Semantic search in a process-oriented knowledge base

Daniel Kimmig*, Andreas Schmidt†*, Klaus Bittner*, and Markus Dickerhof*
*Institute for Applied Computer Science
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: {daniel.kimmig, andreas.schmidt, klaus.bittner, markus.dickerhof}@kit.edu
†Department of Informatics and Business Information Systems,
University of Applied Sciences, Karlsruhe
Karlsruhe, Germany
Email: andreas.schmidt@hs-karlsruhe.de

***Abstract*—*MinaBASE*, a process-oriented knowledge management system, currently features a simple full-text search as well as a more sophisticated expert search. While the former is much easier to use, the latter has much higher precision and recall characteristics due to a more content-aware filter-mechanism. In this paper we present a combination of both approaches, which preserves the intuitiveness of the full-text search while offering the same precision and recall as the expert search. This is achieved by using a single input box for entering queries and semantically evaluating the given items according to the ontological concepts and relationships of the knowledge base and giving respective automatic suggestions. In contrast to regular autocompletion-widgets, the suggestions are not simple keywords, but rather elements of taxonomies as well as numeric input boxes for specification of their properties.**

***Keywords*-process knowledge management, microsystems technology, semantic search**

I. Introduction

At our institute we developed *MinaBASE*, which is a process-driven knowledge management system, that models manufacturing processes of microsystems technologies [1]. For information retrieval purposes an expert search has been implemented as part of it. This filter-mechanism offers several input boxes for the different types of properties of the information objects in *MinaBASE*, to define several interconnectable restrictions, which must be fulfilled by the entries of the result set. The input boxes support an automatic completion of values which are valid for the specific types of properties. The goal of this completion is to support the user while defining the restrictions. The restrictions are then mapped to database JOINs using a Criteria-API of a object relational mapping tool, with the result of a high precisioned search due to referential integrity. Since it is necessary to switch between the various input boxes with different meanings, it can easily become tedious to define multiple restrictions. The automatic completion is a helpful feature, but for their suggestion it is absolutely necessary for the user to know which values are possible for the respective type of property and therefore also which values can be contained in the knowledge base. While the filter-mechanism is useful for technical experts, inexperienced users are easily overwhelmed by the complex input elements. Admittedly you can model and approximately represent complex facts with such user interfaces, but the success of internet search engines like, e.g., Google or Bing has created a different level of scale regarding interactivity and usability concerning the input masks within the domain of information retrieval. This is why we implemented full-text search based on the popular Lucene [2] library, which allows a more intuitive interface. This requires the relationally structured data of the knowledge base to be transfered into a representation comprehensible by the search index. The indexing of the documents results in a very performant search, whose findability can even be increased using techniques from text mining such as tokenization, stemming, stopword removal, n-grams or synonymous expansion. Due to the shallow structure of the documents however, it is difficult to depict the ontological relationships, which can be found implicitly within the entities of *MinaBASE*. An example for this is the deduction of knowledge from a taxonomical classification of objects which are difficult to translate into the shallow structure of documents. This implicit knowledge is therefore not available during the search process, which results in a low precision and recall of the full-text search component. It is admittedly possible to complete this information by using hooks in the request handling of Lucene and performing repective database queries to augment the information of the search index, but then you nevertheless lose the performance advantage of optimized index structures. For this reason, there is a need for a different approach, which combines the ease-of-use of the full-text-search interface with the high precision and recall of the content-aware expert search. In this paper we'll present a variant of the search-component which is a combination of the previously described approaches of the current state of *MinaBASE*. The use of a central input box and the use of an intelligent mechanism which completes the field automatically for the support of the taxonomy relation of the technical aspects as well as their parameters are important issues.

 ISBN: 978-1-61208-181-6

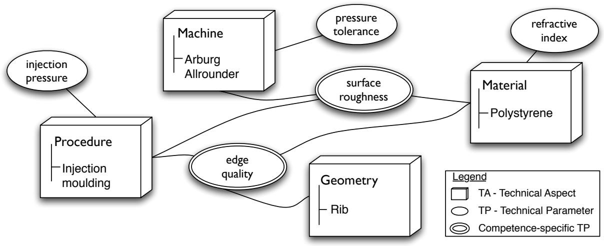

Figure 1. Schematic representation of a *MinaBASE* competence

The paper will be structured as follows: The next section will present the underlying process knowledge database *MinaBASE*. Then, a use case will be described that shows how the application-oriented search queries can look like. A solution approach will be described in Section IV, which will be evaluated briefly in Section V. The paper closes with a comparison of our approach with related work in Section VI and conclusions.

II. *MinaBASE* Process Knowledge Database

The micro system technology is considered to be one of the key technologies of the 21st century. It deals with micro systems, which means an intelligent, miniaturised system of sensors, data processing and/or actuators. Their product development is complex due to their small size, the free choice of materials as well as process-, design-, and manufacturing combinations. A specific manufacturing process is often required for each specific product, which results in low standardization of technologies. New challenges in the field of knowledge management are grown out of this, as there is a need to model manufacturing processes independent of specific products to facilitate product development by making it easier to retrieve universally valid knowledge about a manufacturing competence. An example for suchlike competences is the “injection moulding of PMMA on the Arburg Allrounder machine”, which is illustrated in Figure 1. The applied manufacturing procedure (injection moulding), the material (PMMA) and the machine (Arburg Allrounder) are central concepts of this competence. The so-called technical aspects (TA) that serve to model these materials, machines, and fabrication technologies are the smallest information entity in *MinaBASE* [3]. TA are arranged in taxonomies using generalization hierarchies. The number and contents of taxonomy trees can be specified and modified during runtime, such that a flexible structure tailored to microsystems technology can be defined. TA can be assigned properties that are referred to as technical parameters (TP). A TP is specified as a character string, integer, or floating-point number and references an attribute, e.g., density. As in the object-oriented approach, the TP of a TA are passed on to partial hierarchies located below in the taxonomy. In addition, lower hierarchy levels can further refine the inherited TP by specifying general value ranges. As a set of various TA from disjunct taxonomies, competences declare other TP, such as the edge quality and surface roughness [4].

III. Use case

After explaining the functional background, the structure of the underlying knowledge base we’ll now have a look at a use case, which shall demonstrate how a typical query can look like. For the comprehension of this use case it is important to know that the query doesn’t concern the details of the manufacturing competences from the technical point of view, but it is rather seen from an application-perspective. A first example for this is the use case ”cutting out a rectangle of a very thin, transparent, biocompatible plastic film” in context with *MinaBASE*, which is illustrated in the following Figure 2.

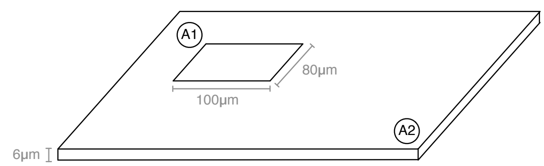

Figure 2. Structuring of a thin film polymer

It shows the physical properties of the desired application schematically. A rectangle of 100 μm width and 80 μm length shall be cut into a 6 μm thin, transparent and biocompatible plastic film. In this use case the rectangle, that is to be achieved, as well as the plastic film can be identified as the central input for the search, which can be conferred to the concepts geometry and materials, modeled in the aspect-taxonomies. The result of the search are those manufacturing competences, which are able to bring such forms in suchlike plastic films. The solution space can be narrowed down further by specifying more precise boundary conditions. One of these limiting conditions might be for example the exactness of the rectangle (quality of the edges or the precision of the sides angles). The search-component must be able to deduce these most relevant additional attributes appropriate to the use case by semantic interpretation of query terms and consultation of the knowledge base as well as offering corresponding context sensitive input mechanisms. Additional attributes for these use case are the dimensions of the rectangle (length and width) and also its precision. Due to this information you can make conclusions about convenient manufacturing procedures, which allow the creation of rectangles in a desired shape, like for example ”laser cutting”, ”laser milling” or ”precision milling”.

Furthermore the defined material qualities transparency and biocompatibility of the thin plastic form the starting basis for narrowing down the solution space to certain polymers, like for example polyurethanes (PU), polyethylene

 ISBN: 978-1-61208-181-6

(PE) and also polycarbonates (PC). With the aid of these basic information and the relationships which can be derived from the aspect-taxonomies (specialization of plastics, compliance of the additional attributes) a solution space of manufacturing competences can be constructed as a result of the query. Hereto the associated technical aspects need to be evaluated in order to find those competences, which describe the application of one of the mentioned process on the materials with the desired geometry and which correspond to the modeled additional attributes. In this manner an applied enquiry with qualities of the manufacturing competences can be seen as a solution for the use case. The actual possibilities regarding the query processing will be described in more detail within the next section.

IV. Concept

Based on the previous use case, this section will explain the request handling for the semantic search and distingush cases of possible inputs. Figure 3 displays a mockup of user interface prototype. The requests will be initiated by entries into a single input box, which leads to a semantic interpretation of the query terms according to the concepts that are modelled in the knowledge base.

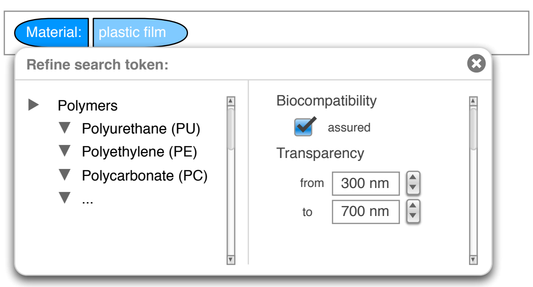

Figure 3. Semantic interpretation of keyword-based typed search tokens

As described, the central input box consists of a set of typed search tokens to which the system makes meaningful associated suggestions. The aim of this process is a better refinement of the search. The amount of such tokens can be seen as a set of restrictions, which need to be fulfilled by the competences which shall appear in the search result. Analogous to the described use case there is the plastic film as a base material filled into the input box in the Figure 3. Using synonyms of the material taxonomies, the engine suggests polymers and certain subtypes thereof. Additionally the engine finds most relevant parameters that are valid for the different taxonomy nodes like biocompatibility and transparency. After the user is done specifying the base material, he could for example enter the term ”rectangle”, which is the desired structure of the use case. In this case the engine would detect a node from the geometry taxonomy and load its most relevant parameters width and length as well as edge rounding and sidewall angle. While the user defines these restrictions, the result set of matching manufacturing competences, which are to be displayed underneath the suggestion box, is updated in real time in the background. This will enable an instant feedback mechanism enabling users to make incremental changes to their restrictions and see how these affect the result set. The rest of this chapter will focus on possible input values for the search-component and describe the specific cases in more detail. Figure 4 shows the possible interpretion of the given input. As you can see, one or more technical parameters (TA) can be specified. Additionally, the TAs can be described more detailed by adding constraints in form of technical parameters (TP) and even concrete values or value ranges for these TPs.

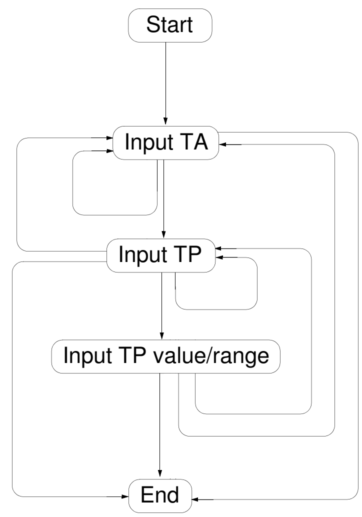

Figure 4. Input semantic for the keyword-based search

Case 1: Specification of a technical aspect

After specifying a technical aspect (for example a material), you can differentiate between the two following procedures:

(i) Overview on all technical aspects, which are below the selected node in the taxonomy: This makes sense, if you want for example to have an overview on the available technical aspects like the populated materials, processes, machines, etc. In this case the user interface offers all entries of the corresponding subtree. After the term is entered completely and recognized as a technical parameter these items are shown automatically by the suggestion-component and therefore can be selected directly. Afterwards the selected value overwrites the value that was entered before and represents a specialization of it.

 ISBN: 978-1-61208-181-6

(ii) Overview on the available technical parameters for this aspect: Every technical aspect is linked to one ore more technical parameters, which describe it more precisely. Furthermore these technical parameters are inherited from the root of the taxonomy to the leaf nodes.

To have an overview of the available technical parameters, all possible parameters for these technical aspect are shown in the input field after entering the technical aspect and the -tp (**t**echnical **p**arameter) flag. It contains the directly associated parameters as well as the parameters inherited from the upper levels of the taxonomy. By selecting one or more of these parameters and specifying a value or a range (see case 3), the search area within the taxonomy gets smaller as fewer nodes will fulfill the specified parameter value. Alternatively, instead of using the -tp flag, these parameters can be shown by focusing one of the entries in the subtree of the former case.

(iii) Overview of all technical parameters for this aspect and their specialization: If not only the parameters for the current technical aspect shall be shown, but also those parameters, which are defined below this aspect within the taxonomy, then this can be achieved by appending the flag -atp (**a**ll **t**echnical **p**arameters) after the technical aspect. If one of those parameters is selected, the origin of this parameter, which is a technical aspect that is located below the previous aspect in the taxonomy, is automatically selected and the previously typed or selected technical aspect is overwritten. This means that the selection of a specialized parameter will instantly lead to the selection of the technical aspect, that defines this parameter. Hereby a specialization of the previously entered term is achieved.

Case 2: Specification of a technical aspect and the corresponding parameter and items

After entering a technical aspect and a corresponding technical parameter, a concrete value or a range can be set for this parameter. If necessary the prior selected technical aspect gets replaced by an aspect which is in a lower part of the taxonomy, but only if it is required by the selected value or range of the parameter.

Case 3: Specification of more technical aspects

In this case the entry of a technical aspect (optionally with parameter and value/range) is followed by the entry of further technical aspects. Starting with the second technical aspect, the previously entered aspects functions as an additional filter. Therefore only those technical aspects are presented in the autocomplete box, for which there are competences, which also include the previously entered technical aspects.

Case 4: full-text search:

In case of not finding any terms within the concepts of the technical aspects, the technical parameters or the competences, the search-component switches automatically to the full-text search and then it tries to find hits within the foregoing concepts.

V. Evaluation

To evaluate our concept, we build a simple hardcoded version of our semantic fulltext flavored search. The first tests are encouraging, the search is much more precise as the previously implemented fulltext search. Additionally, after a short learning phase the time to retrieve the relevant datasets was significant shorter than using our old expert search.

VI. Related Work

The idea to bridge the gap between user friendly keyword-based search and expressive, formal and structured queries has fostered a lot of research in the past. The approaches that are closest to our paper are the following: SQAK [5] attempts to make it easier to access structured information stored in databases by using a keyword-based approach. Users require knowledge about SQL as well as a deep understanding about the structure of the underlying database for non-trivial use cases. SQAK attempts to overcome this situation by diagnosing the structure of the database and offering a keyword-based input mechanism requiring only small amount of knowledge about the structure of the database, that is translated into more complex SQL queries. DBXplorer [6] allows to search across relational databases using keywords by utilizing so called symbol tables that contain meta information about the database. Similiar to inverted document lists in information retrieval these symbol tables allow to match incoming keywords to rows and columns of the database, while also supporting queries spanning across multiple tables using dynamically built join trees. As we have only a limited set of tables (technical aspects and parameters) as starting points for our semantic search, the use of symbol tables is not adequate for our scenario. Our main focus is semantically evaluating the incoming search terms and suggesting, e.g., most relevant technical parameters or subconcepts for better refinement of the search restrictions. GINO [7] and Ginseng [8] provide a natural language interface for entering queries in a quasi-English language with guided entry of ontological concepts as well as derivation of triple-sets and SPARQL queries for query processing. While the concept of guided entry is similar to our automatic suggestion of parameters and aspects stemming from the knowledge base, we dont consider natural language interface suitable for our domain, since numeric input of parameter values and ranges is simpler to capture within a suggestion box since they are associated directly with the concept they are linked with in the knowledge base. XXploreKnow! [9], Q2Semantic [10] and SPARK [11] are focused on translating keyword queries into formal description logic queries. The approach of mapping search terms to knowledge base entities, exploring the connections between

 ISBN: 978-1-61208-181-6

them and utilizing the acquired knowledge thereof for further refinement of the search is similar to our approach. The difference is, the structure of their knowledge base is built on tools stemming from the field of the semantic web technologies, while *MinaBASE* uses a traditional relational database with a flexible entity-attribute-value schema.

VII. Conclusion and Future Work

This paper presented an approach to extending the *MinaBASE* process knowledge database, a system for managing the knowledge in the field of microsystems technology. By means of this approach, a semantic search can be implemented, that maps application-oriented properties to concepts of the knowledge base, which allows easier access and a more intuitive discoverability of knowledge entities. After the concepts of *MinaBASE* were explained in more detail in Section II, a concrete use case for the semantic search was shown in Section III. Afterwards we presented our concept in Section IV by first describing a mockup of the user interface according to the use case and then explaining the different input cases that need to be distinguished as part of query processing. In a future version, we plan to implement a customizable version of our semantic search. In this implementation the possible states, transitions, actions and preconditions associated with the transitions will be configurable. If this version will be implemented as a interpreter or with the help of a software generator is still an open question.

References

[1] M. Dickerhof, “Prozesswissensmanagement für die Mikrosystemtechnik.” *Statusseminar MikroWebFab, Karlsruhe*, 2003.

[2] E. Hatcher and O. Gospodnetic, *Lucene in Action (In Action series)*. Greenwich, CT, USA: Manning Publications Co., 2004.

[3] M. Dickerhof and A. Parusel, “Bridging the Gap—from Process Related Documentation to an Integrated Process and Application Knowledge Management in Micro Systems Technology,” *Micro-Assembly Technologies and Applications*, pp. 109–119, 2010.

[4] M. Dickerhof, O. Kusche, D. Kimmig, and A. Schmidt, “An ontology-based approach to supporting development and production of microsystems,” *Proc. of the 4th Internat. Conf. on Web Information Systems and Technologies*, 2008.

[5] S. Tata and G. M. Lohman, “SQAK: doing more with keywords,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’08. New York, NY, USA: ACM, 2008, pp. 889–902.

[6] “DBXplorer: A System for Keyword-Based Search over Relational Databases,” in *Proceedings of the 18th International Conference on Data Engineering*, ser. ICDE ’02. Washington, DC, USA: IEEE Computer Society, 2002.

[7] A. Bernstein and E. Kaufmann, “GINO - A Guided Input Natural Language Ontology Editor,” in *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, Athens, Georgia (US), 2006.

[8] E. Kaufmann and A. Bernstein, “How useful are natural language interfaces to the semantic web for casual end-users?” in *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ser. ISWC’07/ASWC’07, 2007, pp. 281–294.

[9] T. Tran, P. Cimiano, S. Rudolph, and R. Studer, “Ontology-based interpretation of keywords for semantic search,” in *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ser. ISWC’07/ASWC’07, 2007, pp. 523–536.

[10] H. Wang, K. Zhang, Q. Liu, T. Tran, and Y. Yu, “Q2Semantic: A Lightweight Keyword Interface to Semantic Search,” in *In Proceedings of the 5th International Semantic Web Conference (ESWC’08*, 2008.

[11] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu, “SPARK: adapting keyword query to semantic search,” in *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ser. ISWC’07/ASWC’07, 2007, pp. 694–707.