Feature-Based Modelling and Information Systems for Engineering

Emilio M. Sanfilippo[1,2](✉) and Stefano Borgo[1]

[1] Laboratory for Applied Ontology (LOA-ISTC),
National Council of Research (CNR), via Alla Cascata 56/C, Povo,
38123 Trento, Italy
sanfilippo@loa.istc.cnr.it
[2] Ph.D. School in ICT, University of Trento, Trento, Italy

Abstract. We use methods based on ontology engineering to individuate the shortcomings of feature-based modelling approaches in product lifecycle data management, and propose an alternative view.

Our aim is to contribute to the development of information systems for the integrated management of product lifecycle knowledge. In particular, we are looking for suitable approaches to model the variety of engineering features as used in intensive knowledge-based product development tasks, in particular dealing with manufacturing and engineering design.

Keywords: Feature · Manufacturing · Ontology engineering · Design

1 Introduction

Product development is a knowledge intensive task in which several teams interact at different times and from distributed geographic places by using heterogeneous computer modelling systems [1]. In order to be machine-processable and cognitively transparent to software agents and to the variety of stakeholders, product knowledge has to be represented in computational languages, with formal semantics, and driven by experts' conceptualisations.

Traditional computer-based technologies for product data modelling, like Computer-Aided Design (CAD) systems, as well as conceptual and data models for engineering are mainly focused on geometric specifications of product knowledge. Nowadays, however, experts need to represent and share qualitative knowledge about the product at hand, that is, knowledge concerning the engineering intents, like functional and material knowledge as well as constraints on machining tools, product management and costs [2]. The quest to add qualitative knowledge into quantitative product models led in the 1970s to the development of *feature-based* product modelling approaches and technologies [3].

Much of the research work in this area has been focused on the development of algorithms for the automatic detection of features in design models to allow the integration of CADs with downstream applications like Computer-Aided

M. Gavanelli et al. (Eds.): AI*IA 2015, LNAI 9336, pp. 151–163, 2015.
DOI: 10.1007/978-3-319-24309-2_12

Manufacturing (CAM) and Computer-Aided Process Planning (CAPP) systems. This has stimulated the development of Artificial Intelligence-based methods, among which knowledge-based expert systems for the automatic generation of manufacturing process plans from a set of input constraints [4].

Despite the amount of work, the use of feature-based technologies is hampered by the lack of a robust methodology for feature representation. Ontology engineering approaches are being actively exploited for product development purposes but even in this case the lack of a shared framework has lead to a number of disconnected and application-based ontologies that deal with feature-based applications in very different ways. Today's engineering ontologies concentrate on formal representations of the concepts for specific application requirements without attempting a deep characterisation of their meaning according to experts' conceptualisations, i.e., giving up to cross-community interoperability.

We aim to fill this gap. The development of information systems is a complex engineering process and the task we are concerned with, namely the formalisation of a broadly applicable knowledge base framework for CAD/CAM integrated systems, requires to systematically analyse the concepts at stake, and that of feature foremost, before moving into application concerns.

The paper is organised as follows. In Section 2 we provide a quick overview of feature concepts as used today in product modelling. The state of art of feature models in engineering is given in Section 3. The problems in existing feature-based modelling approaches are discussed in Section 4. The ontological analysis and formal representation of the notion of feature are described in Section 5 and Section 6, respectively; Section 7 adds an example.

2 Features for Product Modelling

Feature-based systems have represented an evolution of computer-based geometric modelling approaches since the 1970s, and are nowadays the prevalent approach for computer-aided product development. These systems provide support for product data modelling behind the specification of geometric constraints by managing product lifecycle information required during the different stages of product development [5]. In particular, features are used to represent and reason over multiple quantitative and qualitative aspects of product lifecycle, spanning from geometry to e.g. functional information, manufacturability constraints, production costs and material tolerances. Feature-based approaches have stimulated the development of expert systems for engineering design and manufacturing purposes, as well as concurrent and collaborative modelling environments for different product development tasks [6].

Historically, much of the work in the area of feature-based modelling focused on the so-called *geometric features*, namely shapes recurrently used in engineering projects like counter bore, slot, chamfer and rib. This focus broadened over the years leading to the introduction of qualitative feature information typically based on specific requirements, e.g., non-geometrical information needed for design applications, manufacturing process planning or mechanical stress analysis [3]. As a consequence of this variety, feature-based models and terminologies tend to be driven by application concerns, that is, the information attached to the identified feature is tuned to either the product lifecycle phases at stake, or to the application domain in which their use is considered [7].

Consider, for instance, the manufacturing and the engineering design domains. In manufacturing one of the main application concerns of the feature-based approaches is the creation of process plans according to design specifications [8]. In CAPP applications, the design model of the part under consideration is analysed to find the most appropriate solution for its manufacturing. In these systems, machining method, tool access direction, workpiece set up constraints, among other information, is attached to geometric features, giving rise to the so-called *manufacturing feature* [4,9]. These are understood as portion of material to be removed (subtractive feature), or added (additive features) to obtain the desired final geometry. For instance, a hole feature is the volume removed by a drilling cutter [9]; if the cutter penetrates the material frame, resulting in a set of circularly connected inner boundaries, the feature is a *through hole*; if the cutter does not penetrate the frame leaving a base face, the feature is a *blind hole* [10]. In the case of engineering design, among other feature types, the so-called *functional* features are particularly used to merge information on a geometrical shape with details concerning its purpose(s) and expected behaviour(s) within a certain product [11]. A pocket, for example, is a functional feature when it has the function to allow a certain assembly constraint to hold.

Other research communities broaden the meaning of feature in other directions, for instance, aiming to merge shape information with product's characteristics and sub-assemblies. Groover [12, p.634] defines product features as "the characteristics of a product that result from design". Similarly, Brown [11] considers features as things like product's colour, mass, portions of surfaces, etc.

In the area of civil engineering, Nepal et al. [13] take features to be "meaningful real world entities to which one can associate construction-specific information" [13, p.13]. Along the same lines, in mechanical engineering, Anjum and colleagues [14] consider physical items like metal components (e.g. screws) as features for assembly purposes.

3 Features in Engineering Models

Several initiatives focus on the development of feature specifications (data modelling standards, computational ontologies, taxonomies) for disparate applications within the product lifecycle information modelling.

The ISO standard *Automation systems and integration–Product data representation and exchange*, commonly known as STEP (ISO10303) [15] is considered the most relevant effort towards the standardisation of product data across the entire product life-cycle. Within STEP, AP224 is an application protocol dedicated to feature-based product modelling. It specifies recurrent shapes used in manufacturing scenarios. At the core of the AP224 is the concept of manufacturing feature, meant as volume of material to be removed and that results from machining. STEP provides a classification of several feature types, which are employed in various research projects and modelling systems [16].

Ma and colleagues [6] proposed to look at features as general modelling elements resulting from the aggregation of geometric and non-geometric parameters. Their purpose is to provide a layout for feature data specifications in the form of a schema specifying the type of the data to be included for feature representation. The key advantage of their approach is to provide a general and adaptable method for feature data specification, by which the commonalities and differences between different representations can be checked while remaining independent from specific application domains.

Different research communities have proposed to use computational ontologies for feature-based product knowledge representation and data sharing between CAD systems, to facilitate the integration of CADs with downstream applications like CAM and CAPP systems, as well as to provide formal tools for feature recognition and manufacturing verification. For example, the *Core Product Model* (CPM) ontology represent an engineered product as the aggregation of form, function and feature, where the latter is meant as "a subset of the form of an object that has some function assigned to it" [2]. The CPM is reused across different research projects. Dartigues et al. [17] extend it to the integration of CAD/CAPP systems. Their *Feature Ontology* is formalised in KIF.

The *Common Design-Feature Ontology* (CDFO) is an OWL ontology for feature-based CAD models exchange [18]. Feature classes are extracted from CAD systems like Catia V5, Pro/Engineering, SolidWorks and classified into a taxonomy.

The *Manufacturing Core Ontology* (MCCO) was presented by researchers at Loughborough University [14] as a common semantic foundation for modelling and sharing manufacturing knowledge. The concept of feature, meant as "a distinctive attribute or aspect of something" plays a key role within MCCO, because manufacturing operations and tools information is attached to the part to be manufactured with respect to its geometric features. The ontology is specified in Common Logic.

Kim et al. [1] proposed a classification and OWL/SWRL formalisation of assembly features to automatically reason over product knowledge, to reuse assembly models and to facilitate data sharing across applications. The classification is enriched with classes about manufacturing processes, products and materials, among others, so that it can be used to foster CAD/CAM integration.

Recently, Wang and Yu [10] proposed a feature ontology split in two modules, the STEP Box and the Feature Box. The former consists of a partial OWL formalization of ISO10303-AP203. The latter is a feature library that describes features as combinations of the STEP Box elements using OWL axioms and SWRL rules. The authors show how their system is able to automatically recognise a number of STEP features in design models.

4 Bottlenecks of Feature-Based Modelling Approaches

Despite the amount of work in the engineering community on the formal representation of features, as witnessed in the previous two sections, the use of feature-based approaches and systems is hampered by the lack of a shared and systematic understanding of what counts as a feature, and the diversity of methodologies that this situation led to. Overall, we can say that today features are taken to be macro modelling elements with little machine-processable knowledge attached to them. This is probably explained by the early success obtained by the formal representation of form features and, in contrast, the puzzling heterogeneity of non-morphological information. The lack of a unifying framework for the new types of information makes integrated features much harder to model and manage. Without a solid system for non-morphological features, relevant information for engineering purposes cannot be shared or even modelled, limiting the development of CAD/CAM integrated systems [4].

Additionally, research communities have pointed out from the initial development of the feature-based approaches a contrast between the application nature of the feature-based proposals [3] and the guiding idea that feature models should serve as means to reliably share and integrate information spanning all the production phases, thus independently from application needs [18]. This situation has led to modelling approaches that treat features as aggregations of geometric and non-geometric parameters [6] without addressing the basic issue of what features are supposed to be. As a result, if we assume that geometric elements and features are different things, due to the kind of knowledge they carry, it is unclear how to separate them. Assuming they are similar as one would think working in manufacturing applications, it remains unclear why certain geometric configurations correspond to one feature and others to several [4,5].

From the ontological perspective, these issues point to lack of understanding of the entity one is modelling. This concerns the identity and unity criteria that guide the notion of feature: it is neither clear what a feature is (identity), nor how a feature can be considered as a whole entity (unity).

Let us consider the following case. The block in Fig. 1 can be considered from the conceptual design perspective as a single functional feature, because a functional meaning can be attached to the whole geometry of the piece, which is e.g. functional for assembly purposes. From the detailed design perspective, one can consider the geometric feature formed by A and B and bisected by rib C as a single slot feature, whereas from the machining viewpoint one can consider two different features, namely A and B. This happens because for machining two different operations may be required for the realisation of A and B, while these point to a single morphological element from the design perspective. Additionally, one might want to specify the materials used for the piece, as e.g. A, B and C realised on wood.

Imagine now to have four models of the block: i) the conceptual design, ii) the detailed geometry, the iii) the manufacturing and the iv) material models. Which methodology and criteria should support their integration? We can rely on a formal representation of the geometry of the features, e.g., by following the approach proposed by [4]. The geometry would constitute a basic layer upon which further application- and domain-driven formalisations can be added to enrich the integrated feature-based model. However, the formal representation of geometrical entities does not tell us whether A and B constitute a single feature:

unity criteria of physical entities are quite complex and cannot (and should not) be derived from the choice of a geometrical formalism. Additionally, geometrical formalism by itself is not suitable to manage qualitative knowledge. For instance, it cannot support how to attach functional specifications to feature geometry and topology, or the integration of product morphology to its raw materials.

A mathematical approach to product-related knowledge is well-suited for the development of algorithmic procedures for feature extraction from CAD models, but does not suffice to embed qualitative expert's knowledge into models. From this perspective, what is needed is a qualitative representation of the elements used in a engineering system for product modelling, that is, a formal treatment for engineering concepts, feature above all.

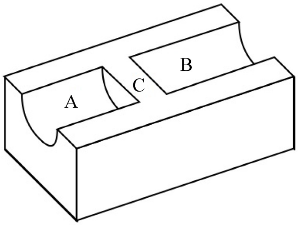

Fig. 1. An example of feature adapted from [5]

Previous work about the application of ontology engineering for feature-based systems has led to the release of multiple ontologies. Nevertheless, research efforts have focused either on application requirements, or on the logical representation of the modelling elements at hand. Little attention has been given to the issue of understanding what features are, how they can be distinguished from pure geometric entities, how they can be enriched with qualitative knowledge and how to characterise feature notions in a way that is stable and re-usable across communities and applications.

5 Classifying Features: An Ontological Viewpoint

From the analysis of the literature two contrasting notions of feature emerge:

F1-feature: Feature as the modelling component of product modelling systems that supplement quantitative geometric models with qualitative engineering knowledge. In this view, a feature is a set of information entities added to a product model for reasoning about the device under design.

F2-feature: Feature as an element of a physical product like a characteristic (e.g. a quality on a par with color and weight), a physical component (a wall of a building), or a geometric configuration (a hole, step, chamfer, etc.). In this view a feature, with its qualifying properties, is related to a physical product by means of specific relationships, depending on the feature types (cf. Sect.6).

These two views have been co-existing and exploited in the literature for at least 20 years. In the former case, a feature exists only within the context of a model: a hole is seen as a helpful, yet abstract, notion that allows a CAD model to convey a variety of useful information about a concavity in the product like why it is needed and how it is obtained. Salomons and colleagues [7] had this view in mind when they stated that a feature is “a carrier of product information that may aid design or communication between design and manufacturing, or between other engineering tasks”. In the same years, Shah and Mantyla [3] pointed to the second view claiming that features are “the generic shapes or characteristics of a product with which engineers can associate certain attributes and knowledge useful for reasoning about that product”. Here a feature is a fully fledged entity of the physical world: a hole is seen as an actual part of the product.

These two perspectives are strictly related: F2-features are the result of manufacturing activities, the very activities that are set with the goal to realise the features in the sense of F1, i.e., the modelling elements. At the same time the F1- and F2-features cannot be confused: a CAD model may specify that a hole feature of the designed part has a diameter of 0,5 cm with a tolerance value of 0,1mm. Yet, each realisation of the CAD model will have a hole which, while compliant with the specification, has its own specific diameter within the tolerance range. Analogously, a feature is present in a physical product [3] only in the sense of F2, as it would make no sense to claim that a computer-based modelling element is a constituent of a material product.

From this perspective, a F1-feature is an “information aggregate” that satisfies some unity condition for an application purpose, in the sense that various information models can be aggregated to count as a single whole element. For instance, the geometry, functional and manufacturing models of the example in Fig.1, while being three different information models, can also be taken to represent a unique modelling feature in, e.g., a CAD/CAM integrated system. A F1-feature is therefore a *whole* element that exists only within a (computer-based) model, and is part of a larger element, typically the product represented in the model.

At a closer look, a F2-feature may be considered not a feature *per se*. Rather, one could claim, it is a feature only *within* an engineering context. Imagine, for example, an engineer performing a quality test to verify whether a hole on a block of wood is within the prescribed tolerance limits. In a weak reading of F2-features, the hole is seen as a feature during the test activity since it has to be checked against some given specification. Yet, the hole as such, i.e., outside this activity, is not a feature. This view suggests that, according to the terminology in [19], F2-features are *anti-rigid* entities, i.e., *being a F2-feature* is a property that an entity has only within some engineering concern or activity: the particular slot A of Fig. 1 may stop to be a feature once the product is complete while remaining the very same slot.

6 Formal Representation

In order to formalise the readings of the F1- and F2-feature notions introduced in Sect. 5, we now adopt the DOLCE foundational ontology as presented in [20] (DOLCE-CORE).

Foundational ontologies are formal theories for the specification of general, upper-level notions, like object, quality, region, which are common to different modelling scenarios. Differently from domain- and application-driven ontologies, which are focused on specific modelling tasks, a foundational ontology has a large scope and can be highly reusable for different purposes. Its notions are based on the philosophical theories of Formal Ontology, which guarantee solid conceptual bases to its categories. Furthermore, since a foundational ontology is mainly aimed at providing a semantic transparent conceptual framework, it requires the use of a rich axiomatisation; therefore, expressive formal languages are preferred over computational and tractable ones. There is nowadays a spread consensus among the scientific community about the impossibility of a unique foundational ontology for all modelling scenarios, since different research communities do not often share the same ontological commitments. It is rather favoured the development of a library of foundational ontologies, including formal mappings among the different modules to facilitate their comparison.

DOLCE has been explicitly designed with a cognitive-bias aimed at capturing the ontological categories underlying natural language and common-sense thinking. It has been employed in various knowledge representation tasks, from social roles and organisations, to business process modelling, engineering design and manufacturing scenarios. Its conceptual framework is limited to *particulars*, entities that, differently from *properties*, exist in time and cannot instantiate themselves. Examples of particulars are Maradona, the Pisa tower and the authors of this paper. Particulars in DOLCE-CORE include *object*, *quality* and *concept*, which will be shortly introduced. The DOLCE-CORE axioms are indicated by **DL**n where n is the axiom number in [20]; we write **DL**n^* for the axioms of DOLCE-CORE which are only informally given in [20].

In DOLCE-CORE an object (O) has primarily a spatial quality (SQ) identifying its location (DL1*); $I(y, x)$ is red as "y inheres in x", and refers to the inherence relationship holding between a quality and its bearer. A quality (Q), among which SQ, existentially depends on its bearer (DL22), namely it cannot exist without it. Intensional properties are introduced in the domain of quantification as (reified) concepts (C) and classification (CF) is used as a sort of (possibly intensional) instance-of relation between a concept and the entities satisfying the properties it describes. $CF(x, y, t)$ holds if y, at the time t in which it is present (PRE), satisfies the property x (DL18). Then, only concepts can classify other entities (D17). Concepts are classified in DOLCE-CORE by a finite number of disjoint spaces, called SP_i, (DL2*), whose structure we do not discuss here.

DL1* $O(x) \rightarrow \exists y\, (SQ(y) \wedge I(y, x))$
DL22 $Q(x) \rightarrow \exists y(I(x, y))$
DL18 $CF(x, y, t) \rightarrow PRE(y, t)$

DL17 $CF(x,y,t) \rightarrow C(x)$
DL2* $C(x) \leftrightarrow \bigvee_{i\in\{1,\dots n\}} SP_i(x)$

For the purposes of this work, we concentrate on the DOLCE-CORE concepts (C) that refer to the "content" of engineering models. In this sense, we distinguish between *what* is described by e.g. a CAD model, i.e. the set of properties that the corresponding physical products have to satisfy (to be considered of a certain type), from the support (a CAD file, or a piece of paper) in which these properties are represented (by means of a graphical or verbal language). We call the latter *representational artefact* (RA): it has the function of *representing* various concepts specified in modelling languages. For instance, by looking at Fig.1 we need to distinguish: (i) its content, i.e., a geometric form with a number of feature; (ii) the content's specification in a graphical language, namely the drawing; (iii) the representational artefact, i.e., the specific page when this article is printed, or the video screen when Fig.1 is digitally visualised.[1] Clearly, one and the same concept can be represented in different representational artefacts.

Formally, we introduce RA specialising the object class O (A1). The relationship of representation RPT holds between a representational artefact in RA and a concept C at a certain time T (A2). A representational artefact implies the co-existence of the represented concept (A3). An instance of RA may represent more than one concept (A4) but in this case there must exist a concept of which all these are parts (A5).[2] Informally, this says that a concept can be complex, e.g., the concept of a car includes the information entities about its components (frame, engine, seats and so on).

A1 $RA(x) \rightarrow O(x)$
A2 $RPT(x,y,t) \rightarrow RA(x) \wedge C(y) \wedge T(t)$
A3 $RA(x) \wedge PRE(x,t) \rightarrow \exists y\ RPT(x,y,t)$
A4 $RPT(x,y,t) \wedge P(z,y) \rightarrow RPT(x,z,t)$
A5 $RA(x) \rightarrow \exists wt(RPT(x,w,t) \wedge \forall zt(RPT(x,z,t) \wedge P(z,w)))$

In the DOLCE-CORE framework concepts have a static nature as they are invariant across time. In design, however, it seems reasonable to allow concepts to evolve. For instance, the concept of a product under design might change over time due to customers' requirements or to the designer's activity. This can be modelled by adding a temporal parameter to CF: $CF(x,t,y,t')$ holds if entity y, as it is at time t', satisfies x, as it is at time t (A6). We thus adopt (A6) as a replacement of axiom (DL18). Additionally, we want to talk about relationships holding among concepts themselves: $CH(x,y,t)$ says that concept x, as it is at time t, is characterised by concept y (A7). By (A8), we have that if concept x classifies entity y and z characterises x, then y is also classified by z. For instance, if a plank concept is characterised by the concept *being rectangular*, the instances of the plank have to be instances of *being rectangular*.

---

[1] In another view, which we do not exploit here, the physical support is the ink on the paper.

[2] We assume that a "reading" of the RA is (explicitly or implicitly) fixed.

A6 $CF(x,t,y,t') \rightarrow C(x) \wedge PRE(x,t) \wedge PRE(y,t')$
A7 $CH(x,y,t) \rightarrow C(x) \wedge C(y)$
A8 $CF(x,t,y,t') \wedge CH(x,z,t) \rightarrow CF(z,t,y,t')$

We can now introduce the class of feature modelling elements, the F1-features, indicated by FC (feature concept), as a specialisation of C. In particular, a feature x implies the existence of a concept y that x characterises (A9). The features as product element, the F2-features, form the class PF (physical feature). Here we concentrate on the strong reading of F2-feature described at the end of Section 5. That is, we assume that a F2-feature is a feature *per se* independently of specific engineering concerns and activities.

FC serves to classify the members of PF. So a feature concept can only classify physical features (A10), while a physical feature can be an object, a quality or a DOLCE-feature (A11). Recall that DOLCE-features are physical entities constantly dependent on other objects, like edges, bumps and holes, see (A12) where we write DP for the dependence relation. Note that we now have three distinct notions of feature at play: F1-features (FC), F2-features (PF) and DOLCE-features (F). The first two are engineering-based notions, the third is ontological.

A9 $FC(x) \rightarrow \exists yt\ CH(y,x,t)$
A10 $CF(x,t,y,t') \wedge FC(x) \rightarrow PF(y)$
A11 $PF(x) \rightarrow O(x) \vee Q(x) \vee F(x)$
A12 $F(x) \wedge PRE(x,t) \rightarrow \exists y(O(y) \wedge PRE(y,t) \wedge DP(x,y))$

Regarding PF features, we need to distinguish three cases. Let x be a physical feature, then: If x is an object, then there is an object y, not a PF, of which x is proper part (A13); if x is a quality, then it inheres in an object x (A14); if x is a DOLCE-feature, then there is an object, which is not a PF, upon which x depends (A15). (Clearly, there are important interrelations among these cases but we do not exploit them here.)

A13 $PF(x) \wedge O(x) \rightarrow \exists y\,(O(y) \wedge PP(x,y) \wedge \neg PF(y))$
A14 $PF(x) \wedge Q(x) \rightarrow \exists y\,(O(y) \wedge I(x,y))$
A15 $PF(x) \wedge F(x) \rightarrow \exists y\,(O(y) \wedge DP(x,y) \wedge \neg PF(y))$

As noted in the analysis of the literature, F1-features (FC) can be associated to domain information depending on the modelling lifecycle phase, or to application scenarios. We provide the formal representation of some application-driven FCs, but the same modelling methodology can be used for others.

Form features (FC_{Fr}) are defined as the elements in FC whose instances have proper parts which satisfy a unity criterion (U), namely they constitute a whole entity (D2). Material features (FC_{Mt}) are the elements in FC characterised by some material concept, called C_{Mt} (D3). Similarly, functional feature (FC_{Ft}) are feature characterised by some functional concept (C_{Ft}) (D4).

D2 $FC_{Fr}(x) \triangleq FC(x) \wedge \forall ytt'\ (CF(x,t,y,t') \rightarrow \exists zv(CF(z,t,v,t') \wedge P(v,y,t') \wedge U(v,t')))$

D3 $FC_{Mt}(x) \triangleq FC(x) \wedge \exists yt(C_{Mt}(y) \wedge CH(x,y,t))$

D4 $FC_{Ft}(x) \triangleq FC(x) \wedge \exists yt(C_{Ft}(y) \wedge CH(x,y,t))$

The formal representation of manufacturing features is about different ontological entities as in this case one has to consider the manufacturing process required for the feature realisation, possibly together with its sub-processes and the required machining tools. Therefore, we need to talk about a manufacturing plan, that is, a manufacturing concept (C_{Mf}) classifying a manufacturing process (E). In this case, the classification holds between C_{Mf} and E relatively to the time of E itself. Also, we have that a physical feature (typically present at the end of E) depends on an object (O) which participates "passively" in the process (PC_p). Informally, this amounts to say that O is the workpiece, i.e., it undergoes the manufacturing process. Finally, other objects participate "actively" in E, e.g., the manufacturing resources employed during the process. Given these qualifications on the complexity of predicate C_{Mf}, (D5) gives the general definition for manufacturing features.

D5 $FC_{Mf}(x) \triangleq FC(x) \wedge \exists yt(C_{Mf}(y) \wedge CH(x,y,t))$

7 Ontology-Based Feature Modelling: An Example

The ontology-based modelling approach introduced in the previous section is now applied to the formal representation of the features in Fig.1. As noted in Sect. 4, current approaches presented in the literature do not provide sufficent support for the integration of multiple qualitative knowledge aspects.

We formalise four different perspectives on the product features, namely the form, the functional, the material and the manufacturing perspectives. Let f be the F1-feature of the product concept *cob* in Fig. 1, then f classifies the F2-feature pf of any instance of *cob* and pf has three parts: the F2-slot feature on the left (pf_1), the F2-slot feature on the right (pf_2), both classified by the same F1-slot feature (f_s), and the F2-rib feature (pf_3) classified by the F1-rib feature (f_r). See formula ($f1$). We thus have pf as the complex F2-feature relative to the geometric information of A, B and C in Fig.1.

Since f is also a F1-functional feature, it is characterised by the functionality concept cft (f2). Similarly, f is characterised by material concept *cmt* (f3) while the manufacturing perspective is given in (f4). Since f is characterised by cft, *cmt* and cmf, we obtain that its corresponding pf satisfies the functionality, the material and the manufacturing concepts (T1).

f1 $FC_{Fr}(f) \wedge CH(cob,f,t) \wedge CF(f,t,pf,t') \wedge CF(f_s,t,pf_1,t') \wedge CF(f_s,t,pf_2,t') \wedge CF(f_r,t,pf_3,t') \wedge pf = pf_1 + pf_2 + pf_3$

f2 $FC_{Ft}(f) \wedge C_{Ft}(cft) \wedge CH(f,cft,t)$

f3 $FC_{Mt}(f) \wedge C_{Mt}(cmt) \wedge CH(f,cmt,t)$

f4 $FC_{Mf}(f) \wedge C_{Mf}(cmf) \wedge CH(f,cmf,t)$

T1 From f1, f2, f3, f4 and A8:

$$CF(f,t,pf,t') \wedge CH(f,cft,t) \wedge CH(f,cmt,t) \wedge CH(f,cmf,t) \rightarrow CF(cft,t,pf,t') \wedge CF(cmt,t,pf,t') \wedge CF(cmf,t,pf,t')$$

We have just showed the general modelling approach by which qualitative knowledge relevant to Fig. 1 can be specified by means of our theory. A more detailed formalisation requires to specialise further the relationships across the types of features and the ontological entities. For example, the overall functionality of pf may be subdivided across its physical feature parts and, similarly, the internal structure of the event relative to the manufacturing feature can be used to clarify how the F2-feature is realised.

8 Conclusion

The development of knowledge-based system for product-lifecycle management is a challenging task, as it requires the formal representation of detailed engineering knowledge, as well as the integration of various qualitative knowledge aspects. As we stressed in the paper, no stable, nor well-founded approach is currently available for this purpose.

We presented an ontological analysis of feature-based product modelling notions that is aimed at supporting both product knowledge specification and qualitative knowledge integration. We concentrated on the classification of feature notions by distinguishing between modelling elements and real-world entities, and by investigating their dependencies upon other ontological and engineering notions. In one case, features are meant to embed qualitative knowledge into product models, while in another they are actual entities on a par with the associated physical products. From this distinction, we argued that feature types should be distinguished at the modelling element level. In particular, we discussed engineering features as objects, as qualities and as DOLCE-features although it is still unclear whether these categories are exhaustive. In the end, we showed an approach to formalise and integrate various features qualitative models following our analysis and provided an example related to design and manufacturing.

Acknowledgments. This work was partially funded by the VISCOSO project financed by the Autonomous Province of Trento through the "Team 2011" funding programme, and the FourByThree project funded by the European Horizon 2020 program (grant agreement 637095).

References

1. Kim, K.-Y., Manley, D.G., Yang, H.: Ontology-based assembly design and information sharing for collaborative product development. Computer-Aided Design **38**, 1233–1250 (2006)
2. Fenves, S., Foufou, S., Bock, C., Sriram, R.D.: CPM: A core model for product data. Journal of Computing and Information Science in Engineering **8** (2008)

3. Shah, J., Mantyla, M.: Parametric and Feature Based CAD/CAM. Concepts, Techniques, Applications. John Wiley and Sons (1995)
4. Zhou, X., Qiu, Y., Hua, G., Wang, H., Ruan, X.: A feasible approach to the integration of CAD and CAPP. Computer-Aided Design **39**, 324–338 (2007)
5. Mantyla, M., Nau, D., Shah, J.: Challenges in feature-based manufacturing research. Communications of the ACM **39**(2), 77–85 (1996)
6. Ma, Y.S., Chen, G., Thimm, G.: Paradigm shift: unified and associative feature-based concurrent and collaborative engineering. Journal of Intelligent Manufacuring **19**, 625–641 (2008)
7. Salomons, O.W., Houten, F., Kals, H.J.J.: Review of research in feature-based design. Journal of Manufacturing Systems **12**(2), 113–132 (1993)
8. Amaitik, S.M., Kilic, S.E.: STEP-based feature modeller for computer-aided process planning. International Journal of Production Research **43**(15), 3087–3101 (2005)
9. Han, J.H., Pratt, M., Regli, W.C.: Manufacturing feature recognition from solid models: A status report. IEEE Transactions on Robotics and Automation **16**(6), 782–796 (2000)
10. Wang, Q., Yu, X.: Ontology based automatic feature recognition framework. Computers in Industry **65**, 1041–1052 (2014)
11. Brown, D.C.: Functional, behavioral and structural features. In: ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers (2003)
12. Groover, M.P.: Automation, Production Systems, and Computer-integrated Manufacturing. Prentice Hall Press (2007)
13. Nepal, M.P., Staub-French, S., Pottinger, R., Zhang, J.: Ontology-based feature modeling for construction information extraction from a building information model. Journal of Computing in Civil Engineering **27**(5), 555–569 (2013)
14. Anjum, N.A., Harding, J.A., Young, R.I.M., Case, K.: Manufacturability verification through feature-based ontological product models. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 1086–1098 (2012)
15. ISO, Industrial Automation Systems and Integration - Product Data Representation and Exchange. Part 1: Overview and fundamental principles, iso 10303–1:1994(e) ed. (1994)
16. Amaitik, S.M., Kilic, S.E.: An intelligent process planning system for prismatic parts using STEP features. International Journal of Advanced Manufacturing Technology **31**, 978–993 (2007)
17. Dartigues, C., Ghodous, P., Grüninger, M., Pallez, D., Sriram, R.: CAD/CAPP integration using feature ontology. Concurrent Engineering **12**(2), 237–249 (2007)
18. Abdul-Ghafour, S., Ghodous, P., Shariat, B., Perna, E., Khosrowshahi, F.: Semantic interoperability of knowledge in feature-based CAD models. Computer-Aided Design **56**, 45–57 (2014)
19. Guarino, N., Welty, C.: An overview of ontoclean. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 201–220. Springer-Verlag, Berlin, Heidelberg (2009)
20. Borgo, S., Masolo, C.: Foundational choices in DOLCE. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. Springer Verlag, Berlin, Heidelberg (2009)