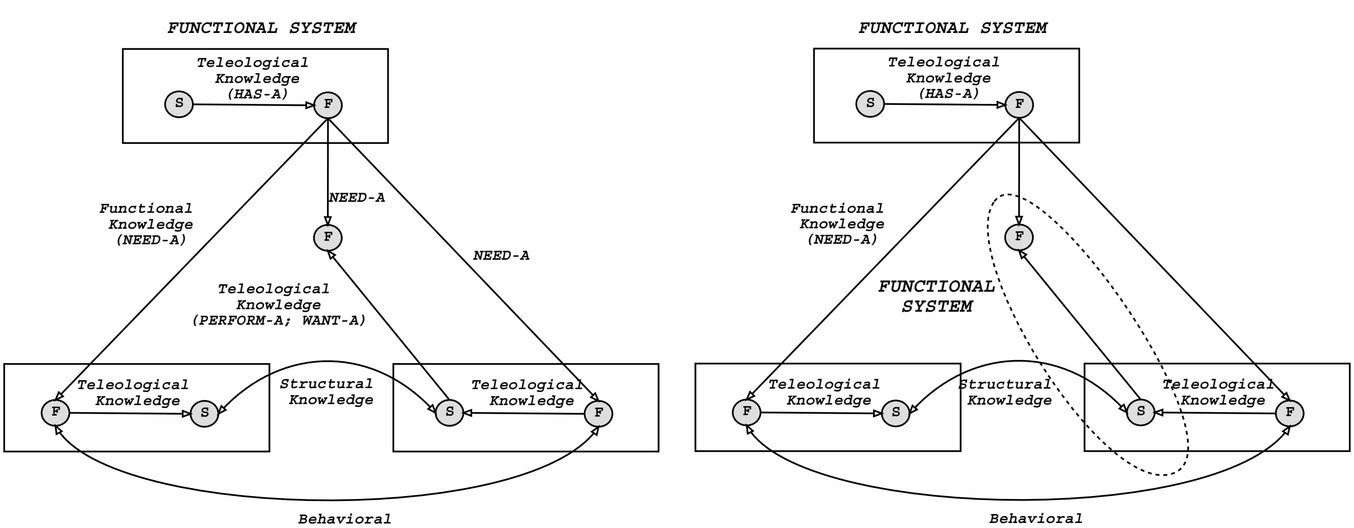

Fig. 7. This set of mereological relations are used to model First-Species Wholes.

- First-Species Wholes (Functional Systems described by the *Has-a* relation) characterized by relation of foundation is guided by Teleological knowledge.
- First-Species Wholes whose moments are Second-Species Wholes, where teleological knowledge represents a requirement on behaviors and structures of the design object. They express what is normally called Expected Behavior in the FBS model.
- Second-Species Wholes where the relation of foundation is guided by Structural and Behavioral knowledge (Functional Aggregates and Behavioral Systems).
- Second-Species Wholes whose fractions are First-Species Wholes, where the relation of foundation is managed by Teleological knowledge and the same structure accomplishes different functions (e.g. the relations expressed by *Perform-a* (see Fig. 8).

In Fig. 9, it is shown a synthesis of these mereological conceptualization with reference to the bike example.

Thinking at artifacts in terms of Second-Species Whole means to build *matter ontology* and *device ontology* where the principles by means of which the relationships among object parts are introduced is founded on topological and behavioral issues. These kind of ontological representation are typically *bottom-up*.

On the contrary thinking of artifacts in terms of First-Species Whole implies to create a functional representation as in our framework where *top-down descriptions* are more important. This distinction can be considered as the starting point for the definition of a formal theory on composite design objects in the context of engineering design (where both First-Species – Functional Systems – and Second-Species Whole structural aggregates – must be managed).

In this way, it will be possible to start a study of a mereological theory for the acquisition and representation of functional knowledge by means of which to define a methodology for the development of KB-System supporting engineering design core knowledge. In the next section, the relevance of these relations under the perspective of the dynamic knowledge will be analyzed.

5.2. Dynamic knowledge representation

The topic of Dynamic knowledge representation is well known and it has been widely tackled in the literature on automatic planning and configuration. Here, we do not deal with these topics because we are more interested in defining a conceptual framework for the acquisition of dynamic knowledge coming from experts rather than in developing models for the automatic resolution of design problems through the application of problem solving methods [15].

As previously observed, one of the main problems of knowledge engineering concerns the correct specification of all design patterns followed by expert designers of specific engineering domain. The design patterns adopted by experts indicate which parts of the object should be designed *before*, which should be designed *after* and which can be designed *in parallel* with others. The adoption of these design patterns is particularly important in the case of complex objects design activities where the goal is the reduction of side effects triggered by decisions about one part of the object.

With respect to the FBS model, these side effects are founded on the circumstance that the behavior effectively derived from a structure does not satisfy the expected behavior [21]. We have stated above that a way for the representation of design patterns can be the functional representation of the design object. Here, we want to show how the conceptual relations previously presented can be helpful in the acquisition of dynamic design knowledge. In this section, it will be described how a well known

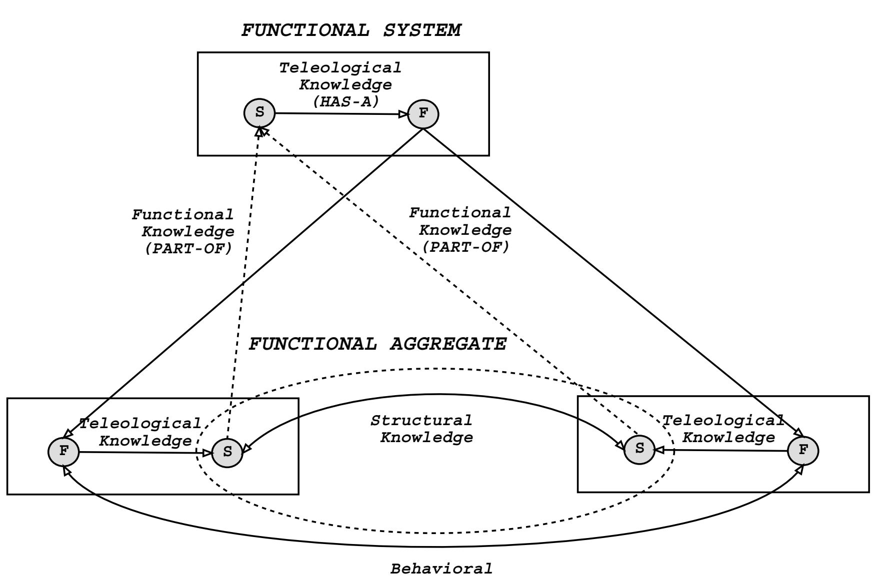

Fig. 8. Functional Aggregates defined on the basis of the *Part-of* relation where structural knowledge is required to design them.

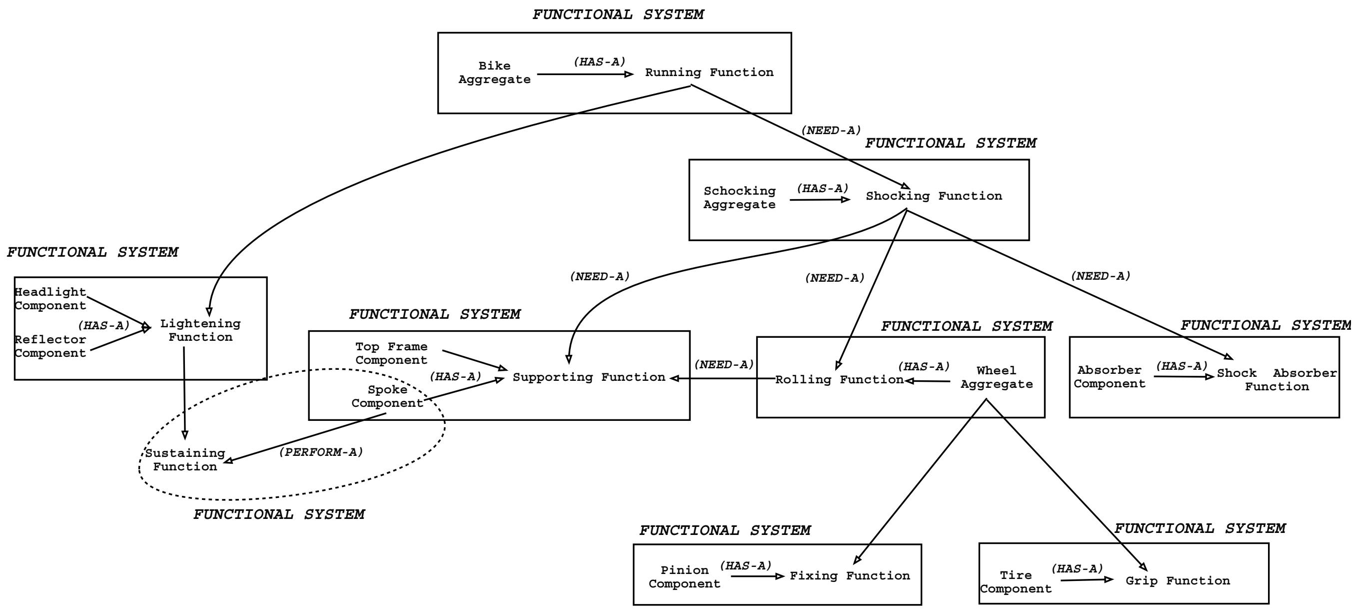

Fig. 9. Functional Aggregates and Functional Systems.

formalism for the study of non-sequential processes, the SA-Nets, can be profitability adopted for representing design patterns according to the functional description of the design object.

Superposed Automata (Occurrence net) is a subclass of Petri Nets defined in the frame of languages for the analysis and design of organizational systems, and for the study of non-sequential processes [25]. SA-Nets are a state transition formalism, whose simpler elements are "arcs", "states", and "transitions". A formal presentation of this language is out of the aim of this paper, but the primitive elements of the formalism will be informally described exploiting the bike example. The only remarkable syntactic condition we have to take into account before starting the presentation is that the number of arcs incoming to and outcoming from each transition must be equal.

In our extension of SA-Net formalism, named SA*-NETs [26], two different kinds of state are introduced, that represent the Functional Components and Functions (Fig. 10), respectively. The states which indicate Functions are of two types: the START State and the END states. The first expresses the starting point of the design phase, while the latter expresses the conclusion of the design activity connected with that function. SA*-NET elementary process is a before/after sequence. In our framework SA-elementary processes can be stated:

- Between Functional Components and End State.
- Between Start State and Functional Component State.
- Between End State and Start State.
- Between End State and End State.

For synchronizing two transitions it is necessary to satisfy the constraint on the balancing between incoming and outcoming arcs. Fork and Join Transitions are adopted to manage processes which can be stated without priority dependencies. Hook process describes the feedback (Reformulation processes in the terms of FBS Model). The main idea in adopting this formalism for the representation of engineering design dynamic knowledge is to use SA*-NET transitions system for the definition of the design patterns. Thus an SA*-NET Model aims at representing the design sequences among the functional components which constitute the functional partonomy [24] described above.

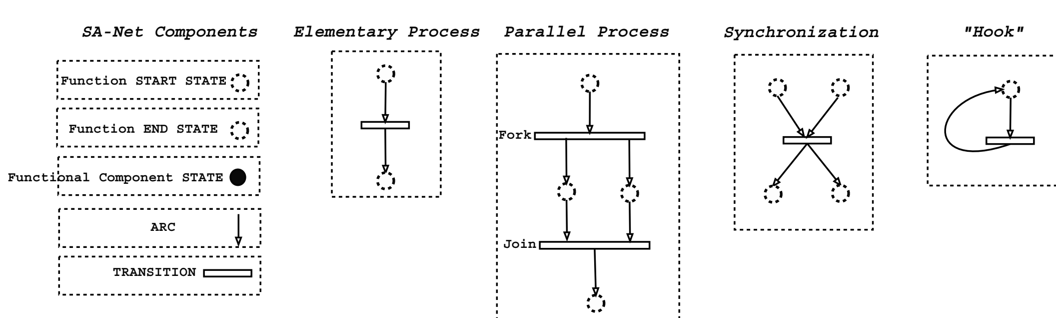

Fig. 10. SA*-NET formalism.

In the following, we will explain our conceptual framework for the acquisition and representation of engineering design dynamic knowledge through the bike example: it will be shown how the SA*-NET model can describe both procedural dependencies among functional parts and functional dependencies on functional components.

As shown in Fig. 11 the SA*-NET states which represent a Functional Component are always preceded by the states which represent the functional conditions involved in the decision of its structure. According to the description of the ontological model introduced above, in this case the decision on the Top Frame structure depends on behavioral requirements coming from the Support function. Otherwise, Top Frame as Support function component is part of the Shocking Functional Aggregate: this means that the behavioral requirements on its structure are strictly dependent on the teleological requirements coming from the Shocking Functional System. The use of Hook processes in this case has the aim to represent that the Supporting expected behavior could be modified under Shocking functional requirements. The synchronization between Supporting and Shock Absorber means that Top Frame must be designed before Shock Absorber. The State END Absorber cannot fire since END Supporting State is not marked.

However, not all the functional parts of the artefact must be designed according to a before–after sequence: it is possible that some functional parts belonging to a given Functional System could be designed in parallel. The B side of Fig. 11 shows that the Functional Component Pinion should be designed before the Functional Components Spoke and Tire. There are then cases where the design of a functional component inhibits the design of some others. With reference to the previous considerations about the functional ontology framework, this happens when different functional components are in *Has-a* relation with the same Function (i.e. in the bike, reflector and headlight, as shown in Fig. 11).

Up to now we have talked about Functional Components belonging to a same Functional System. Besides this, structural conflicts may occur among functional parts which belong to different Functional Systems. In this case, a synchronization must be introduced to point out that the structure of the Functional Component being designed cannot be definitively decided without checking some other structures belonging to different functional systems. SA*-NETs appear particularly suitable in the representation of these structural conflicts among functional components of the design object because they derive from a Petri Nets' subclass that has been developed for the study of

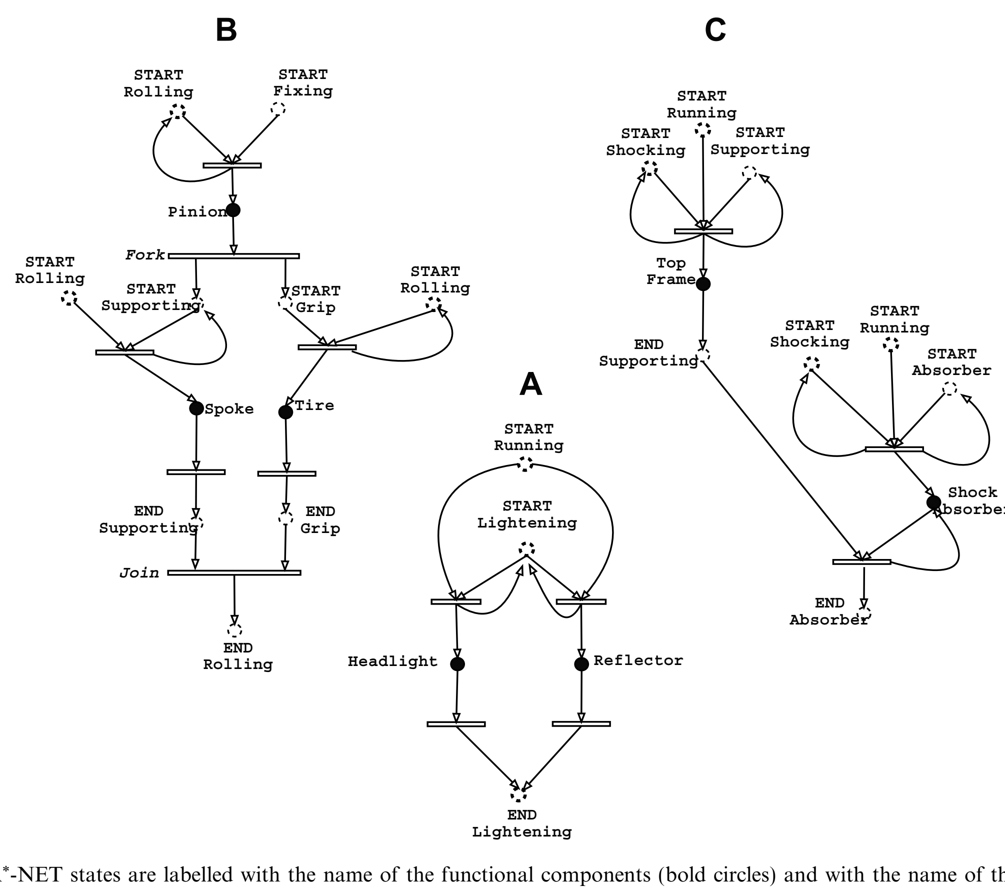

Fig. 11. Part C: SA*-NET states are labelled with the name of the functional components (bold circles) and with the name of the Functional Systems associated to the functional components (blank/dotted circles). Part B: Spoke and Tire are functional components that can be designed in parallel. Part A: Headlight and Reflector cannot be both designed.

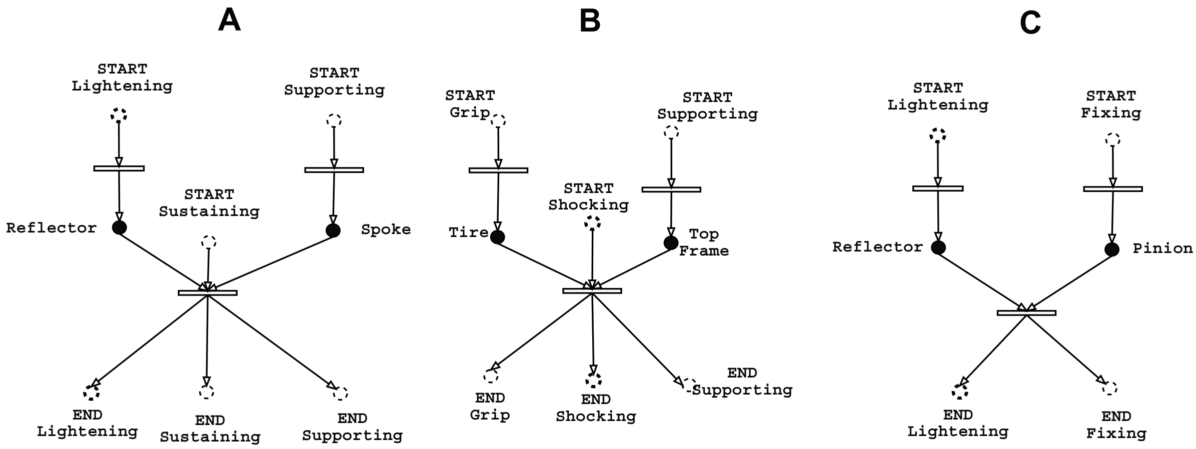

Fig. 12. Part A: The synchronization between Spoke and Reflector depends on the *Influence* relation between them. This influence is originated by the fact that spoke offers sustain for the positioning of reflector; Part B: The synchronization between Top Frame and Tire is motivated by the fact that Tire belong to the Rolling Function System; Part C: In this case no Functions enter in the synchronization because only structural/topological constraints must be taken into account.

non-sequential processes. Here we use transitions composition for representing structural conflicts among the functional parts of the design object.

On the basis of this conceptualization, in the engineering design context Reformulation Processes can be motivated on the basis of:

(1) Functional influence reasons such as the above discussed case of Reflector and Spoke Functional Components (see the A side of Fig. 12).
(2) Behavioral requirements (see the B side of Fig. 12) which depend on the fact that a Functional Component belonging to a Functional System is part of a Functional Aggregate belonging to another functional systems (i.e. Tire is a functional component of the Rolling Function System which is the Wheel Functional Aggregate).
(3) Structural requirements which depend on topological motivations. In the bike example it is reasonable that because of the adjacency between the pinion and the reflector a structural conflict can happen (see the C side of Fig. 12).

During the design activity expert designers take into account all these potential structural conflicts, and on this basis they decide for the most suitable design process. Our framework, through the integration of ontological knowledge and dynamic knowledge, aims at developing Knowledge-Based Systems for supporting the reduction of Reformulation processes in the sense of FBS model. Fig. 13 shows the entire SA*-NET analyzed above.

6. Conclusions and future work

This paper has presented a framework for the representation of knowledge involved in the design of complex mechanical objects. Such objects are generally configured exploiting CAD tools, but the need for new solutions able to support designers in their daily activities to preserve time and money is becoming a very important research trend. In this direction, artificial intelligence techniques are useful: in particular, this paper has shown how the adoption of Knowledge-Based Systems' approach has allowed to clearly identified the different kinds of knowledge involved: static, dynamic and experiential. Our work has been devoted to capture all these aspect of the knowledge involved in the design of complex objects into a unique framework. In this sense, we have adopted the FBS model by Gero as a suitable starting point. With respect to traditional methodologies for the development of Knowledge-Based Systems, like KADS [27] and MIKE [28], the FBS model is more specific for the development of applications supporting experts in the design of complex mechanical objects, since it focuses on the functional aspects of the different components of the product. A methodology like KADS, in our opinion, is too generic to understand the real and heterogeneous nature of the knowledge involved in the different design steps in manufacturing of mechanical objects.

Two main results could be achieved through the usage of this conceptual framework in the different knowledge engineering phases. From one hand, it provides knowledge engineers with a Top-Down Composite Functions Model that is a conceptual instrument able to facilitate the modelling of heuristic knowledge defined in according to the effective way of working testified by engineering designers. On the other hand, the integration between SA*-NETS and ontologies supports knowledge engineers in better understanding how an expert designer thinks about design steps of an object according to the functions it must satisfy.

The framework has been applied in the design and implementation of the IDS project [26], a collaboration between the University of Milano – Bicocca and Fontana Pietro S.p.A., an Italian enterprise leader in design and

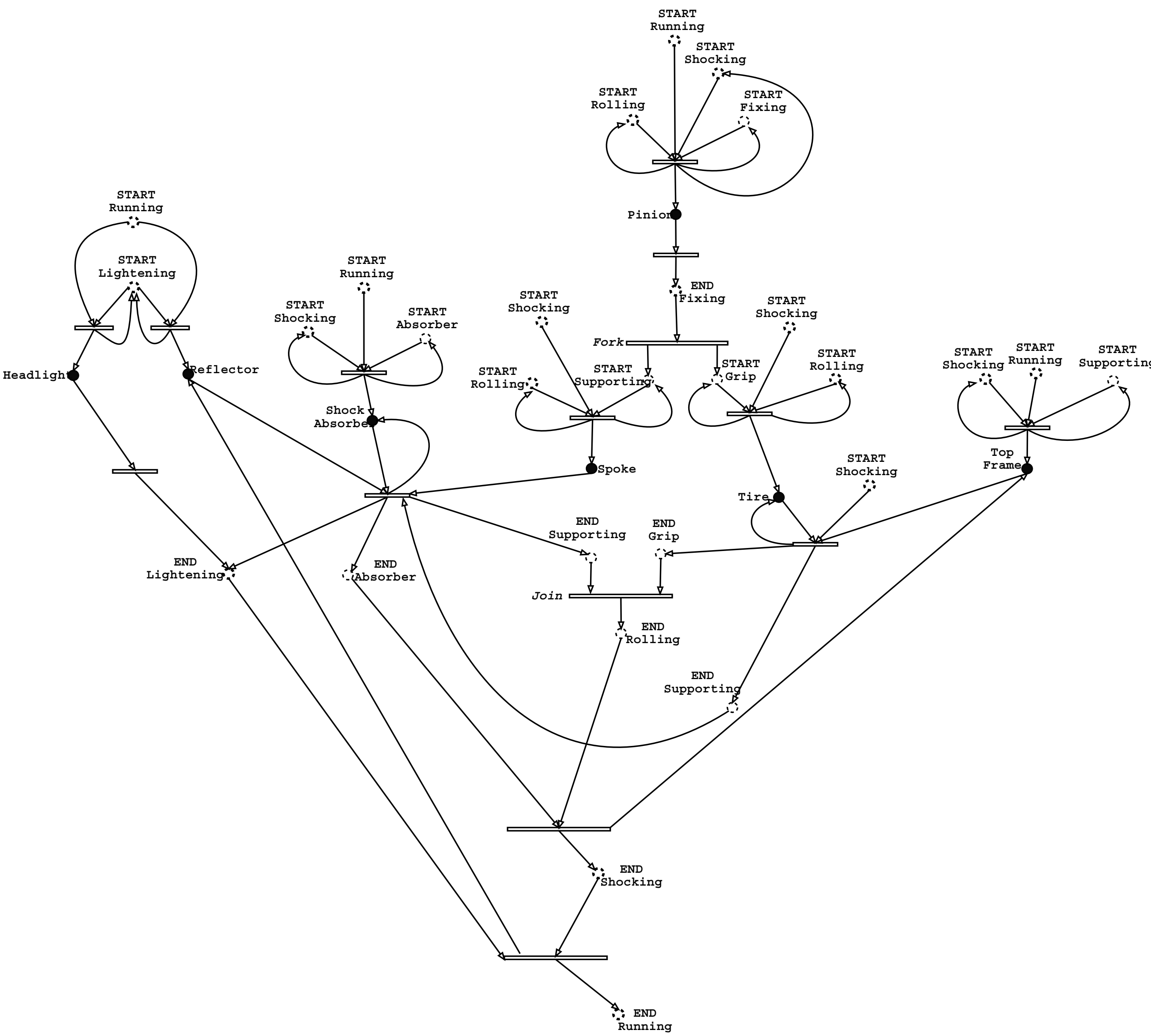

Fig. 13. For each phase behavioral and teleological requirements must be taken into account by designers.

manufacturing of dies for car bodies, where a Knowledge-Based System integrated with CATIA V5.0 (i.e. the CAD tool adopted by Fontana Pietro designers) has been developed to automate partially some phases of the die design process. The framework described above has been used as both a useful guide during the knowledge acquisition sessions and the knowledge representation of main aspects of die design and manufacturing. Moreover, it has allowed to build a model of the die that is shared among all the designers of Fontana Pietro, with significant benefits from the knowledge sharing and maintenance point of views.

Anyway, the framework is general enough to be applied in the design and implementation of Knowledge-Based Systems to support the design of configurable objects in several domains. In particular, the application to the domain of chemical blends production (e.g. rubber compounds, medicines) has been already investigated as a very promising direction of future works.

References

[1] F. Puppe, A Systematic Introduction to Expert Systems, Springer-Verlag, Berlin, 1996.

[2] D.C. Brown, M.B. Waldron, H. Yoshikawa (Eds.), Intelligent Computer Aided Design, Proceedings of the IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design (IntCAD91), Columbus, OH, USA, 30 September–3 October 1991, IFIP Transactions, vol. B-4, North-Holland, 1992.

[3] R. Sriram et al., Knowledge-based System Application in Engineering Design Research at MIT, MIT Press, 1989.

[4] J. Gero, M.L. Maher, A framework for research in design computing, in: 15th ECAADE – Conference Proceedings, 1997.

[5] J.S. Gero, Design prototypes: a knowledge representation schema for design, AI Magazine 11 (4) (1990) 26–36.

[6] M.L. Maher, B. Balanchandran, D.M. Zhang, Case-based Reasoning in Design, Lawrence Erlbaum Associates, 1995.
[7] E. Husserl, Logische Untersuchungen. Zweiter Band. Untersuchungen zur Phnomenologie und Theorie der Erkenntnis, Niemeyer, Halle, 1900/1901 [second ed. 1913; Logical Investigations (J.N. Findlay Trans.), vol. 2, Routledge & Kegan Paul, London, 1970].
[8] N. Guarino, Some ontological principles for designing upper level lexical resources, in: Proceedings of the First International Conference on Lexical Resources and Evaluation, Granada, Spain, 1998.
[9] Y. Deng, Function and behaviour representation in conceptual mechanical design, Artificial Intelligence Engineering Design Analysis Manufacturing 16 (2002) 343–362.
[10] Y. Kitamura, T. Sano, K. Namba, R. Mizoguchi, A functional concept ontology and its application to automatic identification of functional structures, Advanced Engineering Informatics 16 (2) (2002) 145–163.
[11] R. Bracewell, K. Wallace, Designing a representation to support function-means based synthesis of mechanical design solutions, in: Proceedings of ICED 01, 2001.
[12] Y. Umeda et al., Supporting conceptual design based on the function-behavior-state modeler, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10 (1996) 275–288.
[13] Y. Umeda, T. Tomiyama, Functional reasoning in design, IEEE Expert: Intelligent Systems and Their Applications 12 (2) (1997) 42–48.
[14] G. Pahl, W. Beitz, Engineering Design – A systematic Approach, The Pitman Press, 1988.
[15] B. Chandrasekaran, A.K. Goel, Y. Iwasaki, Functional representation as design rationale, IEEE Computer 26 (1) (1993) 48–56.
[16] L. Chittaro, C. Tasso, E. Toppano, Putting functional knowledge on firmer ground, Applied Artificial Intelligence 8 (1994) 239–258.
[17] D. Brown, B. Chandrasekaran, Knowledge and control for a mechanical design expert system, IEEE Computer 19 (7) (1986) 92–100.
[18] I. Nonaka, H. Takeuchi, The Knowledge creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, 1995.
[19] B. Wielinga, G. Schreiber, Configuration-design problem solving, IEEE Expert: Intelligent Systems and Their Applications 12 (2) (1997) 49–56.
[20] Y. Kitamura, T. Kasai, R. Mizoguchi, Ontology-based description of functional design knowledge, Advanced Engineering Informatics 16 (2) (2002) 145–163.
[21] J. Gero, Computational models of creative designing based on situated cognition, in: C&C'02: Proceedings of the Fourth Conference on Creativity and Cognition, ACM Press, New York, NY, USA, 2002, pp. 3–10.
[22] R. Mizoguchi, Tutorial on ontological engineering – Part 3: Advanced course of ontological engineering, New Generation Computing 22 (2) (2004) 198–220.
[23] N. Rescher, Axioms for the part relation, Philosophical Studies 6 (1955) 8–11.
[24] A. Varzi, Stanford Encyclopedia of Philosophy, The Metaphysics Research Lab Center for the Study of Language and Information, Stanford University, CA 94305-4115, 2004, Ch. Mereology.
[25] C. Girault, W. Reisig (Eds.), Application and Theory of Petri Nets, Informatik-Fachberichte, vol. 52, Springer, 1982.
[26] S. Bandini, F. Sartori, Industrial mechanical design: The ids case study, in: J. Gero (Ed.), Proceedings of Second International Conference on Design Computing and Cognition, Springer-Verlag, 2006.
[27] H. Akkermans et al., Commonkads: A comprehensive methodology for KBS development, IEEE Expert (1994) 28–37.
[28] J. Angele, D. Fensel, D. Landes, R. Studer, Developing knowledge-based systems with mike, Automated Software Engineering 5 (4) (1998) 389–418.