Using Secondary Information Sources to Generate and Augment Semantics of Design Information

Sascha Opletal   Dieter Roller   Steffen Rüger

Institute of Computer-Aided Product Development Systems, Universität Stuttgart
Stuttgart, Germany
email: {opletal, roller, rueger}@informatik.uni-stuttgart.de

Abstract

The reuse of design knowledge for use in CAD systems is a promising way to reduce time and cost during the design cycle. To support this, a semantic core for a novel type of informational infrastructure with the focus on supporting CAD systems is introduced, that allows to extract arbitrary subparts of the information base and use it efficiently in related projects.

The key problem addressed in this work is the automated setup and classification of information pieces within several involved knowledge domains. This is solved by connecting depending design methods strongly to information sources outside the actual CAD design environment with a focus on knowledge generation, distribution and application. As a result the approach will support problem solving within the geometric area through a system that can classify information based on context.

Keywords: CAD, Knowledge Bases, Semantic Relation

1. Introduction

Knowledge and information extraction, the aggregation and propagation of relevant pieces as well as relevant structures are needed in many application areas. As the overall goal is the efficient reuse of design knowledge, it is required to be able to extract arbitrary subparts of designs and to apply them to other design situations. Today this is only possible with form features, configuration models and other predefined methods, where all the knowledge and the procedures have already been thought of and have been integrated into the product models by a human designer.

The presented approach uses selected methods of different disciplines of information analysis to effectively support the setup and usage of the semantics involved with CAD models. These CAD model semantics have come a long way in the last years, but are still very low-level compared to the requirements of a system that allows easy reasoning beyond integrity checks. Though there are some abstractions that enhance semantics, like constraints and form features, the meaning of a certain part of a design remains in the dark, apart for special use cases, where all knowledge can be formulated [16]. If there is a special need to evaluate a certain aspect of a design, evaluations have to determine the meaning of a design through feature recognition or analysis of design graphs [15].

The involved techniques for a more generic approach to the problem are stemming from the field of computational linguistics, from where disciplines like information retrieval (IR) and information extraction (IE) have evolved. IR has the goal of finding information and evaluates the correctness or relevance of found documents or information sources. This is done by computing precision, recall and fallout (Eq. 1-3)[3], where R is the set of relevant documents, I the set of irrelevant documents, P the set of found documents, and N the set of not found documents.

$$Recall = \frac{|R \cap P|}{|R|} \quad (1)$$

$$Fallout = \frac{|I \cap P|}{|I|} \quad (2)$$

$$Precision = \frac{|R \cap P|}{|P|} \quad (3)$$

Recall then defines the found amount of documents, precision defines the amount of found correct documents and fallout the amount of not found relevant documents. A problem with this approach is that all correct documents have to be known in advance to optimize the knowledge search and to evaluate it using recall and precision. Once a set of useful documents is found, IE can be used to focus on the gathering of meaning out of the information sources.

[5] defines this as ”An IE system takes as input a text and ’summarizes’ the text with respect to a prespecified topic or domain of interest”. This works generally on unstructured text within a fixed domain that defines how to handle and combine the found information. This process is typically even more

constrained and only works on given scenarios, which can be described as the focus within the work domain. As an example, the domain could be 'Economic news' and a possible scenario for it could be 'Changes in Management Positions'.

A well defined scenario [28] allows that techniques like stop-word elimination, filtering of known subjects and reasoning can be used to strip unimportant expletives that are irrelevant for the task. A more general approach, but also focused on domains, is the field of Knowledge Discovery in Databases (KDD), or data mining. It has the goal to recognize unknown patterns and relations that are not specifically encoded or modeled by using statistical analysis of data sets. An example is to find changes in the consuming habits of customers to evaluate products or to rearrange a business strategy. Data mining is subdivided into different disciplines, motivated by the processing methods through which the data is passed. The focus on **data description** is to give a compact representation that is reduced to the essential information. The **differential analysis** tries to identify data sets that are deviating from given norms or standards. The **dependency analysis** tries to make out relationships between the attributes of information objects. **Clustering** is used to segment the data set into groups of interest.

While being focused on special domains or special purposes and scenarios, a multi-domain approach has the the advantage of being more practical in heterogeneous work environments as found in companies.

2. Knowledge domain and Aggregation

In the product design process, CAD systems cover the geometrical design aspects. PDM (Product Data Management)/PLM (Product Life cycle Management) systems store documentation. CRM (Customer Relationship Management), SCM (Supply Chain Management) and ERP (Enterprise Resource Planning) systems are used for accounting, materials administration, NC programs, or information for advertising. The overall knowledge process depends not only on explicitly structured and stored information pieces but also on unstructured sources like email that is exchanged between coworkers or even information that is not explicitly modeled and stored in the system. The aspect of explicit and tacit knowledge strongly influences the work processes [14][20].

In order to input data into a knowledge base and to be able to reason on it, the construction framework has to i) consist of rules how to build it and ii) has to have as a base a specified domain. There are lots of frameworks for i), like ontologies, semantic networks and so on, but ii) is still a major problem when it is set up from a multitude of domains as described above.

The goal is to avoid the case where somebody needs to write down which concepts belong to the domain of the application and which not - or even model the concepts first. One reason for this is, that everybody has different concepts in mind and also if two people think about the same piece of equipment, they could use many different names for it. [17] describes this by introducing the meaning triangle to relate symbolic descriptors to objects. Checking for correct descriptive symbolics is not the focus of this work, but to correlate given symbolics correctly to each other. This also covers situations where multiple symbols for an object exist, like different names or a filing with different numbers.

Generally a knowledge modeling framework is a specification of a conceptualization. This is a formal description of artifacts and their relations, that are used to build a common base for the concept formation of a person or a group of persons. The goal is to describe the concepts of human thought and communication in an unambiguous way by using a formal fixation of concept hierarchies, relations and involved attributes. This allows the use of computational tools for inference, extraction of information and the generation of searchable indexes. The main approaches to represent knowledge are either symbolic or connectionistic. Symbolic knowledge is formulated in schemata or rules of the different expressions of a model. A connectionistic system stores the knowledge by training and reveals the knowledge through interaction, which means that there is no way to access knowledge directly since the storage process is not transparent.

A knowledge representation can be generally classified by correctness, power of expression, efficiency and complexity. The content of one representation can usually be transformed to another representation, but there is a risk of semantic loss, depending on the power of expression of the target representation or equally a source representation that does not deliver the required input. The chosen representation should be connectable to all involved information resources and will act as an instance that mediates and aggregates the information.

Three frequently used formalisms to model knowledge are semantic networks, frames and ontologies. Generally they consist of concepts and relationships between concepts. A concept can be an entity of many types ranging from structured frames to unstructured data files. The relations (comparable to ontologies) usually have a base semantic similar to the UML, but are often extensible to arbitrary semantics. Expression is high and versatile, but structures have to be built manually according to a meta-level. A designer is required to define the base structure and uses elementary methods to set up concept structures and relations. Base semantics of relations are usually:

- IS-A, A-KIND-OF (a kind of) used to represent heritage or information of generalization
- PART-OF, HAS-A-PART used to represent aggregated information
- MEMBER-OF, INSTANCE-OF used to represent instantiation and individualization

The product development is usually carried out within a knowledge frame that spans between persons and teams. There exist relations of various semantics between all involved (also abstract) units like teams, db-objects, prototypes, projects, and so on. Though some relations are not explicitly encoded, they are supplemented by a persons mind while working with the given data set, which means that the collection of data together with implicit knowledge forms a semantically connected knowledge system.

Applied in product development, the vertices of such a system represent data objects and the edges represent the semantic relations between the objects. Once the structures and relations are set up, they are fixed. The user of the system can only generate instances by using the structures of the network to fill in data. Most knowledge representations are tailored to specific use cases, but can also be enhanced or modified to be used in other contexts. However, structure and expression are often too fixed, while rules are tailored for specific reasoning.

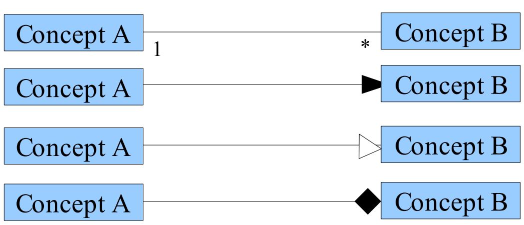

Figure 1. UML relations between classes

Similar semantics are used also by the wide known UML notation (see fig. 1). From top to bottom is shown the graphical annotation of an association with multiplicity, the directed association, the generalization, the aggregation or composition.

Derived from such semantics, the knowledge propagation to people and places where it is needed can be decided. This will save huge amounts of time, but needs a powerful annotation scheme. Changes in a CAD model can for example trigger actions in depending data like cost calculation, manufacturing processes and so on. Such a system needs an active component to be able to connect knowledge, push knowledge or coordinate knowledge generation.

A combination of different knowledge acquisition methods and representations can be used to form a system that organizes information without the need to form structures with fixed concept frames for the deposition of knowledge. The information should give rise to structures and not vice versa.

3. Related Work

Recent approaches to model knowledge often use static setups of rule bases and concepts to capture the knowledge domain. Currently used [25] informational infrastructure for the development and manufacturing of products distributes different information types over specialized systems.

A system with an intelligent tutoring agent is described in [26], where the preprogrammed domain together with the allowed actions form the base for reasoning. [10] suggests a domain independent knowledge manager that is very flexible in terms of application, but omits how the domain knowledge can be gathered and be distinguished from non-domain knowledge. In [22], the domain knowledge is acquired automatically by evolutionary learning but on a very limited domain. Traditionally the search for domain knowledge stems from NLP (Natural Language Processing), that searches for meaning in unstructured free text, based on large text corpora of a specified domain. A method to use meta-information to classify documents based on the citation information of authors is presented in [9].

A domain independent approach is described in [21], but the topics and domains have been preselected, which is undesirable for the problem addressed in this paper, where more than one domain is involved. [12] uses a highly sophisticated annotation to classify the physical effects of single mechanical modules. Using them as building blocks, mechanical functions can be created by connecting the available and desired input and output forces. [9] adresses the reasoning aspects behind constructability.

A more interactive approach that uses evolutionary algorithms to design products is presented by [6]. The validation function from each generation to the next is the human who tells if he likes a generated design or not, which resembles an information retrieval.

Many more approaches exist to solve domain specific problems. The problem discussed in this work is in contrast to these approaches, as the focus is to establish a framework that is able to bootstrap itself and store knowledge domain independently.

4. Capturing Design Knowledge

The aspects involved in developing an object or structure to meet set criteria are considered as design knowledge. This includes the lead to a solution, the process how criteria of the application area are met, and also the solution itself. Many aspects of the geometry fall into this definition; some can be analyzed automatically, others can only be captured by hand. Generally an algorithm can log the use of utilities and the steps of creation for a specific object, but the intention of the designer and what the object should be good for or where it will be used in and what else would be appropriate in this specific situation is beyond the perception of the computer. There are design goals and supporting information that is unique to special domains, but also special domains are involved within a project, that don't cover mechanical assemblies and require special knowledge capturing methods as they also contribute to a solution.

The acquisition and storage of the different types of domain knowledge can be characterized into domain independent and domain dependent knowledge. Domain independent knowledge should be equal in use and acquisition across design domains. This can be captured at a higher abstraction level [19] above the feature based design and the product configuration. Form features represent geometric macros that capture and preserve the design intent as geometric setups and machining instructions. They can be considered domain independent, since they are used to model geometry in many projects. Certain groups of semantics which are annotated or classified through usage are similar across the domains and can therefore be directly transferred. The basic geometric construction methodology is domain independent and is used in every design system. Next to the geometric information there is process information within the CIM/CIE environment that has influence on the design process. Those organizational aspects can be modeled in the same way across domain boundaries.

Domain specific knowledge in contrast is captured by information and methods that are unique to a design domain and cannot be transferred across domains. As example consider the design of a molding form that is used to form components out of mold. In order to get the ready component out of its form, the form needs to be fitted with ejection channels, where the component can be pushed out. The placement of those ejection channels needs to be formalized and calculated by a specialized tool for each new molding form. Such knowledge is domain dependent, since it only applies to the domain of molding forms.

It is however most certain that this knowledge can be reused within different projects that are developed within this design domain. Those domains need customized treatment for capturing, transfer and application of reusable knowledge. The general approach to handle these problems is very similar. The applied construction methods can be generalized and be fed with specific knowledge for the actual domain. These specialized methods can be seamlessly integrated into the information process through a standardized interface.

The feature recognition serves as a first measure to compare the design situation at hand with the stored information in the knowledge base. Features are used in conjunction with restrictions to capture the design intent in a product model, so that changes do not destroy the original intent of the designer. Features of a design can be captured in two ways: During the actual design process, the designer assigns features to his geometric structure knowingly by using a certain feature function and unknowingly by creating a structure that holds certain features that were not explicitly constructed. The features that are assigned through feature functions can be directly annotated in the database and be refined by questioning the user to input his intent. Implicit features and unannotated features can be found during a preparation phase of the knowledge base, where new information is being gathered out of the stored objects. This includes a feature recognition process to find additional knowledge about objects through the presence of features.

The feature recognition is also very important in the information retrieval process where a design problem is analyzed into a new feature structure and this feature structure is then searched in the knowledge base as an exemplified query. Additional information is supplied through the characterization of the design situations in the overall information process. For every applicable situation found through the recognition of features, there has to be a measure to decide how much of the found information is applicable to the given situation. This is done by a similarity function that takes into account the geometric similarity and the process similarity. The geometric similarity is computed through the involved objects in a design situation with respect to the needed solution. The similarity of objects can be determined through design history, annotation and features. The similarity of the stored design situations is compared through the similarity of the involved objects to the objects in the current situation. Further comparison is done on the information process level, such as context in terms of design teams, as well as requirements and input from different organizational units of the company.

5. Modeling Information

The conventional approach to knowledge based design means that the user fills the knowledge base with rules and by doing so decides how the product model will behave or can be reused [2]. Knowledge in this context can be described as [14]: "Knowledge is information that is relevant, actionable, and at least partially based on experience."

A special focus has to be on the starting phases of the product development process as these phase are characterized through a high need of interdisciplinary information. A knowledge base that is used in such an environment needs the capability to cover objects and relations of many different kinds. These requirements emerge from the dynamics of the product development process and as a result, the knowledge base does not represent a stable and consistent state, but needs to reflect several development processes that are running in parallel. The most important tasks are to find and use methods to synchronize the work processes and to be able to recover from inconsistencies. The methods of representation for the product data have to support incremental refinement and extensions to the knowledge which is gathered during the product development process.

The process of knowledge acquisition is usually guided through a descriptive framework that serves as a meta-level to set up the rules and methods to fill in data (see fig. 2).

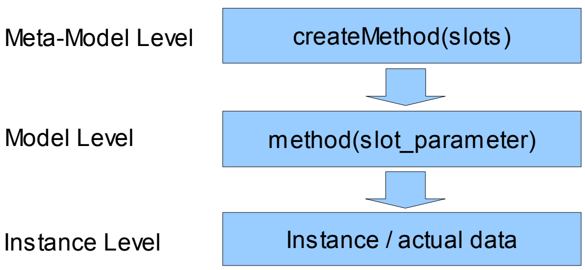

Figure 2. Levels to build a data model

A good knowledge representation and the associated tools should support the need to acquire, consolidate, and distribute information among involved clients to create a measurable advantage. The collaborative environment should support this through a collection of processes and tools, which will lead to new ideas, processes and techniques. Several knowledge modeling techniques were developed over the years:

- Frames
- Ontologies
- Semantic networks

The individual models each focus on different aspects of knowledge. Ontologies as an example are suited to provide and organize additional CAD process information in an effective way, since the development of a product is done within a project organization and driven by persons and teams, which are themselves considered as acting objects in the information structure. They create several different work objects, such as requirements, descriptions, CAD-models, NC-programs, or material for advertising. Between these logical units exist relations of different kinds.

Those ingredients define a semantic network. An extension, called the *active* semantic network (ASN) [24][7], has been developed at the Institute of Computer-aided Product Development Systems at the Universität Stuttgart to support active propagation of knowledge. It provides means to store all information that is created during a cooperative development process. The main component is a mechanism that executes automatic actions, driven by the states and situations during the use of the semantic network, which is of great importance to coordinate the development in an effective way. The structure of the active semantic network consists of objects that are connected in a form that resembles a net. In this net, the vertices represent objects of the product development knowledge and the edges represent the context and relations between those objects. All relevant information for the product development can be stored in this active semantic network, such as input from quality control, marketing, buying department, service and recycling. The knowledge model of the active semantic network is dynamic and can support the growth of knowledge and new knowledge types during the product development process.

The active part of the semantic network allows for the propagation of changes through the whole net and the triggering of actions based on those changes. It bears the following functionality:

- Inference mechanism
- Message passing
- Communication handling
- Execution of tasks

Inferences are used on the knowledge base to execute rule-based recombinations of information and automatic calculation of formulas. Through the passing of messages to users or the establishment of communication channels, the system can react on problems that it or the users cannot solve by themselves or where human attention is needed. A novel aspect when compared to other knowledge representations is that the active component fills the role of an assistant to the designer. It can execute routine tasks and bring the attention to problems that have occurred during the development process.

The active semantic network can support the designer with knowledge from an expert that would be out of reach or hard to come by in a standard work environment. The designer can be notified if a design decision has consequences on the subsequent product development phases, such as the manufacturing process and product cost. By providing the knowledge of areas where the designer is no expert in, the active semantic network takes the place of a colleague with the necessary expertise.

The problem to define the domain of knowledge as a basis for the subsequent usage in reasoning processes is not easy. It is often done by hand and thus tailored explicitly to a specific application area. However domains and contexts in all industries and application areas differ a lot, even between companies within the same field of activity. Thus, the automated set-up and maintenance with respect to new requirements is an enormous task. A knowledge base that builds itself and also has the ability to self-organize is the key for a substantial push in the re-use of a company's stored information where it is desired to keep the integrity of the structures of databases. This motivates the approach of a layer on top of all involved information systems, that organizes and groups information pieces according to their relation and relevance for projects.

6. Use Case: The Information Need for Sheet Metal Design

The application and data flow of the semantic core is described from the perspective of a CAD designer that interacts with the knowledge system. Many factors drive the informational need with cost being the most significant, as the work area of development and design contributes 75 percent of the total costs.

The capability and experience of a human designer generally makes or breaks a project in terms of initial quality and the time to reach a satisfactory base construction. As the initial design is taken and refined in several iterations [23], using input from other involved teams (quality control, mechanical testing, production units, …), it is desirable if the system could automatically provide the information from established solutions and paid attention that the product design meets the requirements (norms, cost, quality …).

The design process of a new product requires precise knowledge on function and manufacturing of the final product, as the function of the product is determined by its mechanical construction. The quality of the construction depends on how it was manufactured and which materials were used. Also most new products are not designed from scratch, but built on or advance an established base construction of previous designed and proven products.

The designer has to take all these requirements into account when starting the design process. The goal is to integrate human design knowledge into the CAD system, so that certain design decisions can be made automatically or be suggested in the relevant context to the human designer. A "semantic CAD system" can be formed, based on a semantic core, that has the primary task to support the design process by operating as an assistant to speed up the process and guarantee high quality standards by supporting the design process as detailed as possible. These supporting steps consist of:

1. Idea of the design / 2. Articulation by entering it into a CAD system / 3. Proposal of the system / 4. Transfer or modification of the proposed design

A precisely articulated idea, aided by the support of adequate design methods will make it possible to give the designer suitable help that is tailored to his needs. To see the information flow during the CAD process, standard problems will now be discussed to motivate the presented approach and to show the benefits it provides. The proposal of the system is derived from the consideration of the knowledge base, which will be checked according to three different controlling areas: design controlling, component controlling and realization controlling, with each of the controlling areas being presented below. The necessary data flow for the monitoring is made up of several levels:

1st level: design work
At this level all information about the user interactions, the used materials and the geometry of the design is collected which represents the raw data for the knowledge processor.

2nd level: knowledge processor
The knowledge processor receives the raw data and generates knowledge by consulting relevant sources of information that are from the same context as the current workpiece or work context. After aggregating relevant information, the knowledge processor synthesizes a helping information piece and presents it to the designer or applies a modification directly to the work-piece.

3rd level: decision of the designer
The designer has several options how to continue with the provided information. Proposed design methods of the knowledge processor can be applied and change the design. Alternatively, the information on the current design only appears passively, is noticed and the design work continues. Feedback is gathered by other means, for example when requests are made for more information on the same topic. The decision will be monitored and used to evaluate and refine the decision score.

The global problem is to recognize the intentions of design work in the correct context within the information process and stored information. The knowledge processor is responsible to gather all the information needed to support the design functions. Therefore all relevant data has to be extracted from a modeling process and also from the other information systems involved. This is guarded by three different surveillance types:

1st section: design controlling

- Verification of design rules for the design, manufacturing, installation and cost reduction: rules are extracted from active and older, already completed projects. With these rules a learning process is able to use relevant and correct older structures for new designs with the goal to minimize design errors
- Connectivity of the knowledge processor to a quality system: in case of defects or frequent customer complaints their cause has to be analysed. If the design is responsible, it is recorded for future projects to avoid mistakes
- Compliance with the requirements regarding existing tools and machinery: avoid expensive

acquisitions of special tools if it is possible to manufacture with existing work equipment

2nd section: component controlling

- Comparison of similar designs: a work history is stored and accessed during the phase of modeling. In the case of similarities between older and current designs, proposals are made by the system. If the designer agrees with the proposals they are automatically included in the current design
- Integration of products from third party manufacturers and completed parts: the use of standard parts has a high potential for cost reduction. Therefore parts will be identified during the design as third party parts that can be bought cheaply and thus avoid the redesign of parts or unproven designs

3rd section: realization controlling

- Recognition of failure in design: not feasible designs should be detected as early as possible so that the designer does not waste time in producing something incorrect
- Consideration of the material properties: if a bending needs to be made in the sheet metal module and the sheet metal is too thin, it has to be avoided by giving a corrective proposal
- Recognition of the possible unfolding of sheet metal design: invalid designs have to be avoided

As example the following design situation is considered where the designer needs to be advised why it has a bad shape. An optimized shape is provided by the knowledge processor with support of the intelligent CAD methods based on the controlling sections.

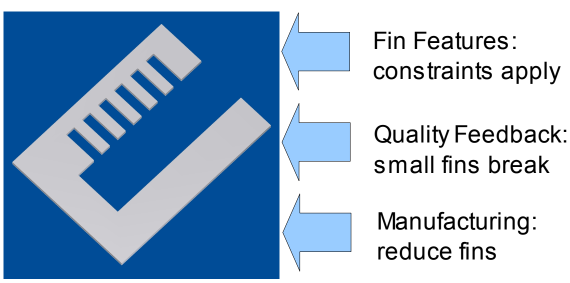

Figure 3. Sheet metal with thin fin elements

The model is checked for various criteria. On one hand it is checked what kind of material is used and what material properties it has. On the other hand the shape and form of the model is analyzed. In this example the material used is a metal plate (see fig. 3). The fins were created with a feature that was used six times. A feature consists of one or more modeling elements. The analysis of the design recognizes that there are multiple (five) fins standing out, which are relatively thin developed. The problem in this design is that too fine structures are not in accordance with the design rules derived from manufacturing machinery constraints and quality feedback. In the production a fine stamp would be required, which is very sensitive to destruction. It is also very likely that too subtle fins will break off in the process of manufacturing.

The design has to be examined whether the fins are not too close to the edge of the metal sheet or if it might be too thin to keep the fins in a solid position. To modify this model into a better shape the slim feature is modified in the way that two of the slim features side by side create a new thick feature. Then the system removes all slim features from the model (see fig. 4) and by doing this reuses derived and proven information.

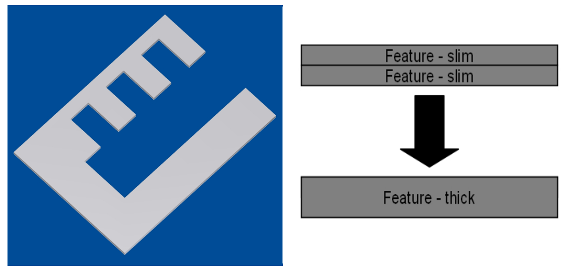

Figure 4. Optimized sheet metal fins

The thick feature will be applied three times to the model. With the thick feature two massive fins are generated, which are much less sensitive to damage in the manufacturing process. A robust stamp can now be used with more resistance against break off. The spacing of the fins to the sheet metal edge was classified as uncritical and was retained.

The required knowledge has to be stored and retrieved from the knowledge base to be applied to a relevant design situation. This means that procedural and other knowledge has to be classified and related to the current and possible future working contexts. The knowledge to address the problems described above could be formulated as:

```
begin compare_features;
  if number of features in model browser is >= 2 then
    count all features of equal form;
  else no modification possible;
  end if;
end compare_features;
begin merge_features;
  if count of equal features is odd number then
    merge = true;
    create new features;
  /* 2 slim features merge to 1 thick feature */
  else merge = false;
  end if;
end merge_features;
begin start_modification;
  if merge = true then
    save positions of slim features;
    calculate new positions for thick features;
    delete slim features from model browser;
    insert thick features at new positions;
    msgbox with information about modifications;
   end if;
end start_modification;
```

These and similar code snippets can then be classified using the keywords „features“, „constraints“, „modification“, „manufacturing“, „sheet metal“, and „quality control”. The use of relevant knowledge has to be decided from the working context, that is partly derived from the user interactions and partly of the design situation and previous knowledge that has be used and applied. The situation could be as described in fig. 5, where the relations between the areas are given. The vertices then represent the actual knowledge to be applied.

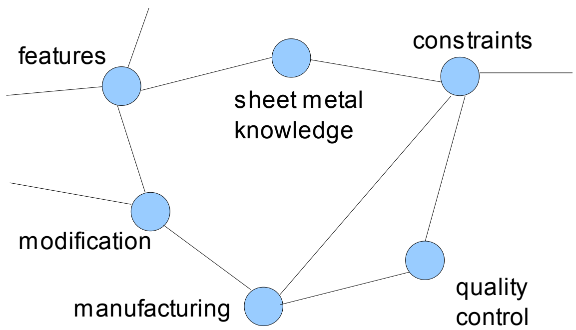

Figure 5. Knowledge for sheet metal design

If the user works with features and executes a verification process for manufacturing constraints or quality control, the system guides him to the sheet metal knowledge and applies related information accordingly. A score of quality through evaluation of several sources linked to relevant topics then forms the base of the semantic core. A great difficulty is to store the geometric information in a readily available form for the adaptation to design tasks. It is impossible to extract design knowledge out of a pure geometric design. The machine would not know what the functions and intent of the single product parts are.

To know which stored products could be a knowledge source for a design process, the products have to be semantically annotated. In a modern CAD system this is already done in a very limited way. If a feature-based system is used, then the product model stores the type of features that the engineer uses during the design work. This can be driven further, in a way that the designer gets a semantic toolbar and annotates product parts and groups by hand while he constructs them. From this annotation the system can learn and through comparison of geometric elements it can annotate other elements in the database using a probability function.

The annotation layer is needed to be sure, that the topic of the analyzed design situation fits to the current design problem. If the analysis was only based on geometric properties, then there would be many unwanted solutions. Almost any geometric shape can be transformed to fit in a given situation, but it has to be decided which geometry is wanted and which is not useful in a given situation.

7. Building a Semantic Core

Various information sources need to be integrated into the knowledge process as discussed in the design example. Although the information flow is directed to the requirements of the designer, all other involved knowledge processes can profit in a similar way. The approach superimposes the existing information sources with a guiding structure to provide a new level of insight by providing more contextual information than it is possible with the isolated systems. The need for reasoning and knowledge aggregation is therefore greatly emphasized.

One key feature as described in the above sections is to avoid the need to build up structures or to identify, limit or input the domain by hand. Therefore the first step is a classification of the involved information systems that are used as resources provided by the company. This will be the foundation where the information is stored, identified and connected to relevant information from other systems.

Some systems may have a dominating role regarding the intended purpose, e.g. an information in a PDM system would be regarded as a primary source, but information extracted from email would be treated as a (maybe unproven) secondary source that can enhance the value of the primary source. The following step is to set up an initial key set as a seed from which to build the base core semantics. By extracting significant descriptors like document or project names and relating those keys, the information resources are analyzed for collocations of pairs of those keys. To refine the confidence of a found or supposed relation, as shown in fig 6., the internal base is switched with external resources like Wikipedia [27], Google [11], or ontologies like OpenCyc [18]. Those resources cover many domains and are used to identify relationships between the key identifiers. The results help to compute if there is a strong or a weak relationship and if there are more relevant key identifiers that were not found during the initial seed.

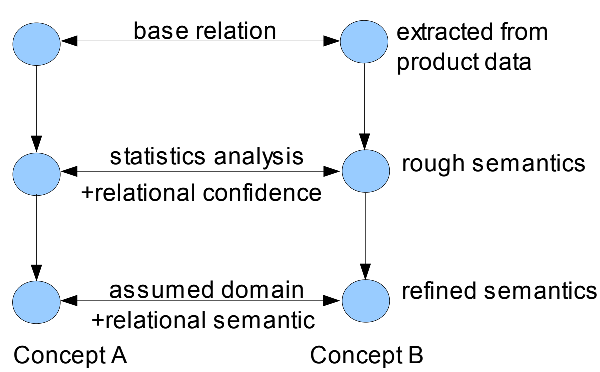

Figure 6. Finding and refining semantics

Through the extraction of internal and after that again external relationships of concepts by methods of information retrieval, the semantic confidence is rising. The semantic relations of interest from the aspect of reasoning in CAD are relations of the type:

"part of"/ "works on" / "similar to" / "instance of" / "new version of" / "has relation to"

This seed acts as a first rough information structure which is not static but refined in subsequent steps, with the work flow and relations of the applied methods shown in fig. 6.

The semantic core drives all informational processes regarding retrieval and push of information. A reference is built by classification of all database content for fast access. Besides the overall classification of knowledge resource, there is also a need to establish an internal resource semantic. The goal is to be able to extract relevant knowledge for active knowledge delivery based on context. To keep the semantic core up to date and valid, there is a need for constant training on the proposed model, where a significance evaluation of the pushed content is done by observation how the knowledge is used in criteria like access, context and expanding of usage paths along the network.

A data set that would lead to the situation described in step 6 of fig. 7 would look as shown in table 1 with knowledge sources α,β and γ, where the influence strength is $\alpha>\beta>\gamma$, given by the type of the resource.

Table 1. Relational Information

	A	B	C	D	E
A		αβγ			α
B	αβγ			αγ	αβ
C					α
D		αβγ			
E	αγ	αβ	α		

Table 1 shows the results delivered by the information resources as relations between the concepts A-E. Raising confidence, apart from the influence strength of resources and the amount of congruent found relations, can be done by monitoring the usage of the semantic core. This is used as an approach to avoid asking the users for feedback. After the initial setup, the relations can be further refined through usage observation based on criteria like access, context and time of usage or informational movements along the network. This is a passive process, since it is difficult to detect if the piece of knowledge is directly copied or used and modified. A heuristic that enables the system to cope with contradictions and user disagreement is introduced at the end of this section.

Problem solving capabilities can be integrated at this point through inference mechanisms and also external solutions can be requested automatically by sending messages to experts. If someone introduces a new concept or term into the databases by creating new resources or documents with a key that was not in the original seed, it is evaluated against the original seed and integrated into it. If the creation happens within a given context in the semantic core, e.g. while working with a certain resource, the relations can be applied to the newly introduced component automatically. The final semantic core takes as input the relations between subcomponents and the overall design tree. Fig. 7 shows the process of setting up relations and the subsequent building of advanced relational semantics.

After the initial identification of concepts, they are statistically verified by using publicly available services and information resources like Google [11], Wikipedia [27], forums, product databases or web crawlers.

Identifying standing terms

Standing terms are composed of smaller units, but form a key term when combined. They are easy to identify. A search for standing terms is done by exact string match and delivers the following hit values:

sheet metal 10.2m deep drawing 8.6m

Singling out random terms

Equally, random terms with no meaning or no relevant meaning can be singled out or prepared for deep analysis by taking apart the components. For random terms or a non-standing term like "fin drawing" very low hit counts are returned:

Sheet metal xcad1	1 hit
Sheet metal cartest	70 hits
fin drawing	269 hits

Identifying and separating domains

The above approach can also be used to identify domains by searching for strongly connected concepts and dividing them from other hot spots. Since we use a global domain of all domains, there can only be a tendency of strong belonging computed, but not the final truth, since all domains are interconnected at some point without a static border separating them.

Table 2. Domain relations

	metal	extrusion	plastic	design
metal	-	3,8m	6,2m	13,9m
extrus.	263k	-	275k	4,1m
plastic	6,4m	261k	-	8,6m
design	25,6m	4,1m	8,7m	-

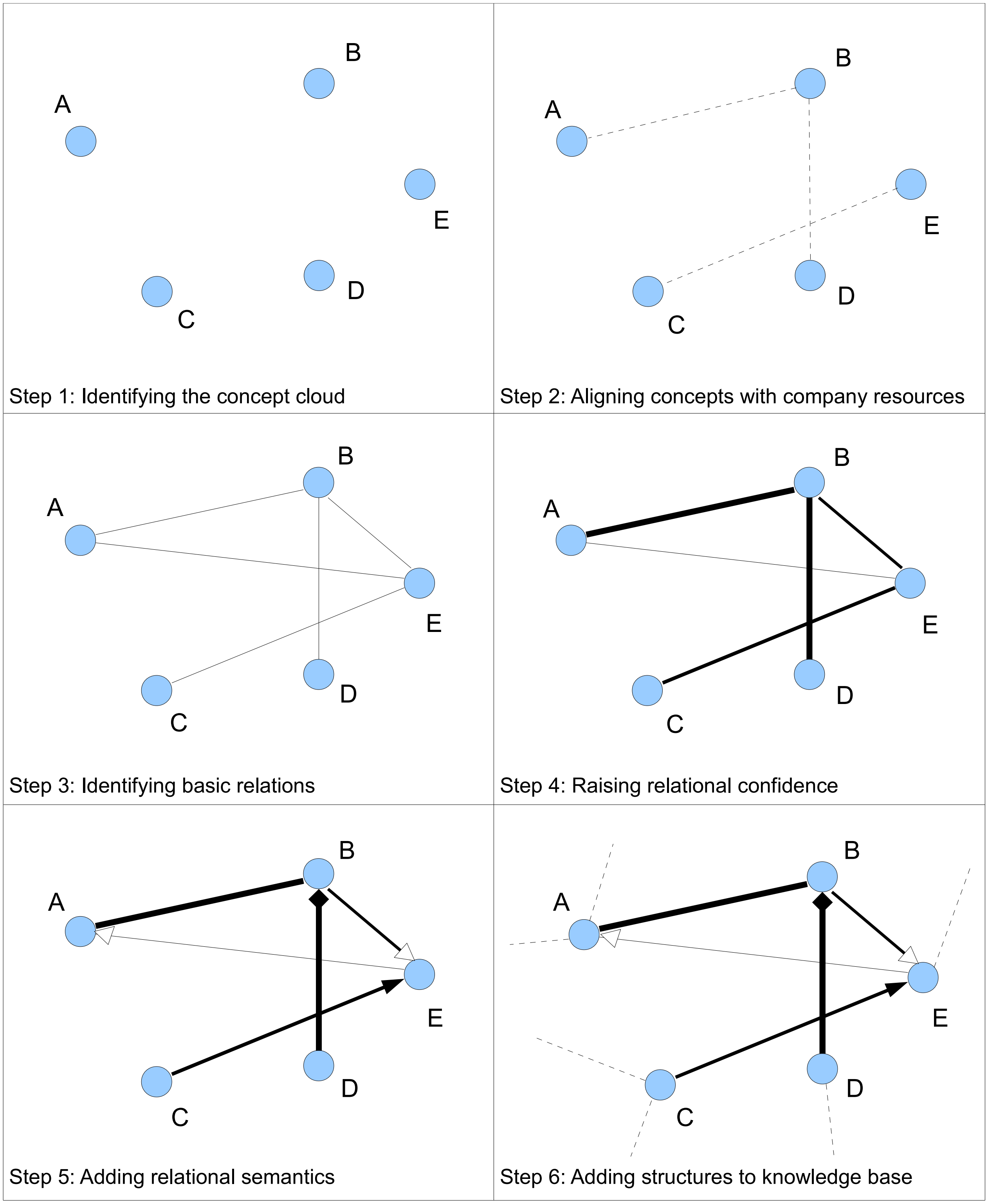

Figure 7. Setting up relation semantics between concepts and adding confidence

Domain relations

Combinations of terms reveal if they are semantically connected or not. A few examples of hit values by Google are given in table 2. It is interesting to note that different sequences of terms result in different hit values. The values can be cross checked with the inverse values, for example the combination of the terms “extrusion” and “plastic” delivers around 300k hits, while subtractive hit values are returned as:

extrusion -plastic 5,7m hits of 9,4m hits (extrusion)
plastic -extrusion 209m hits of 271m hits (plastic)

Domain division
Similarly as identifying the domain membership of a concept, the division and intersection points of domains can be computed (see table 3).

Table 3. Domain division and intersection

	metal	extrusion	plastic	design
crop	743k	115k	691k	1,2m
wheat	674k	172k	597k	741k
tractor	247k	126k	772k	925k
field	10,2m	3m	5,1m	41,8m

Reverse values confirm the findings of divided domains: metal -crop 54,3m / 619m total
metal -tractor 52,7m/619m total

Also intersections are visible since the term “field” is too general and appears in many domains. This is shown through high hit values of combinations within the domain of sheet metal working, but also within the domain of agriculture:

field crop 3,2m / field tractor 2,1m

The results strongly depend on the distribution of domain content in the search databases. The rough belonging to domains is computed as the sum of both ordered search results with a cutoff that filters random hits as: order_normal + order_negated - 2*cutoff > 0, where cutoff is empirical at around 900k and has to be related to the average distribution of positive hits.

Refined semantics
Based on the computed rough semantics from domain and non-domain hits, a dedicated domain source is added to provide higher semantic confidence between the recognized terms, using reliable information sources like encyclopedias or ontologies.

A list of paragraphs of the Wikipedia entry on „sheet metal” [27] is shown in fig. 8 with a section containing a classification of the context in fabrication and similar topics regarding metal working.

Contents
1 Processes
1.1 Stretching
1.2 Draw ing
1.2.1 Deep draw ing
1.3 Cutting
1.4 Bending and flanging
1.5 Punching and shearing
1.6 Spinning
1.7 Press brake forming
1.8 Roll forming
1.9 Rolling

Figure 8. Paragraphs about “sheet metal” [27]

The Wikipedia entry on „sheet metal” describes the relations to other topics:

“**Sheet metal** is simply **metal** formed into thin and flat pieces. It is one of the fundamental forms used in **metalworking**, and can be cut and bent into a variety of different shapes. [...] extremely thin pieces of sheet metal would be considered to be **foil** or **leaf**, and pieces thicker than 1/4 inch or a centimeter can be considered **plate**.“

Combined with the statistical information of the semantic relations and the involved terms, the concepts can be extracted. A simplified part of the extracted semantic graph for sheet metal and its surroundings is shown in fig. 9.

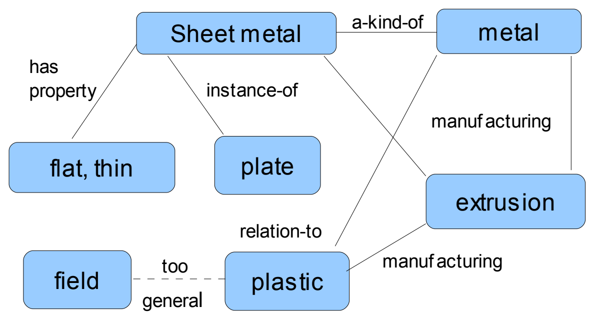

Figure 9. Semantic graph for sheet metal

8. Comments on Stability and Regression

The need of a semantic core is visible in most efforts to build an advanced CAD system. The requirement to build the domain by hand is usually met shortly after finalizing the building framework. The approach described in this paper can be applied to all domains where enough data exists to extrapolate domain dependencies and semantics of relationships between concepts. With regard to the stability of the discussed examples, the most important aspects are:

Reliability*:* the approach to use unfiltered web-content to compute concept distribution has the advantage to be able to use multi-domain information. The drawback is, that the correctness cannot be guaranteed, since all domains are mixed into a whole. Identified hot spots can be searched for in reliable sources of domain knowledge like encyclopedias. The problem is, that not all information has to be in there, which is in turn solved by the general multi resource approach, that considers multi domain information, which is supposed to be correct within the domain itself and then extended beyond domain borders.

Knowledge Acquisition*:* speed and reliability may be increased by using an ontology as the starting point and as a base to guide knowledge acquisition when building knowledge based systems. Ontologies have however the disadvantage that they are often fixed to a specific domain or even subdomain which makes it difficult to use them for a multi domain interactions.

Specification: if company resources are matched against the external information sources and used as a filter, then the domain knowledge of a company can be extracted as a specification for all involved information which can then be classified accordingly, so that the established structures of the company serve as an additional source of how to relate the information.

Re-Usability: the knowledge base is the foundation for a formal encoding of the important entities, attributes, processes and their relationships in the domain of interest. This formal representation supports the reuse of information and shared components in subsequent projects.

Search: the structure of the knowledge base may be used as meta-data, serving as an index for a repository of information, delivering meta-data for facilitating searches for product knowledge.

Maintenance and regression: the presented approach is able to reevaluate the content over time. This eliminates the need to maintain the system as it keeps itself up to date automatically and eliminates weak semantic links. By using a weighted voting algorithm to train and evaluate the confidence of semantic connections, a simple heuristic takes into account the interactions of the knowledge sources on the target relation. Relations found and verified in information resources place a weight on the relation, based on their relevance, as shown in table 4. Since the users have to use the knowledge base, they also have impact by voting (see table 5) to influence and reject structures. By using three areas of reached weight (see fig. 10), it is decided whether a relation is stable, discarded or still in the evaluation phase.

Table 4. Voting weights for sources

Source	Found	Not Found	Contrary found
Google	1	3	2
Wikipedia	2	2	2
OpenCyc	3	1	2

Table 5. Voting weights for users

	Agrees	Rejects	New Relation
Using	1	3	5
Browsing	2	2	5
Regression	3	1	5

This means that changing requirements can be automatically considered by the design methods based on the semantic core. When new company rules or norms become effective, the design system tells the designer where he is outdated.

Designs depending on machining aspects can be verified by linking the designs in the knowledge base and have it analyzed for dependencies. Transfer of ideas between designers is actively promoted since they all use the same knowledge base and are exposed to influences from their colleagues. Although all the above aspects can be applied for many design systems, the emphasis on a boot strapping setup for a multi domain knowledge base is a unique approach.

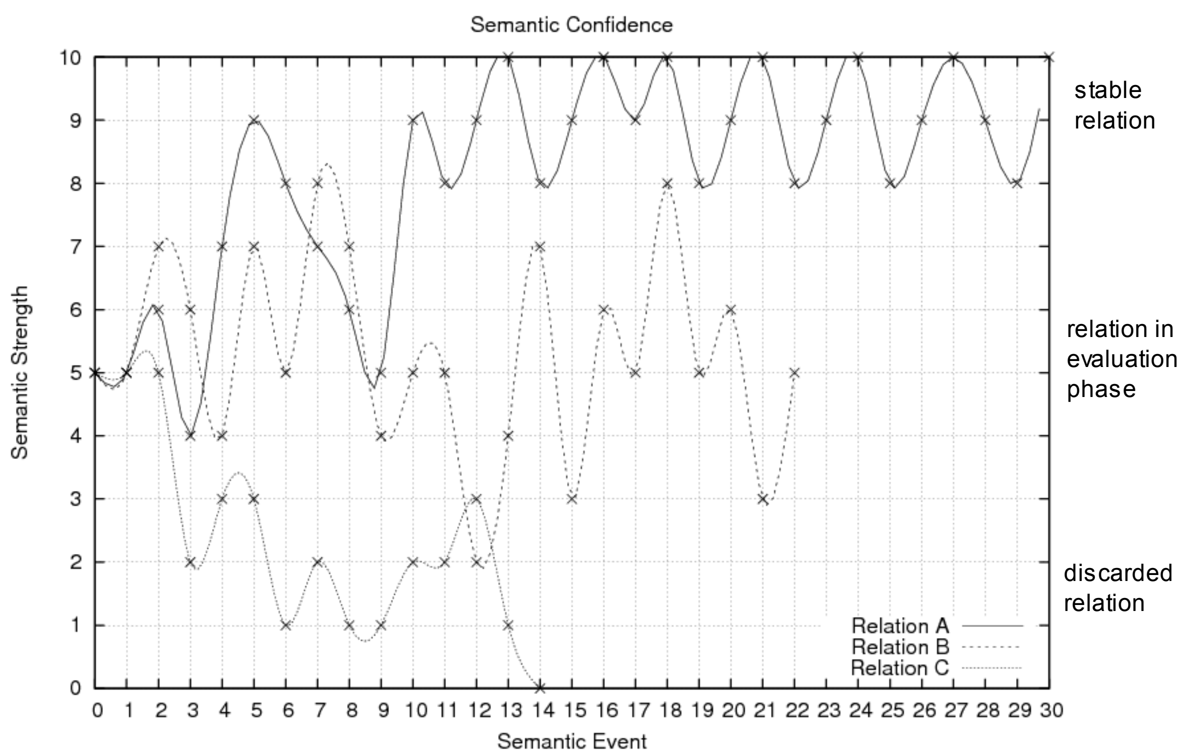

Figure 10. Phases of stable, evaluated and discarded relations

For a given context, the knowledge base supports content such as product models and geometry relations as well as company related standards, norms and rules. The explicit build-up and annotation that involves a user as active part has been avoided by the involvement of multi-domain information resources that are combined to extract a semantic confident relation.

9. Application and Conclusions

By using the actual work context in conjunction with the extracted statistical significant concept relations, matching patterns can be recognized and be used to deliver and apply relevant knowledge to the design process. In the following example (see fig. 7) of the design of a steering wheel, the size of the circles reflects the importance of a certain aspect, as derived from the merged conceptualization of a steering wheel from human and online sources.

The separation of the identified contexts is computed by the occurring frequency of its concepts. The names of the contexts are manually added to mark them as they can be identified through statistical clustering. The focus can now be given to a certain context, depending on where the user is executing the work. If a frequently use of forming concepts is detected, it can be deducted, that the work focus is in a certain design area and knowledge according to related concepts from this context is then presented.

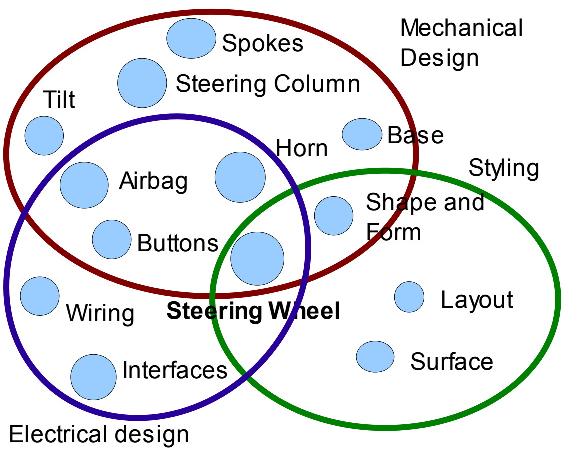

Figure 7. Context related to steering wheel design

Based on a common design context, qualifying situations have to be found in the knowledge base. Depending on the selected semantic method, the system identifies the geometric knowledge that needs to be applied to the user's problem. This information typically consists of a geometric object as discussed in the examples above. It can however be any information aspect that is stored in the knowledge base. After the identification of a geometric solution, it is extracted and transformed into a neutral form, since most certainly the design situation at hand has different parameters and dimensions, so a direct part transfer would not be possible. Out of the neutral form a solution is generated with respect to the actual used parameters. This solution is then proposed to the user who can accept or modify it to his needs.

The design context between two design projects has to match in order to separate relevant from irrelevant solutions. The design context is identified through buildup and use of geometric constructs, additionally to the non geometric information that also delivers characteristics to evaluate the design context. An incorrect design context would yield solutions that are not suited or not optimized for the design situation at hand and would need considerable effort to make them compliant.

If more than one solution has been identified, they have to be evaluated against each other to determine the best fitting expression. This is again based on design context, similarity of design situations and user requirements. The best fitting solution is determined by how strongly it needs to be transformed to the actual design situation and if it fits after the application of the given geometric parameters.

Once a relevant solution is found in another design project in the knowledge base, the solution has to be transformed and applied in a fitting manner into the current design situation. If the solution itself is already reused information or based on a part library, the modifiable parameters are reset and it is transformed into a neutral base form. This base form is then instantiated with the given parameter values.

If the solution is an original part design complex, the modifiable parameters have to be identified before it can be used in the new design project. The transformation takes place on several logical layers. Typical geometric transformations include rotation, translation and scaling, supported by feature transformations, that include the assignment or change of feature types to the involved object. Topological changes regarding the hull or boundary of the object may need to be applied.

A set of anchor points is recorded from the user's actions, that are the driving force in the transformation to decide how to configure the geometry that has to be integrated. The initial application of the solution to the design situation is done by feeding the available parameter information into matched anchor points of the solution object. The solution object is then configured and generates a variant according to its restrictions and parametric dependencies in the knowledge base. If an exact fitting solution cannot be determined, it will be left to the user to make the final adaptations.

For scenarios other than mechanical assembly, there has to be a considerable research effort to capture the domain specifics that are needed for a transforming operation.

10. References

[1] Opletal, S., Roller, D., Rüger, S., "Semantic Core to Acquire and Distribute Design Information", The 2nd Int. Conf. on Advanced Engineering Computing and Applications in Sciences (ADVCOMP'08), IEEE Computer Society Press, 2008

[2] Altmeyer J., Ohnsorge S., et. al., "Reuse of design objects in CAD frameworks", IEEE/ACM Intern. Conf. on Computer-Aided Design, 1994

[3] Baeza-Yates, R., Ribeiro-Neto, B., "Modern Information Retrieval", New York: ACM Press, Addison-Wesley, 1999

[4] Bjørner D., "Domain Engineering" in BCS FACS Seminars, Lecture Notes in Computer Science, BCS FAC Series (eds. Boca, P., Bowen, J.), Springer, 2008

[5] Cardie, C., "Empirical Methods in Information Extraction" in AI Magazine, Vol. 18, 4, 65-68, 1997

[6] Case, K., Graham, I.J., et. al., "CAD Genetic Algorithms for Evolutionary Form and Function Design" in Advances in Manufacturing Technology - XVI , Cheng, K., Webb, D. (Eds), Professional Engineering Publishing Ltd, Proc. of the 18th National Conf. on Manufacturing Research , Leeds Metropolitan University, UK, pp 103-107, 2002

[7] Dalakakis, S., Dieterich, M., Roller, D., Warschat, J., "Multiagentensystem zur Wissenskommunikation in der Produktentstehung-Rapid Product Develoment". Wirtschaftsinformatik 2005, "eEconomy eGovernment eSociety", Ferstel, O.K., Sinz, E.J., Eckert, S., Isselhorst, T. (eds.), Physica-Verlag, 2005

[8] de Moya-Anegón et. al., "Domain analysis and information retrieval through the construction of heliocentric maps based on ISI-JCR category cocitation", in Information Processing and Management 41, Elsevier, 2005

[9] Fischer, M., Staub, S., "Constructability reasoning based on a 4D facility model", in Proc. of Structural Engineering Worldwide. Elsevier Science Ltd., 1998

[10] Flycht-Eriksson A., "A domain knowledge manager for dialogue systems", in Proc. of ECAI 2000, ed. Horn W., Gesellschaft für Informatik, Berlin, 2000

[11] Google Search, Google inc., Mountain View, USA http://www.google.com/advanced_search?hl=en

[12] Kitamura, Y., et. al., "Functional Metadata Schema for Engineering Knowledge Management" in Proc. of WWW 2005, Chiba, Japan, 2005

[13] Macquarie, P. B., "Tacit Knowledge in Organizational Learning and Social Aspects of Technology", IGI Publishing, Hershey, USA, 2008

[14] Leonard, D., Sensiper, S., "The role of tacit knowledge in group innovation", California Management Review, Vol. 40 No. 3, 1998

[15] Pratt, M., Regli, W.C., et.al., "Manufacturing Feature Recognition from Solid Models: A Status Report", in IEEE Transaction on Robotics and Automation, Vol. 16, No. 6, 2000

[16] Mok, C. K., Hua, M., et.al., "A hybrid case-based reasoning CAD system for injection mould design", Int. Journal of Production Research, 1 – 18, 2007

[17] Ogden C.K., Richards I.A., "The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism", 8th ed.; New York: Harcourt, Brace &World, 1946

[18] OpenCyc Ontology, OpenCyc Knowledge Base Copyright 2001-2008, Cycorp, Inc., Austin, TX, USA http://www.opencyc.org/

[19] Opletal, S., Dalakakis, S., Roller, D., "Towards Semantic-Based CAD user Interface and Core Components", in "Applications of Digital Techniques in Industrial Design Engineering", Proc. of the 6th Int. Conf. on Computer-Aided Industrial Desingn & Conceptual Desingn, Eds: Pan, Y., Vergeest, J., Lin, Z., Wang, Ch., Sun, S., Hu, Z., Tang, Y., Zhou, L., Int. Academic Publishers, Beijing World Publishing Corporation, Beijing, 2005

[20] Opletal S., Roller D., et.al., "Pro-active environment for assisted model composition", Proc. of Cooperative Design, Visualization, and Engineering (CDVE07), Springer, 2007

[21] Piskorski, J.; Xu F., et.al., "A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Boot-strapping", in Proc. of the 3rd Int. Conference on Language Resources an Evaluation (LREC), 2002

[22] Ponsen, M., Munoz-Avila, H., et.al., "Automatically Acquiring Domain Knowledge For Adaptive Game AI Using Evolutionary Learning" in Science of Computer Programming, Volume 67, Issue 1, pp. 59-75 Special Issue on Aspects of Game Programming, 2007

[23] Rembold, U., Nnaji, B.O., Storr A., "Computer Integrated Manufacturing and Engineering", Addison-Wesley, UK, 1994

[24] Roller, D., Eck, O., Dalakakis, S., "Knowledge based support of Rapid Product Development" in Journal of Engineering Design, Taylor & Francis Ltd, Vol. 15, No. 4, 2004

[25] Singh V., "The Cim Debacle: Methodologies to Facilitate Software Interoperability", Springer, 1997

[26] St-Cyr O., Yves Lespérance Y., et.al., "An Intelligent Assistant for Computer-Aided Design" Proc. of AAAI 2000 Spring Symposium Series on Smart Graphics, Stanford, USA, 2000

[27] Exemplary Wikipedia page on "sheet metal", Wikimedia Foundation Inc., San Francisco, USA http://en.wikipedia.org/wiki/Sheet_metal

[28] Feiyu, X., Uszkoreit, H., Li H., "A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity" in Proc. of ACL 2007, Prague, 2007