Integration possibilities of a context-based search engine into a project planning portal in the mechanical engineering domain

Raiko Eckstein, Andreas Henrich
University of Bamberg
Chair of Media Informatics
D-96052 Bamberg, Germany
{raiko.eckstein, andreas.henrich}@wiai.uni-bamberg.de

Abstract

Product development processes in the mechanical engineering field are getting more and more complex while the requirements of cost and time reduction increase. Thus, there is an increasing necessity to support engineering designers by doing their everyday work. The paper has the aim to show a way to integrate a framework for context-sensitive information retrieval into a project planning portal. This kind of software targets the successful execution of processes in enterprises and facilitates cooperation between the involved project teams. The included search functionality often is insufficient for the companies goals: promotion of reuse of existing parts and information like best practice and lessons learned documents. The proposed extended search functionality includes context information about both the engineering designer himself and the documents to anticipate the user's information need and thus improve the quality of the retrieval results. Amongst others, especially process and workflow information is gathered to supply adequate information for the current working task of the engineering designer.

1 Introduction

In the field of engineering design projects are nowadays mostly done in a distributed way spread over several teams and places. Additionally, several domain experts are included in these projects: mechanical engineers, electrical engineers, mechatronic engineers and computer scientists. All these experts fulfill certain roles in the projects and are responsible for parts of the final result. To manage the interaction between the different parties usually some kind of electronical project planning takes place. Ranging from simple excel sheets up to full blown project planning portals these solutions have in common that they aim at supporting and ensuring the success of the project.

To leverage the knowledge in the company it is necessary to increase the reuse of results of previous projects. In contrast to a simple query-based search, a context-sensitive search engine can return more relevant search results according to the user's current information need. There exists a large variety of potential influencing factors which affect the anticipated search results starting from personal domain knowledge, current state of the project, available product models to company guidelines concerning construction, e.g. for specific design objectives such as low-cost production.

Our intended search engine exceeds the usual extent of document coverage of a common search engine. In the domain jargon the interesting documents are subsumed as *product models* which describe all emerging representations of the final product. That includes paper-based, digital and physical product models [Lauer *et al.*, 2007]. Arguably, it is difficult to return a tangible model in a digital tool, but often the existence and creation of physical models is noted somehow and therefore a pointer to the real model can be returned.

It is intended to cover all evolving product models during the whole development process starting from initial product specifications from the customer to the final product documentation. In between lies a vast amount of different document types which are needed to represent a product during construction. This starts from text-based requirement specifications, cost calculations, feasibility studies and time schedules. In later phases (design and elaboration phase) the description of the product models changes to CAE (Computer Aided Engineering) documents, such as 3D models, technical drawings, or simulations and their results. As a first (incomplete) overview the document types can be subsumed as structured and unstructured text documents, 2D and 3D models as well as simulation data.

The diversity of artifacts requires a differentiated view of the documents. They usually contain several features which have to be considered during similarity search. A CAD (Computer Aided Design) drawing not only contains geometry information but also includes material data and topology information about assemblies, parts and subparts. Due to those different features the necessity of different index structures and combining algorithms arises.

Furthermore, our context-sensitive search framework incorporates metadata gathered from documents and context information to describe a document and its function in a process more precisely. These different views at a document enable a multi-criteria search and ranking which is illustrated in figure 1. It visualizes the change in the information need of the user during the proceeding of the process as well as the different weightings of the features depending on the process progress. The different aspects of each information need and each artifact are represented in separate ways. Thereafter, a weighted combination of the similarities/relevancies of the different aspects is created applying the metrics $m_{i,r,f}$ and weights $w_{i,r,f}$ selected for the information need i, the representation r and the feature f.

For the document-specific features extractors were implemented for common document types and exchange formats typical in the domain under research, e.g. STEP (Standard for the Exchange of Product Model Data) and DXF (Drawing Interchange Format). To gather more metadata

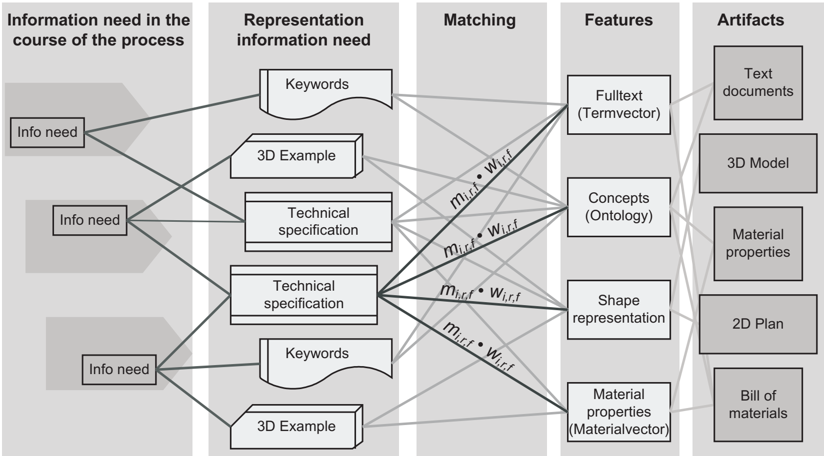

Figure 1: Example of a multi-criteria search

which goes beyond the creator and the creation date, we employ a process management portal which already delivers a huge amount of necessary metadata derived from a process metamodel. That allows the classification of documents into a process, the assignment of documents to different roles etc. which is described further in section 3. For user queries it is now possible to include additional context information which as well can be derived from the project planning portal. The paper therefore outlines the steps to integrate a context-sensitive search engine into a project planning portal and points out which contextual factors can be extracted and which further information would provide additional benefit.

The paper is organized as follows. In section 2 related work for the field of context-sensitive information retrieval is discussed. Section 3 deals with the information content of a project planning software and outlines the contextual factors an integrated search engine can use. Section 4 discusses which additional information would be helpful in searching but is not yet provided. Section 5 shows which steps were taken for the integration of the context-sensitive search engine into the project planning portal. Section 6 introduces the possibilities and problems of user interface design concerning contextual search and search result presentation in terms of being non-intrusive. Finally the paper outlines encountered problems and solutions and provides a conclusion and a view into the future work of the project.

2 Related Work

The aim of our IR (Information Retrieval) system is a goal-oriented provision of information in the product development process by providing reusable artifacts and considering context information. Thereby an improvement of the development process and a lowering of construction time and cost can be achieved.

The *principle of polyrepresentation* [Ingwersen, 1994] from the field of IS&R (Information Seeking and Retrieval) suggests that the information need of a person should be represented by a vast variety of influencing "features" which have an impact on the user query. The same applies to the documents in the index, i.e. their representation is not feature-complete when just the actual contents are taken into account. Therefore, an advanced search engine needs to combine different aspects. For example, a 3D similarity comparison should include information about geometry, topology, material properties, and metadata of an artifact to improve the retrieval results.

Moreover, the inclusion of contextual knowledge will have an important impact on search results [Allan, 2003]. Ingwersen's *label effect* [1982] stated that a user is not expressing the whole information he has about his information need, but only that amount he thinks is *enough* for a human recipient [Ingwersen and Järvelin, 2005]. Therefore, an augmentation of contextual knowledge to the expressed query is necessary.

At this point, the concept of context has to be divided into two aspects: the context of the indexed documents and the context of the person starting a search. According to Dey and Abowd [2000], we understand the latter one as the accumulation of all factors which have an impact on the information need of an engineering designer during his work. The document's context is e.g. characterized by its classification into the process and the role it plays in the process's lifetime.

Furthermore, a search engine can support two opposed ways of query issuing. The classic way consists of querying the search functionality on demand. That means the user asks for search results when needing them. A context-sensitive search engine then may augment the query with collected context information of the user to narrow the search space or state the query more precisely.

The contrary way produces search results proactively when the search engine notices that the user has an information need. The query is assembled through the different sensed contextual inputs which form the information need.

In an enterprise it is a common use case for a purchaser to search for off-the-shelf parts. If it would be possible to find similar parts of an assembly which are already produced or purchased in the company it would be beneficial to use these parts instead of redesigning them from scratch. Such a search will occur quite frequently and will often be queried actively by the purchaser.

In contrast, many tasks of an engineering designer are highly creative and therefore the worker does not want to get interrupted by unnecessary search activities. But during these designing tasks the search engine can proactively analyze the user's context as well as the viewed and edited files. Through inference the search engine can deduce the user's information need. That enables the retrieval of parts in the company's part repository which fulfill the same function and fit in the current project assembly. If the retrieval is successful it not only saves money for the company since the part does not have to be designed and maintained for the whole product lifecycle, but also the different testing tasks for the part can be skipped which is a huge timesaver.

In conclusion, our IR-system should deliver different kinds of artifacts relevant for a designer's information need in a specific process phase.

One of the first references of context-aware computing in literature came from Schilit [1994] where a ubiquitous computing system was introduced which was able to react to different signals of a wireless, palm-sized computer. The field of context-sensitive or context-based information retrieval was then broadened to a wide variety of fields of application.

With the development of the World Wide Web several recommendation systems for web pages were proposed. Examples are Lieberman's Letizia [1997] and Joachim's Web Watcher [1997]. Horvitz et al. [1998] developed a prototype for a context-sensitive help system for Microsoft Office which determined the user's context from several influencing factors which were input for a bayesian network.

The COBAIR project aimed at supporting computer scientists in the software engineering domain [Morgenroth, 2006; Henrich and Morgenroth, 2003]. According to the interactions of a user with the integrated development environment (IDE) the search engine inferred the user's information need and returned artifacts the user might need, e.g. software classes the user could reuse.

The DFKI[1] project KnowMore tried to support users working in a knowledge-intensive workflow. Proactively, i.e. without user action, the system provides information the user needs for his current task. Abecker et al. [2000] modeled the information needs in several ontologies which were subsumed as an *organizational memory information system*. An ontology-based heuristic retrieval method is used to identify concepts of the modeled ontologies and to deliver the needed information.

Our context-sensitive information retrieval framework intends to support the search for all emerging product models during the development process. A system which has the same intentions is the *Design Navigator System* introduced in [Karnik *et al.*, 2005]. It targets the management of design information. The authors distinguish between information related to the design and information related to the design's evolution and history. That leads to six different types of information: requirements and specifications, functional decomposition, assembly structure, part geometry, annotations, and interconnections. The main goal is to consider all information during the design process which includes design rationales that can be reused in other projects by means of best practices. The system allows various access methods. The search functionality supports browsing, design rationale search and geometry-based search amongst others. However, context information is not incorporated.

3 Information content provided by a project planning portal

This section deals with the information contained in a project planning portal and therefore the potential input data for a context-sensitive search engine. Due to a tight integration into this kind of software it is ensured that much information concerning the user as well as the administrated documents can be gathered for querying. The challenge of the context-sensitive retrieval model is to filter and weight those following criteria which are influencing a document's as well as the user's context. Figure 2 provides an overview of the functions and submodules of a project planning software and its interlinkage with the search framework.

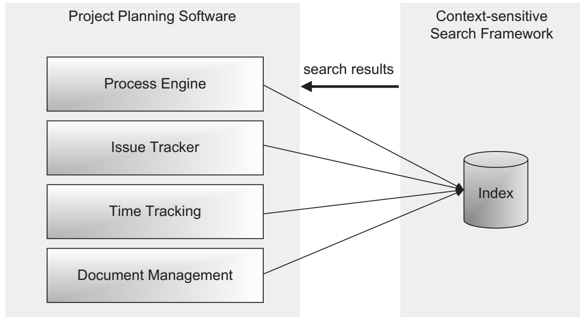

Figure 2: The tasks of a project planning software

3.1 Process Model

Each project usually follows a certain process model which describes the sequence of phases and activities, the documents that can/must be created therein, the responsible roles and the applicable methods. Project planning software usually allows the modeling of process models to fit the company's infrastructure. This modeling includes a partial description of the user's information need as documents are assigned to phases or tasks and accompanying roles. Thus documents might be entry or exit criteria for a phase, they might be revised in a phase or just needed for reference like company guidelines. This knowledge provides a much deeper understanding of a document and its purpose in a process for a search engine.

The classification of documents into a process model is of high significance for a context-sensitive search engine as the user's information need has a strong relation to the process phase the user is currently working in (cf. figure 1). Whereas during early phases more general documents are needed, later phases require more specific documents. This is motivated by the creative process in the beginning where potential solutions are evaluated. When the chosen solution is going to be constructed the designing engineer needs specific documents and partial solutions which meet the defined requirements.

Considering the filetypes it shows that each phase requires specific documents which should be taken into account when ranking documents. In the planning phase the dominating product models are text-based while in the design phase specific CAE models (2D/3D models, simulations etc.) are generated.

By knowing the affiliation of a document to a task or process phase a search engine can then deliver search results which help the user in his current task. This connection can be inferred through the assignment of roles to users which denote the tasks the user is responsible for. This permits the search engine to better infer the user's information need. Additionally, the interactions of the user with the project planning tool are analyzed and taken as input. If the task demands the existence of a finished document the search

[1] Deutsches Forschungszentrum für Künstliche Intelligenz (German Research Center for Artificial Intelligence)

engine can look for similar entry documents and then deliver the inferred result documents from past projects.

3.2 Time Tracking

When a project is executed the process model is instantiated and time, monetary, and human resources are allocated and assigned to the different parts of the project. To allow a successful completion of the project in a timely manner time restrictions for the phases and work tasks have to be set. A project planning software can assure that deadlines are followed in enabling notification mechanisms if a due date is approaching. Additionally, project managers can get a quick overview of the project's progress. Time tracking and the (financial) acquisition of amounts of work are additional factors a context-sensitive search engine can use in ranking a document.

If it is known how much time a work package will approximately take a search engine can upvalue documents from other projects with similar prerequisites to allow skipping this phase or at least to diminish the expenses. For example simulations can take a long time to complete, especially acoustic simulations, which makes it impossible to carry out too many change and test cycles. If simulation results are presented to the engineering designer he may postpone the actual simulation which is still needed but does not have to be executed unnecessarily.

Monetary values of documents can have an influence on the ranking as well. If the user is currently creating a document of high value – because of the required manual effort or the costly simulations, prototypes etc. – it seems to be beneficial to present similar or related documents as early as possible in order to boost reuse and the exploitation of analogies. To this end, the "value" of a document as well as its fit for the current situation of the user should be considered by the search system.

A context-sensitive search algorithm can take advantage of these factors when ranking documents because the search query comprises a more precise description of the user's information need. That leads to more precise answers and helps the user to deal with the information overflow.

3.3 Issue Tracking

In the software engineering domain issue tracking systems like JIRA[2] and Bugzilla[3] are widely used to manage the release process and the resolution of bugs. In product development small changes in parts often have impacts on other parts or modules which have to be resolved as well. A project planning software can support the assignment of those changes to a certain user. By inferring the next and time-pending tasks the context-sensitive search engine can recommend solutions for the raised issue querying similar projects. Additionally, it may provide and include *best practice* and *lessons learned* documents to prevent the user from following the wrong solutions and from repeating done mistakes in previous projects. The availability of further information enables the search engine to augment the proactive query with the task context which is derived from the issue tracking system.

3.4 Document Management

The comparison of several companies in the product development domain showed a variety of ways how documents produced during process execution are stored. This starts from storing files on a network share which is a semantically – at least for an indexing module – weak representation of the files. Storing documents in a PDM (Product Data Model) system denotes the other extreme as the user is forced to annotate files with metadata. Furthermore, those systems include workflow components which allow the classification into the process. The administration and storing of versions and variants supports configuration management.

A project planning software can pursue three different approaches. The simplest way is to just save a link to a file which was created during the process with all the arising problems, e.g. stale links due to the moving of files. If the system offers an integrated DMS (Document Management System) all the needed data (document and associated metadata) can be retained together and later used for an enhanced similarity search. The drawback of this approach is the potential redundancy as the files usually are as well stored in PDM systems. They allow a tighter integration with the used application systems in the company. Especially CAE systems often offer interfaces which enable a simple file exchange with a PDM system.

Our approach assumes that a direct connection between the project planning software and the DMS yield the best results. This allows a reliable way to augment documents with metadata. The emerging documents are *checked into* the project planning portal which relays them with added contextual information to the DMS. If an existing document is accessed the portal retrieves the document from the DMS, handles locking and makes the document available to the user. This final approach assumes the existence of a pluggable interface which enables the integration. Hereby no superfluous redundancy will occur and the document's context can be accessed from current data.

3.5 User Profile

Often it is necessary to store additional data for and about the user, e.g. his expert-level. This information often is used as a user interface filter which enables or disables several options according to the user's experience. Such a user profile will contain several types of information about the user. The user can provide information about his fields of interest, about expert knowledge, his former projects, his current roles, etc. A conceivable enhancement is the integration of skill databases which provide information about experts in an enterprise who can help or deliver information for specific tasks. A context-sensitive search engine will take this context information into account when handling a query to improve the description of the user's information need.

Search engine results frequently include information a user already knows and therefore clutter the result pages which makes the target-oriented information acquisition difficult. The consideration of contextual information can act as a filter which fades out the already known and therefore irrelevant result items.

3.6 Search functionality

Project planning tools support the retrieval of information stored in the program. But the supported extent of the search engine differs for the different tools on the market.

Usually, the integrated search supports the reactive approach which enables the user to query the search engine when he has an information need. The query is entered

[2]http://www.atlassian.com/software/jira/

[3]http://www.bugzilla.org/

into a textfield and the search results are presented in a one-dimensional list. Often the user has some filtering options like restricting the search to the current project or to include/exclude certain parts of the application, e.g. excluding search results from the issue tracker. The advanced search for special metadata is often implemented only rudimentarily which hinders the efficient retrieval of documents that would satisfy the information need of the user.

Often the search only includes the built-in descriptive pages which depict the documents and process parts (phases, roles and methods) but do not include the actual document content.

4 Desirable improvements for project planning software

As pointed out above a project planning portal enables document traceability which subsumes the localization of artifacts in a process. That information depicts a temporal relationship. If a causal relationship between parts of documents could be preserved in terms of requirements traceability, the search engine could support tasks where requirements change in later phases – which happens frequently – and present documents which were created because of the original requirement and now need a revision.

A similar improvement would be the storage and traceability of design rationales which denote the rational background why an artifact is designed the way it is [Lee and Lai, 1991]. This information is valuable in several situations. If design changes are necessary in later phases mutual interactions between part functions are better visible and explained by design rationales. That helps avoiding modifications which are impractical or impossible in regard to other product functionality. Furthermore, this information encapsulates design knowledge which should be reused in other projects to benefit from past experiences. Studies concerning the collection of design rationales can be found in [Regli *et al.*, 2000] and [Nomaguchi *et al.*, 2004].

More precise contextual information about specific artifacts like parts or assemblies could be derived if additional systems would be connected with the project planning portal or with the context-sensitive search framework. An ERP (Enterprise Resource Planning) system can deliver information about order and lot sizes as well as manufacturing costs which can play a role in document ranking. That pays off if parts which are already purchased in high quantities can be incorporated in new assemblies which are under construction.

In the domain of mechanical engineering several measures exist which describe the progress of a process. The *degree of maturity* can be derived in various ways which include the ratio of finished tasks to the number of all tasks, the ratio of finished requirements to all requirements amongst others [Pfeifer-Silberbach, 2004]. That information abstracts from the measurement of the process progress in terms of finished tasks or documents and focuses on a more product-centric degree of progress.

5 Steps taken for Integration

This section deals with the steps that were taken to integrate a context-sensitive search engine framework into a full-blown project planning portal software. Hereby we use the web-based process management portal *project kit* which was provided by our industrial project partner *method park Software AG*[4].

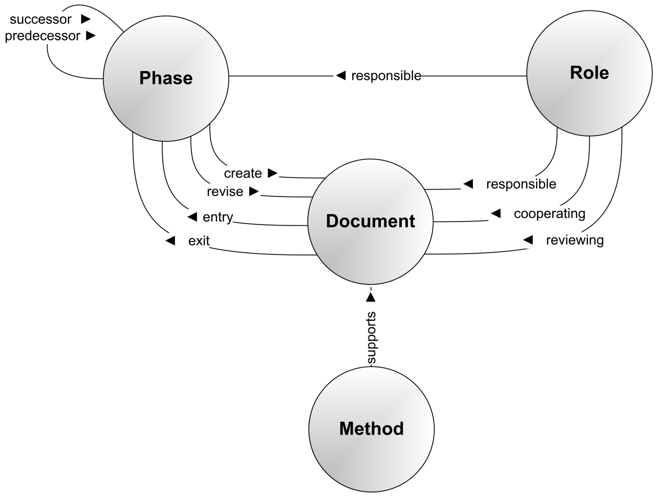

Figure 3: Document-oriented model process

5.1 Integration of a model process

As this research takes place in the domain of mechanical engineering a model process is needed to reveal domain-specific problems. A process was modeled according to the VDI[5] Guidelines 2221, 2222 (part 1) and 2223 [VDI, 2006]. This process consists of four main phases: planning, conception, design and elaboration. Our search engine supports the designing engineer during construction. Therefore, we omit the additional covered phases before start of production which are needed in process planning but do not yield an extra added value for our target group.

Due to the document-oriented process approach mainly found in the addressed domain documents which represent the product models are the driving parts in the process. Documents are created in certain phases. The user utilizes methods to produce the required product models. Special states can be assigned to documents, e.g. a document might represent a *milestone* which denotes the final outcome of a phase and therefore has to be created.

To approximate a more real-world model taking into account norms and practical experiences the phases were further divided which resulted in 15 phases with attached entry and exit criteria. This step was done with the help of two automotive suppliers and other research groups from the academic mechanical engineering environment. Before finally deploying the project planning software the process has to be tailored to meet the company's requirements.

As the project planning software includes a definable process metamodel the modeling of all kinds of processes is supported. The used process metamodel is displayed in figure 3. It defines four major process elements. Due to the document-oriented nature the *document* is the center of those elements. Documents are created or revised in a *phase* and can be entry or exit criteria for a phase, i.e. a phase cannot be finished before all necessary documents are delivered. The same applies to the preconditions. A phase cannot be started if a needed document has not been finished yet. *Roles* which are assigned to users are responsible for documents. Additionally a role can be cooperating

[4]http://www.methodpark.de

[5]Verein Deutscher Ingenieure (Association of German Engineers) – http://www.vdi.de

in creating a document or do the review process. This review process depends on the modeled document lifecycle which describes the different states an artifact can/must undergo. A change in a lifecycle state can as well be restricted to a certain role for coordination processes concerning the documents. For simplicity the document lifecycle was restricted to four stages: not existent, draft, review and released. Real-world company document lifecycles usually are more detailed.

Finally *methods* are assigned to documents which describe possible ways to create a document. For example after the function structure of a product is defined the possible solution principles have to be determined. One method to support creativity is the morphological box which collects possible technical solutions for sub-functions.

5.2 Identification of hooks for retrieving context

Our context-sensitive search engine can handle multiple contextual factors. To gather this information hooks into the project planning software have to be provided. The context information has to be distinguished into the user and the document context which both have to be determined.

The context-sensitive indexing module must be notified if documents are changed in the project planning software, i.e. when new documents are checked in, when old documents are changed or deleted. If new versions are created this relationship has to be preserved, so that it is possible to revert a document later and the index does not contain stale data.

The indexing step is the crucial part to derive additional data about the documents in the project planning software. Due to the integration of a process model (cf. section 3.1) much more information can be collected which describes a document and its function in the process further than a simple filescan of a network share. In addition the user can be forced to enter additional meta information, e.g. design rationales. With the interlinkage between the issue tracking system (cf. section 3.3) the tracking of design or bug changes appears possible.

The same applies to the influencing factors of a user's context. Although the user's context not just stems from his interaction with the project planning software, a quite reliable assessment of the state in the process he is in can be derived from the information in which projects he is active and to which tasks he is assigned. The history of projects and roles the user worked on leads to a promising assessment of the user's knowledge, special capabilities and technical strengths.

5.3 Integration Architecture

Figure 4 shows the architecture of the integration of our context-sensitive search framework into the web-based project planning portal solution is based on. To minimize the changes in *project kit* the interface between the two systems is established through aspect oriented programming [AOSD, 2007] – more precisely AspectJ [AspectJ, 2007]. All document changes are monitored using aspects which notify the indexing component to update the index. The same applies to the contextual information which is gathered using different aspects.

As can be seen in the figure 4 *context extractors* extract the document's context and transform it into features in the *feature generator*. The query module is split up into two parts. Reactive user queries are received and automatically augmented by contextual data of the user in the *context augmenter* module. If the search engine works proactively the *proactive query generator* constructs the query. Both types of queries are then passed to the *multi criteria matcher* which executes the multi-criteria search using several *metrics* and *weights* for each feature which are depending on the user's and document's context. The search results are finally passed back to the project planning portal which displays them to the user.

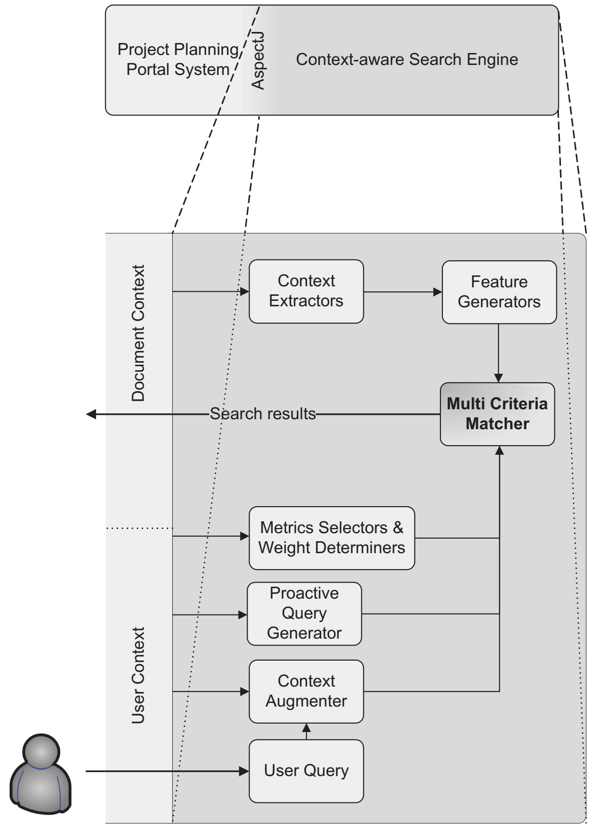

Figure 4: Simplified integration architecture for a context-sensitive search engine into a project planning portal

6 Thoughts on User Interface Design for a context-sensitive Search Engine

The integration of an advanced search engine into an existing application will affect its user interface. Our context-sensitive search framework will support two different ways of offering results. The "traditional" way – which usually is already included in the out-of-the-box search – is to enter the query in a search field. Even if the search engine augments this query with contextual information, the UI design does not have to change as the action appears underneath. The alternative way of issuing a query is the proactive search. Here the search engine tracks the user interaction, tries to infer on what tasks the user is currently working. In addition a synchronization is done with the user profile to see if the user needs help with the current task. If an information need is identified the search engine queries the index itself with a self-constructed query with contextual information. This query can be envisaged as the difference of the short term and the long term context of the user.

Now the presentation of this data is the crucial part, as any form of pop-up window might disturb the user and interrupt him from doing his work – which is disrupting especially in a highly creative task like construction in the mechanical engineering domain. Therefore, the user interface must allow a non-intrusive way of presenting the search results. Furthermore, the user must have the possibility to turn off this kind of search because he might feel patronized by the search engine.

If a web-based portal solution is used, a HTML frame is imaginable which is updated after a period of time or is notified if relevant search results are available. Through the adoption of AJAX [Garrett, 2005] this is easily implemented. A bit more involving but less intrusive way could be the use of some icons, e.g. an unlit bulb and a lit bulb as a notification that search results were found. With a mouse-click the user can display the results on demand.

A completely different approach for the representation of search results is the usage of a faceted search [Yee *et al.*, 2003]. Hereby the user browses search results and has the ability to filter documents by adding facets to the provided search results. This approach seems feasible for the outlined problem from section 1 as the documents in the engineering domain can be classified by several categories. The process-oriented approach (cf. section 3.1) already provides different process meta information. Faceted search can start on top of all documents contained in the index but also supports an initial search based on a user query. The user then may filter the results in adding a facet which describes documents from a certain phase, the document type, the project and so on. Therefore, the user can narrow down the search results to browse only the documents relevant to him.

For using a faceted search the indexing module has to determine the classifications for the facets. Although that introduces problems if only a plain document is available, the existence of the project planning tool which provides additional document information (context) as described in section 3 enables several initial facets. In addition the document extractors during the indexing step can try to infer more facets like part numbers and material codes.

Incorporating context information in a search engine has another pitfall which has to be considered carefully in user interface design. Common full-text search engines return small snippets of the search results which exemplify why they were delivered. The user is able to see the matches of his textual query and the results. If the context-sensitive search engine proactively queries the index or augments a user query that approach is not feasible. Hence, an explanation must be provided so that the user can understand why and how the search results were retrieved. This functionality is necessary to ensure a high user acceptance – if the user does not comprehend why the search results are returned he might reject the search engine.

7 Encountered Problems and Challenges

The encountered problems of the integration of an context-sensitive search engine into a project planning portal can be divided into two parts: adoption and technical problems.

7.1 Adoption Problems

A specialized search engine should strive for full coverage of documents that might be useful for the context it is working in. In the domain of mechanical engineering it should support the retrieval of all documents which are created during construction and are necessary in the process. This can be a drawback for a project planning tool if engineering designers do not document everything they produce but only final results and milestones. For example "analog" sketches might disappear in ring binders and some steps which could be supported electronically are done in the designer's mind and are not archived for later access. Experienced engineers often deduce the functional structure of an assembly and think of possible solution principles and pick the right one. An inexperienced worker could use the method of the morphological box which is a matrix representation of sub-functions and their possible technical solutions. After the creation the best fitting solution for each function is taken and is used as a preliminary design. The retrieval of this kind of document can be useful later if requirement changes occur and another solution for a sub-function has to be chosen.

Other industry partners in the FORFLOW project stated that similar systems were not fully accepted and adopted by the users. Only those documents were archived in a document management system (DMS) which are dictated by the system. This problem evolves if the user does not see an immediate revenue for his efforts in doing this "extra work". Hence, a combination of a system addressing the documents maintained in the DMS and a system indexing the project fileshare seems to be a promising approach.

Rights management and authorization are also problem areas which cannot be solved easily and are partially dependent on the company culture. In some companies there exists a high degree of competition between different departments, i.e. results are not simply shared and therefore are not accessible for other departments.

A search engine that uses context information could present documents from other or older projects which include the solution to a current problem but would not be allowed to present them because of secrecy issues. Since there exists no solution to this problem, it would be imaginable that the search engine does not return the actual document but a pointer to a responsible person or project team which has the rights on those documents so that the engineer can try to obtain access through the official channels. If even this "expert search result" is not allowed at all in the company the main goal of our search engine framework – the support of higher reuse of components and assemblies – cannot be achieved. Nonetheless, prototypes like design studies that are strictly confidential exist in companies which should be excluded from searches.

7.2 Technical Problems

Some technical problems occurred as well. The integration of a search engine into a finished product can introduce unwanted redundancy into the system. A project planning tool needs some sort of persistence layer which uses a certain persistence framework. The context search framework as well needs to store information. Both the context of the documents and the user have to be persisted. This creates duplicate information which not only takes unnecessary space but as well might introduce stale data if some information is not updated correctly. The external storage is required to allow a fast retrieval of documents by its contextual information.

The retrievable documents themselves are stored in an index which saves the document representations. This index does not use a relational database but some kind of index structure like an inverted file which allows faster retrieval of full-text contents. The built-in search of a project planning tool usually uses some indexes which after the integration are superfluous and should be removed because of the higher hard disk usage.

Process planning tools not only try to support the steering of the process but also try to recommend the next steps an engineer might take. Therefore, much process documentation is included which explains methods, documents and process phases. A search engine should deliver this information as well and needs a way to index that data properly, but in many cases the documentation is in more or less proprietary formats.

As these tools offer the customization of the process, hooks are needed which notify the indexing module of occurred changes so that an index update can be triggered.

8 Conclusion and future work

This paper showed the process of integrating a context-sensitive search engine framework into a project planning software which until now only had a simple full-text search. Therefor we proposed an integration architecture which keeps changes in the target platform – the project planning portal – to a minimum. We outlined the advantages of the integration of an advanced search engine which takes process information and user profiles into account to deliver situation specific search results. This leverages the retrieval of existing knowledge in the company and facilitates the reuse of components.

The next steps consist in further exploiting the information supplied from the project planning portal and the inclusion of external contextual information to describe the user and document context more precisely. This will include e-mails, network fileshares and other applications like MS Office and especially for this domain CAE software. Finally specific parameters from the engineering domain will be researched, e.g. the design situation the engineer is in, the stage of maturity of a product, its complexity and its purpose of use.

Acknowledgements

The work described in this paper is part of the Bavarian Research Cooperation FORFLOW consisting of six Bavarian universities working in the fields of engineering design and computer science. It collaborates with 21 companies and is promoted by the Bavarian Research Foundation (Bayerische Forschungsstiftung BFS).

References

[Abecker *et al.*, 2000] Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, Otto Kühn, and Michael Sintek. Context-aware, proactive delivery of task-specific information: The knowmore project. *Information Systems Frontiers*, 2(3/4):253–276, 2000.

[Allan, 2003] James Allan. Challenges in information retrieval and language modeling. *SIGIR Forum*, 37(1):31–47, 2003.

[AOSD, 2007] AOSD, 2007. `http://aosd.net/` - 2007-07-02.

[AspectJ, 2007] AspectJ, 2007. `http://www.eclipse.org/aspectj/` - 2007-07-02.

[Dey and Abowd, 2000] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Proc. of Conf. on Human Factors in Computing Systems (CHI 2000)*, Den Haag, Niederlande, 2000.

[Garrett, 2005] Jesse James Garrett, 2005. `http://www.adaptivepath.com/publications/essays/archives/000385.php` - 2007-07-02.

[Henrich and Morgenroth, 2003] Andreas Henrich and Karlheinz Morgenroth. Supporting collaborative software development by context-aware information retrieval facilities. In *Proc. of the 14th Int. Workshop on Database and Expert Systems Applications (DEXA)*, Prague, Czech Republic, September 2003.

[Horvitz *et al.*, 1998] Eric J. Horvitz, John S. Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, Wisconsin, USA, July 1998.

[Ingwersen and Järvelin, 2005] Peter Ingwersen and Kalervo Järvelin. *The Turn - Integration of Information Seeking and Retrieval in Context*. 2005.

[Ingwersen, 1982] Peter Ingwersen. Search procedures in the library analyzed from the cognitive point of view. *Journal of Documentation*, 38:165–191, 1982.

[Ingwersen, 1994] Peter Ingwersen. Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction. In *SIGIR '94: Proc. of the 17th Int. ACM SIGIR conf.*, pages 101–110, New York, NY, USA, 1994.

[Joachims *et al.*, 1997] Thorsten Joachims, Dayne Freitag, and Tom M. Mitchell. Web watcher: A tour guide for the world wide web. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 770–777, 1997.

[Karnik *et al.*, 2005] M.V. Karnik, S.K. Gupta, D.K. Anand, F.J. Valenta, and I. A. Wexler. Design navigator system. In *Proc. of IDET/CIE 2005, Long Beach, California, USA*, September 24-28 2005.

[Lauer *et al.*, 2007] Wolfgang Lauer, Josef Ponn, and Udo Lindemann. Purposeful integration of product models into the product development process. In *Proc. of the Int. Conf. on Engineering Design, ICED'07*, 2007.

[Lee and Lai, 1991] Jintae Lee and Kum-Yew Lai. What's in design rationale? *Human-Computer Interaction*, pages 251–280, 1991.

[Lieberman, 1997] Henry Lieberman. Autonomous interface agents. In *Proc. of the ACM Conf. on Computers and Human Interface (CHI-97)*, Atlanta, Georgia, USA, 1997.

[Morgenroth, 2006] Karlheinz Morgenroth. *Kontextbasiertes Information Retrieval*. Logos Verlag Berlin, 2006.

[Nomaguchi *et al.*, 2004] Y. Nomaguchi, A. Ohnuma, and K. Fujita. Design rationale acquisition in conceptual design by hierarchical integration of action, model and argumentation. In *Proc. of IDET/CIE 2004*, Salt Lake City, Utah, USA, September 2004.

[Pfeifer-Silberbach, 2004] Ullrich Pfeifer-Silberbach. *Ein Beitrag zum Monitoring des Reifegrads eines Produktes in der Entwicklung*. Phd thesis, Fachbereich Maschinenbau an der technischen Universität Darmstadt, 2004.

[Regli *et al.*, 2000] W. C. Regli, X. Hu, M. Atwood, and W. Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16:209 – 235, 2000.

[Schilit *et al.*, 1994] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994.

[VDI, 2006] VDI. VDI 2221 - 2223. VDI-Handbuch Produktentwicklung und Konstruktion, 2006.

[Yee *et al.*, 2003] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *CHI '03: Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 401–408, New York, NY, USA, 2003.