

Secondary Publication



Weber, Nadine; Eckstein, Raiko; Henrich, Andreas

Searching Multiple Artifacts : a Comprehensive Framework for Complex Search Situations

Date of secondary publication: 14.02.2025

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-1063780

Primary publication

Weber, Nadine; Eckstein, Raiko; Henrich, Andreas (2009): Searching Multiple Artifacts : a Comprehensive Framework for Complex Search Situations, in: Troels Andreasen, Ronald R. Yager, Henrik Bulskov, u. a. (Ed.), Flexible query answering systems : 8th international conference, FQAS 2009, Roskilde, Denmark, October 26 - 28, 2009 ; proceedings, Berlin u.a.: Springer, pp. 251–262, doi: 10.1007/978-3-642-04957-6_22.

Legal Notice

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holders.

This document is made available with all rights reserved.

Searching Multiple Artifacts: A Comprehensive Framework for Complex Search Situations

Nadine Weber, Raiko Eckstein, and Andreas Henrich

Chair of Media Informatics, University of Bamberg,
96052 Bamberg, Germany

{nadine.weber, raiko.eckstein, andreas.henrich}@uni-bamberg.de
<http://www.uni-bamberg.de/en/minf>

Abstract. The paper presents a comprehensive search framework that deals with different types of artifacts and therefore is suitable for situations where the information need behind a search request is sketchy. Such situations are characteristic for the product development domain, on which our research is focused. To support design engineers, various search concepts for specific computer aided engineering (CAE) document types, like e.g. 3D shape similarity methods, have been proposed. Although these methods might be beneficial for certain situations, there is a strong demand for a more generic approach that enables goal-oriented as well as exploratory searching for multiple artifacts such as products, documents, and materials. Hence, we base our framework on the ideas of faceted search, ranking, query-by-example, and parallel coordinates. Furthermore, we extend these concepts by artifact type hierarchies facilitating the generation of artifact type-specific facet data and the adaptability of this integrated approach for other domains as well.

Keywords: Framework, Faceted Search, Ranking, Query-By-Example, Artifact Type Hierarchies, Parallel Coordinates, Product Development.

1 Introduction

The process of developing products consists of multiple, highly creative steps in which a lot of information is created and needed. This information is stored in a variety of systems such as product data or lifecycle management systems (PDM/PLM), enterprise resource planning systems (ERP), computer aided design (CAD) databases, or document management systems (DMS). Besides, also file systems are commonly used archives. Both individuals and project teams utilize them for saving or sharing relevant and useful documents. Consequently, design engineers are overwhelmed with information and do not know where to search in order to satisfy their information needs.

In the literature, several researchers met the challenge of supporting design engineers with useful information and proposed various search concepts [1,2]. CAE-specific documents in particular turned out to be an interesting field of investigation resulting in concepts that are limited to certain types of documents.

On the one hand, a lot of research is done in the field of similarity search methods for CAD models. Using the shape description of the contained three-dimensional (3D) object allows to compare the similarity of the object with other objects in a given data collection. Both, Tangelder et al. [3] and Iyer et al. [4] give a comprehensive survey of existing methods. On the other hand, there are approaches that support engineers by retrieving technical drawings as summarized in [5]. These specific retrieval algorithms are beneficial for situations in which the engineer knows exactly what he needs. For instance, by searching with an example CAD model, an engineer can find similar models or alternatives, he can reuse, in a goal-oriented way. This contributes to the avoidance of reinventing the wheel and at the same time to a reduction of construction time and costs.

But, product development is also characterized by situations where the information need behind a search request is sketchy, i.e. the engineer is unaware of what he is looking for in detail and therefore describes his information need in a vague fashion. This is shown by two example scenarios that we identified in collaboration with our industry partners. Although the design engineer searches for a certain product in the first step, his intrinsic aim is to get knowledge about the product's supplier – especially his range of articles – to benefit from substitutional potentials and to achieve higher order sizes. Another example concerns the task of coping with a critical requirement for a product, like e.g. its thermal resistance to a value of 70 degrees Celsius. For this purpose, the engineer needs information how to achieve this demand. Thus, test reports e.g. for a particular material or solutions made in prior projects can be conceivable information sources. As these scenarios show, design engineers are interested in various kinds of CAE artifacts. Besides documents, product data in particular, but also other artifact information (e.g. about persons, projects, ...) is requested. Since this variety manifests itself in complex search situations, a more generic approach is required that allows a goal-oriented as well as an exploratory search procedure.

As an approach for such complex search situations, we presented in [6] an interactive retrieval model that is based on multiple ideas. To support both goal-oriented and exploratory search, we draw on the original idea of faceted search by offering selection criteria. Moreover, we use a parallel coordinates plot as a method for issuing faceted search queries and for displaying the characteristics of the retrieved results. This framework is extended in [7] by ranking criteria and especially by means of query-by-example (QbE) conditions to integrate methods that allow a similarity based search. Since our search framework has to provide information about several artifact types as mentioned above, the present paper deals with their integration. Therefore, we explain the concepts of our comprehensive framework in Section 2 which gives an impression of our current prototype of a graphical user interface (GUI). Afterwards, Section 3 deals with several aspects of the integration. The automatic generation of specific artifact descriptions on the basis of artifact type hierarchies, their validation and verification, as well as the inclusion of artifact relationships to enhance the search functionality are addressed in the associated subsections. Finally, a conclusion summarizes the paper and details some future work.

2 Framework Concepts in a Sketch

Our framework is based on the idea of a faceted search that represents artifacts by means of so-called facets. A facet represents a particular property or dimension of an artifact and serves as a selection criterion. Therefore, it consists of facet values from which a single value or multiple values can be assigned to an artifact. Thus, the combination of all facets, each instantiated by certain facet values, gives a complete description of a certain artifact [8]. By selecting such facets, or facet values respectively, the user can narrow down the search result and find relevant artifacts in both a goal-oriented and an exploratory way. Moreover, a quantitative overview of the available data is given to the user by means of facet value counts which show to how many artifacts this facet value is assigned. [9]

Although this kind of search mechanism is popular for e-commerce applications, like e.g. <http://www.wine.com>, its functionality is unsatisfactory for a retrieval system utilized by design engineers. For this reason, we enhanced the basic concept of a faceted search by three aspects which are illustrated in a screenshot of the GUI of our search prototype (cf. Figure 1). This screenshot exemplifies a user's search request for products defined by four facets, namely *artifact type*, *3D geometry*, *product group*, and *outside radius*. Such a search request is generated by choosing facets from the ribbon band *Facets* in the topmost section of our GUI. Since we have to support different artifact types, the contained facet list may only consist of applicable facets that do not lead to zero results when selected, and therefore has to be updated constantly. What kind of facets are available in this list depends on the considered artifact type as well as on the user's previous selections. This aspect of artifact type-specific facets is described in Section 3 in more detail.

After the user has chosen a certain facet, it is displayed in the middle section of our GUI as the main part of the user interface. It consists of a parallel coordinates plot, originally developed by Inselberg [10], that enables the visualization of multiple dimensions in a two-dimensional (2D) way. Thus, the parallel coordinates plot depicts every selected facet as a separate axis holding the related facet values and therefore represents the user's search request by a changeable number of parallel axes. After adding a facet to this plot, an engineer can refine his request in different ways. On the one hand, he can select one or more values of the chosen facet leading to a reduction of the result set. This basic procedure is shown for the facets *artifact type* and *product group* in Figure 1, whose chosen facet values are the only remaining points on the axes. Here, the user narrowed down his search by filtering for products which are "disks" or "o-rings". Moreover, the design engineer can add or remove facets in any way to modify the query. Since this form of applying facets establishes a simple filtering on the complete data set, the result set is only an unstructured multitude of artifacts that all fulfill the selected criteria. Especially in the product development domain these sets can be very large and hence confuse more than help. To improve the quality of the search results, we provide the user with the application of ranking criteria. These criteria allow the transformation of an unstructured result set into

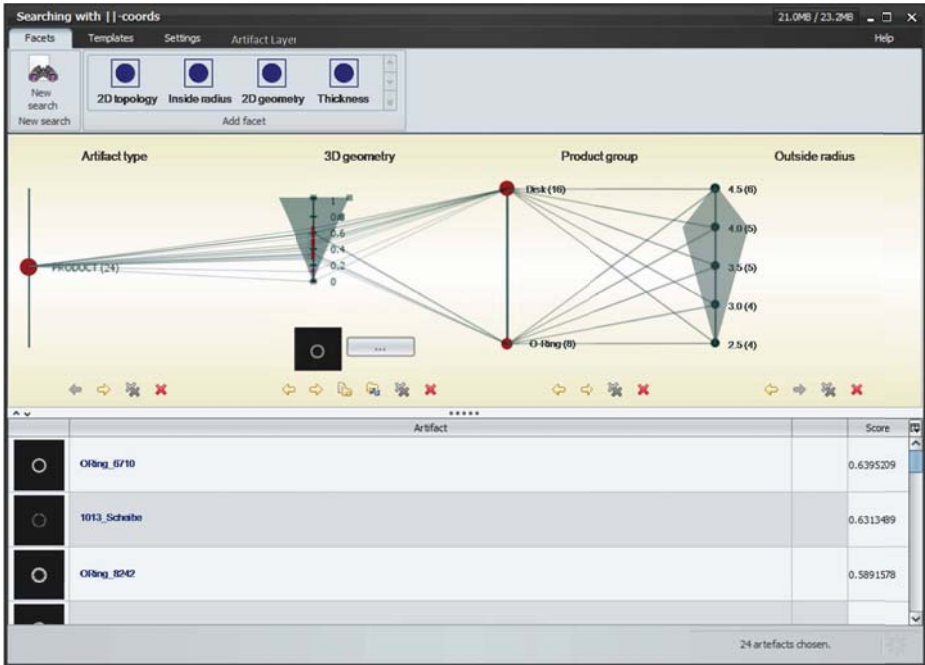


Fig. 1. GUI screenshot of our search prototype

a sorted list and therefore illustrate why some search results are more relevant for a given query than others. Moreover, they enable the user to influence the ranking according to his information need. Our example in Figure 1 shows such a ranking criterion for the facet *outside radius*. Defined as a preference function illustrated as a gray shape (e.g. quadrilateral, triangular, ...), it overlays the facet's axis. This function expresses the user's preferences for specific facet values or value ranges and leads to a ranking of the search results. In Figure 1, the given quadrilateral function defines that the user prefers mainly products with an outside radius of 4.0, but is also interested in other values. Whereas he assigns lower weights to outside radii of 4.5, 3.5 and 3.0, he intrinsically does not look for products with an outside radius of 2.5.

As another necessary feature for design engineers, we extended the basic faceted search concept with QbE conditions that support different types of similarity comparisons. They include a text or keyword based search functionality whose absence is unimaginable these days. But also other types of similarity search methods – especially for the product development domain – have to be available. Searching for example for similar products by means of their geometrical shape helps engineers to reuse existing products. The facet *3D geometry* in Figure 1 demonstrates such a QbE condition. After the user has uploaded an example object (a 3D model in this case), all artifacts in the result set are ranked by the similarity of their 3D geometrical description compared to this example

object. For this approach, we generate appropriate representations dependent on the respective similarity aspect that is addressed. For the 3D geometry aspect, we draw on concepts developed by Vranic et al. [11]. When it comes to 2D, we use representations according to [12] for the topology and to [13] for the geometry aspect. By means of these representations, we calculate a similarity value for each pair of artifact and example object by means of a distance metric [14]. These similarity values, ranging from 0 to 1 (with 1 denoting the identity of two representations), provide an initial ranking of the result set. For this purpose, QbE conditions are overlaid with a ranking criterion by default. The associated preference function has a triangular shape, illustrating that facet values near 1 are ranked higher than facet values near 0. But of course, this ranking can be influenced by the user, too.

All in all, every action of the user modifies the result set which is displayed in the bottom section of our GUI. This section contains a list of all search results detailed by an artifact description which is composed of an artifact identifier (e.g. a product's part number or a document's file path), a score that indicates the relevance of the search result for the given query, and an artifact preview if available.

3 Integrating Multiple Artifact Types

As stated in Section 1, an engineering context is characterized by various types of artifacts. Since the development of products is the main objective of design engineers, searching for reusable products and components in certain situations is an important procedure. Moreover, products are primarily described by documents such as CAD models, technical drawings or technical specifications. These so-called product models as well as general documents (e.g. guidelines or engineer standards) and project-related documents serve as significant information sources, too, that can help to accomplish a current job. In other situations, the engineer needs to figure out material properties or persons/experts who dealt with some problems before. Hence, searching for these artifacts of different types within our framework calls for adequate artifact descriptions that comprise facets typical for a particular artifact type.

In the following subsections, we detail artifact descriptions for exemplary artifact types by presenting first what type-specific information characterizes such a description (cf. Section 3.1). Secondly, Section 3.2 shows from where and how this data can be extracted automatically and what problems have to be taken into account in doing so (cf. Section 3.3).

But not only type-specific artifact descriptions have to be considered. Since products are described in documents which are created and maintained by persons who are involved in projects, there is a network of interconnected artifacts. This network contains useful information and therefore has to be integrated in a search framework as proposed above. Section 3.4 deals with these relationships more precisely and shows how they can be embedded wisely for improving search functionality.

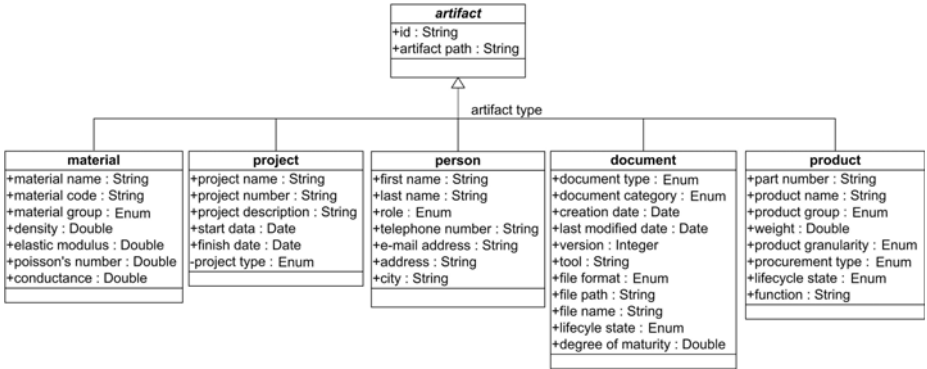


Fig. 2. Hierarchy of artifact types considered in our prototype with some exemplary characterizing facets

3.1 Artifact Descriptions Based on Artifact Type Hierarchies

Every single artifact type has other characteristic parameters that describe this type more precisely. Whereas a document is characterized for example by its document type, file format, creation date, or version, a product is described by its part number, the product group it belongs to, its weight, and other descriptive properties. For this reason, we have to integrate artifact type hierarchies in our framework. They serve as input factors for both the indexing and the search component of our framework because they define which facets characterize an artifact of a certain type. Figure 2 depicts the overall hierarchy of artifact types considered in our prototypical implementation with some example facets defined for these types. Whereas facets in the leaf nodes of the hierarchy are only valid for special kinds of artifacts (e.g. *start* and *finish date* for projects), facets of the hierarchy's root are inherited to all of its subclasses. Thus, every artifact is identified by three parameters, namely an identification number, a path where the original artifact is archived (e.g. path to PDM for products or to DMS for documents), and – according to UML – the ‘virtual’ attribute *artifact type*, which defines the specialization of the artifacts.

But not only artifacts of different types have to be distinguished. Artifacts of the same type can possess various facets, too. This diverseness depends on the substance of certain facets. For example, seals with a circular shape can be described by their outside and inside radii. This is not necessarily valid for products belonging to other product groups like e.g. screws. Consequently, every product group has its own specific facets resulting in a separate product hierarchy exemplified in Figure 3. Since the facet *product group* is a particular facet that defines a further specialization of artifacts, it is indispensable to select a value for it for every product. By means of this *discriminative* facet, the indexing component of our framework knows which path in the hierarchy has to be considered to generate valid artifact descriptions and our search component can decide which facets can be offered for further selections. Thus, we have to differentiate

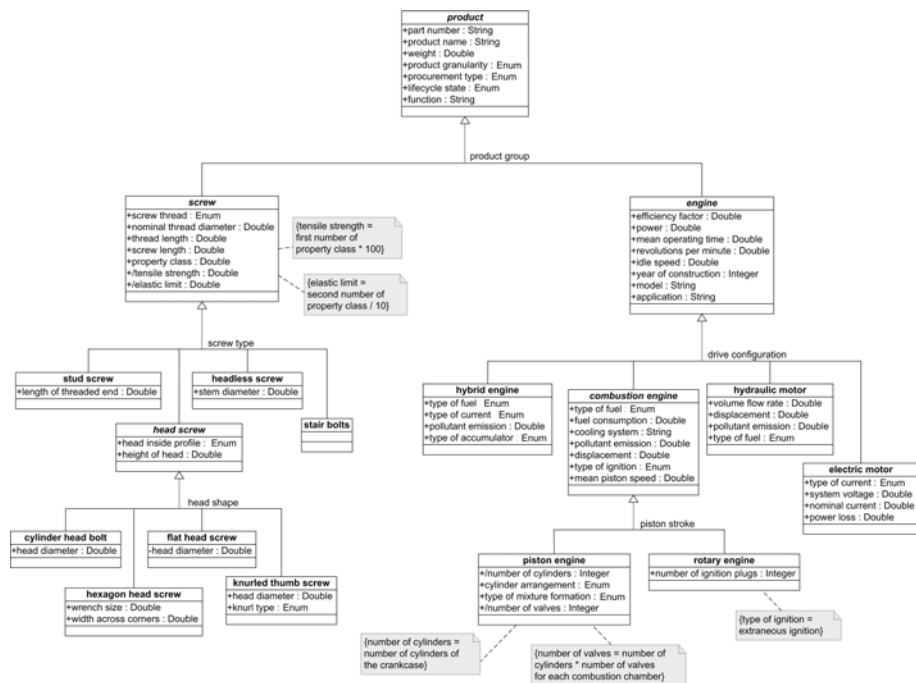


Fig. 3. One exemplary product hierarchy specialized by two selective product groups, namely screws as representatives for parts and engines as representatives for assemblies

between facets that are discriminatory for certain artifacts and facets that are not. How these conditions can be assured is explained in Section 3.3. Other examples for discriminative product facets are the type of a screw (cf. Figure 3), which divides screws by means of their shape into head screws, stud screws, . . . , or the procurement type of a product. Dependent on the fact whether a product is self-made or purchased, either the retail price or the purchase price will be of interest. Of course, the definition of such discriminative facets concerns other artifacts as well. Reliant on a person's role (Is he/she e.g. employee, supplier, or customer?) or a document's type (Is it a CAD model or a technical drawing?) different facets have to be available as search criteria.

Furthermore, there may be facets that can be derived from other facet values. In Figure 3, the screw facet *tensile strength* is an example for such a facet, since its value is calculated on the basis of the screw's property class. By defining a calculation rule (given in the uppermost gray box in Figure 3), this facet has not to be extracted separately, but can be determined automatically. Although the associated rule is quite simple in this case, these rules can be very complicated especially for assemblies with facets depending on multiple other facets (cf. the rule for the facet *number of valves* of a piston engine in the lowermost gray box in Figure 3). Besides these rule-based criteria it is also possible that some

facet values are fix when considering a certain artifact. Every rotary engine for example comes with an extraneous ignition as value for the facet *type of ignition*.

The definition of both discriminative and non-discriminative facets, maybe together with calculation rules or fixed values, results from the design of the artifact type hierarchies which can be customized according to individual or entrepreneurial priorities. This feasibility points out a further advantage of our search framework which is adaptable for other application areas, too.

3.2 Extractor Components to Create Facet Data

The automatic generation of artifact descriptions by means of facets calls for an indexing component that collects artifact-specific data as exhaustive as possible. As stated in previous sections, this data is contained in different information sources. On the one hand, there are multiple systems focused on products and their properties such as PDM/PLM and ERP. But also documents in DMS, file systems, or databases provide information not only about themselves, but also about products or persons. This necessitates particular extractor components for both existing systems and document formats. For our prototypical implementation, we concentrated on extractor components for a subset of product developmental documents – especially product models like CAD models, technical drawings and bills of materials – and researched what artifact type-specific information can be extracted from these documents.

Since the intention of all of these product models is to describe a product in more detail, they allow for the extraction of document-specific as well as product-specific facets. Document-specific information comprises mainly metadata such as the document's category, type, version, creation or last modified date, the tool the document was created in (e.g. the name and release of the used CAD system), and its file format. Additionally, it can include document type-specific data like e.g. the drawing number or scale for technical drawings.

Although every product model has to include data that identifies the described product (ideally its part number), product-specific facets depend primarily on the document's type. A CAD model for example has the purpose of describing a product's geometrical shape in the 3D space. This information can be used to generate an appropriate representation and therewith offer a product facet with a QbE condition. Although the 3D geometry description is often considered as a special property of CAD documents in the literature, we decided to take it as a product facet. Since design engineers in particular have a strong product-oriented view, they rather look for similar products than for similar documents. But of course, this is no restriction for possible queries since each product is linked to all associated documents as explained in Section 3.4. In contrary to CAD models, technical drawings contain both a geometrical and a topological description of a product, but in the 2D space. Their consideration enriches the product's facet description by two further facets with QbE conditions, namely the facets *2D geometry* and *2D topology* as displayed in Figure 1 in the ribbon band *Facets*. Another product facet called *number of subcomponents* results from the fact that CAD models can also contain the product structure for an assembly.

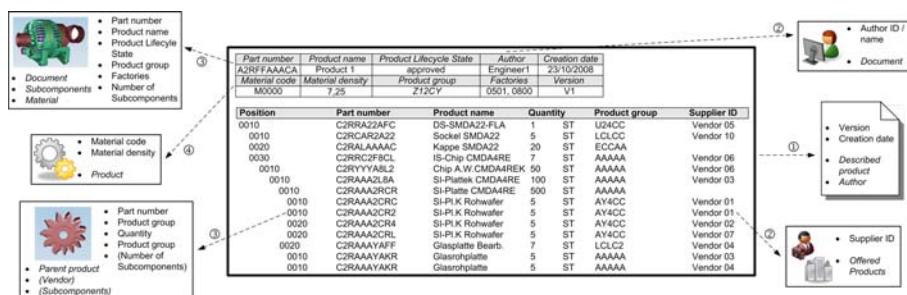


Fig. 4. Conceptual extractor component for bills of materials that generates facet descriptions for four types of artifacts: one document (1), multiple persons (2), several products (3), and one material (4). Whereas the items in italic notation indicate the relations to other artifacts, the items in normal notation describe the artifact facets.

This information is also available from a bill of materials (BOM) whose main purpose is the quantitative listing of all components of a product. Dependent on the entrepreneurial definition of a BOM's content, this document type contains artifact descriptions for multiple products (i.e. for the main product ID and for each of its subcomponents) and can include e.g. supplier-specific data for purchased products as well.

Furthermore, product models can incorporate information about the material a product is made of. This allows for the extraction of material-specific facets like e.g. the material code or name as well as other specific data (e.g. density, elastic modulus, ...) if available.

Besides these three types of artifact descriptions (product-, document-, and material-related), a fourth one exists. Product models – as well as all other types of documents – are created by one or multiple persons who are referenced in the document's content (e.g. in the title block of technical drawings) or at least in the document's properties. As a result, every document enables the generation of one or multiple person-specific artifact descriptions according to this information.

As a consequence, documents – in particular product models – contain facet data for various artifacts. Which ones depends on the document's content resulting in extractors that have to be customized for the existing standards. Figure 4 illustrates such an extractor component using the example of a bill of materials. This extractor generates artifact descriptions for artifacts of four different types (marked by numbers in circles): one for the document itself, one for the document's author, one for the main product it describes, one for the material the main product is made of, one for each subcomponent of the main product, and one for each supplier of a product. As this listing shows, the extracted artifacts correlate. Since these correlations hold useful information (see Section 3.4), every extractor component, that generates more than one artifact description, has to save the existing relationships, too. For completeness, they are included in italic notation in Figure 4.

3.3 Verifying Extracted Facet Data

Since we want to provide the user with as much information as possible, the description of a certain artifact should integrate data from every available information source. But this brings up some problems regarding mainly the incompleteness or impreciseness of data as well as the existence of several facet values. For this reason, it is necessary to integrate a component that checks the consistency of the extracted facet data. This component works on the basis of a facet schema which consists of two parts. The first part defines every single facet by determining a name, an id, the associated data type, and the ability of being a multivalued facet. Being a multivalued facet means that not only one but multiple values of the facet can be assigned to an artifact. For example, an engine's *function* can be described by several facet values like *alter energy*, *activate*, and *perform mechanical work* and therefore has to be a multivalued facet. The second part of the schema contains the artifact type hierarchies as explained in Section 3.1 establishing for each artifact type which facets are required.

By means of this schema, the consistency checker is responsible for the validation and verification of the extracted facet descriptions. Validation means that the consistency checker has to assure that all discriminative facets for a given artifact are available. Only in this case, indexing (and also updating) of the artifact is possible at all. Otherwise, if one or more discriminative facets are missing, the consistency checker should try to collect the missing information from other sources (e.g. from the PDM system) to avoid a loss of valuable data. Verification on the other side stands for the checkup of each facet's data type and multiplicity. In spite of the fact that a facet is not a multivalued one, it is possible that different information sources have different values for this facet. Thus, the consistency checker has to determine which value is the valid one. Configured with customizable rules, the consistency checker knows for example that he should always prefer data extracted from the PDM system or data that was extracted at a later date. If no applicable rule is available, a human expert should be involved to make a decision.

Moreover, the mentioned facet schema is a necessary input for our search component. The defined artifact type hierarchies together with their associated facets support the update of the facet list in the GUI. Dependent on the hierarchy level a user's search request refers to, the search component knows which facets are possible for a further selection without obtaining zero results.

3.4 Artifact Relationships and Their Influence on Searching

As mentioned above, there is hardly any artifact that has no relationships to other artifacts. This brings up a network of interconnected artifacts that can be very complex in some domains. In the product development domain, a product is either a part or an assembly that is made of a certain material. Furthermore, it is described by a product model as a special kind of document, which was created and maintained by persons. They in turn are involved in projects that were initialized with the aim of developing a particular product.

All these relationships hold valuable information for design engineers and therefore have to be integrated in the proposed search framework. We utilize these relationships to improve the search functionality by giving the user the opportunity of changing artifact layers in any way during the search process. This enables the system to deal with complex information needs as in the following example. Let us assume an engineer is looking for products in the first step by selecting facets applicable for his current information need. Ended up with a first satisfying result set, he is then interested in all documents that describe this product. To satisfy this request, it has to be possible for the user to change from the product layer to the document layer, where he can continue his search by filtering on the new result set of product-related documents.

For the integration of such a functional extension in our prototype, we explore different alternatives at the moment. One possible way is the definition of special facets which are responsible for the change (e.g. a facet called *associated documents*). Another alternative could be the use of a context menu for the result set which contains all accessible artifact layers. They have to be determined by analyzing and matching all relationships of the artifacts in the result set to avoid empty results. In short, the semantics is to select a relationship type and then, the result set is switched from the source objects of this relationship to its destination objects.

4 Conclusion and Future Work

Our proposed search framework for complex and sketchy information needs can deal with multiple artifact types. Based on the idea of a faceted search, this is achieved by generating appropriate artifact descriptions by means of artifact type hierarchies that determine which facets describe a certain artifact. As an example, this procedure was shown for the domain of product development, where not only documents, but also products, materials, and other artifacts are interesting for an engineer to cope with his work. But of course, there is no restriction to special domains since artifact type hierarchies can be customized for every other field of application.

Although our framework is sophisticated in many ways, there are still some aspects that have to be addressed. One of them is the enhancement of the search functionality by using artifact relationships as proposed in Section 3.4. Furthermore, we want to investigate several possibilities for a more detailed visualization of the result set. The illustration of suppliers or product groups of the result set (e.g. in form of a tag cloud [15]) as well as a *details-on-demand* function that enables the user to get more details of a certain result [16] (e.g. a product's structure or an explanation why a result is relevant for a given query) are useful benefits that should be integrated. Another important task concerns the evaluation of the framework. Even though our industry partners look upon the presented concept favorably, it is necessary to test the system's performance in real time situations.

Acknowledgments. Our work is part of the Bavarian research cooperation FORFLOW promoted by the Bavarian Research Foundation (BFS). For more details, we refer to <http://www.forflow.org>.

References

1. Karnik, M.V., Gupta, S.K., Anand, D.K., Valenta, F.J., Wexler, I.A.: Design Navigator System: A Case Study in Improving Product Development Through Improved Information Management. In: Proc. of 25th Computers and Information in Engineering Conference, pp. 291–312. ASME (2005)
2. Lou, K., Jayanti, S., Iyer, N., Kalyanaraman, Y., Prabhakar, S., Ramani, K.: A Reconfigurable 3D Engineering Shape Search System Part II: Database Indexing, Retrieval and Clustering. In: Proc. of 23rd Computers and Information in Engineering Conference, pp. 169–178. ASME (2003)
3. Tangelder, J.W.H., Veltkamp, R.C.: A Survey of Content Based 3D Shape Retrieval Methods. In: Proc. of Int. Conference on Shape Modeling and Applications, pp. 145–156. IEEE Computer Society, Los Alamitos (2004)
4. Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K.: Three-dimensional Shape Searching: State-of-the-art Review and Future Trends. *Computer Aided Design* 37, 509–530 (2005)
5. Weber, N., Henrich, A.: Retrieval of Technical Drawings in DXF Format - Concepts and Problems. In: Workshop Proceedings of Lernen - Wissen - Adaption, pp. 213–220. University of Halle-Wittenberg (2007)
6. Eckstein, R., Henrich, A.: Reaching the Boundaries of Context-Aware IR: Accepting Help From the User. In: Proc. of 2nd Int. Workshop on Adaptive Information Retrieval (2008)
7. Eckstein, R., Henrich, A.: Visual Browsing in Product Development Processes. In: Proc. of 17th Int. Conference on Engineering Design (to appear, 2009)
8. Denton, W.: How to Make a Faceted Classification and Put It On the Web (2003), <http://www.miskatonic.org/library/facet-web-howto.html> (May 18, 2009)
9. Yee, K.-P., Swearingen, K., Li, K., Hearst, M.: Faceted Metadata for Image Search and Browsing. In: CHI 2003: Proc. of SIGCHI Conference on Human Factors in Computing Systems, pp. 401–408. ACM Press, New York (2003)
10. Inselberg, A.: The Plane with Parallel Coordinates. *The Visual Computer* 1(4), 69–91 (1985)
11. Vranic, D.V.: 3D Model Retrieval. Ph. D. Dissertation. University of Leipzig (2004)
12. Fonseca, M.J., Jorge, J.A.: Towards Content-Based Retrieval of Technical Drawings Through High-Dimensional Indexing. *Computers and Graphics* 27(9), 61–69 (2003)
13. Fränti, P., Mednonogov, A., Kyrki, V., Kälviäinen, H.: Content-Based Matching of Line-Drawing Images Using the Hough Transform. *Int. Journal on Document Analysis and Recognition* 3(3), 117–124 (2000)
14. Zezula, P., Amato, G., Dohnal, V., Batko, M.: *Similarity Search: The Metric Space Approach*. Springer Science + Business Media, New York (2006)
15. Hassan-Montero, Y., Herrero-Solana, V.: Improving Tag-Clouds as Visual Information Retrieval Interfaces. In: Int. Conference on Multidisciplinary Information Sciences and Technologies (2006)
16. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proc. of IEEE Symposium on Visual Languages, pp. 336–343. IEEE Computer Society Press, Washington (1996)