

# USING LIGHTWEIGHT KNOWLEDGE MODELLING TO IMPROVE PROACTIVE INFORMATION DELIVERY

Oleg Rostanin, Heiko Maus

*German Research Institute for Artificial Intelligence, DFKI GmbH, Trippstadterstr., 122, Kaiserslautern, Germany*

Takeshi Suzuki, Kaoru Maeda

*Ricoh Co. Ltd, Yokohama, Japan*

**Keywords:** Proactive information delivery, Lightweight knowledge modeling, Concept-based task tagging.

**Abstract:** The current work presents an integrated solution for task-centric proactive information delivery (PID) in agile knowledge working (AKW) environments. The approach exploits a lightweight incremental modeling of task relevant knowledge domains and process know-how using concept maps together with concept-based task tagging to improve the quality of PID results. The feasibility of the described approach was proved during the joint research project TaskNavigator conducted by Ricoh Co. Ltd and DFKI GmbH.

## 1 INTRODUCTION

During the last decade, a plenty of approaches for intelligent user assistance in knowledge intensive working environments were developed. Knowledge intensive work consists of both strictly structured processes that can be formally modeled and enacted using workflow management systems (WFMS) and agile processes (agile knowledge work, AKW) that are highly dynamic that makes them difficult to be formalized (Elst, L.v. et al, 2003).

Our current work concentrates on the support for AKW, e.g., developing a software or writing a project proposal. Although AKW is dynamic, it is required to be managed to be successfully completed in time. Task list management (TLM) tools are often used for flexible time management and planning in AKW environments. TLM tool is an ideal place for intelligent assistance, e.g., proactive information delivery (PID) that is required by a knowledge worker coping with tasks. Generally, PID has two main purposes, i) minimize information overload by providing information adapted to the current task's needs, ii) diminish users' risk of overlooking important documents relevant to their tasks. We distinguish light- and heavy-weight PID depending on the needed modeling effort (Holz, H. et al, 2005).

Heavyweight approaches to learning-on-the-job (Rostanin, O. et al, 2006) aim to educate users by providing information according to users' information needs and skill level. Such approaches claim to ensure a precise information delivery. Their major problem is a relatively large effort on process, user and information modeling needed to introduce these methods in an enterprise.

We propagate a lightweight PID approach in combination with a TLM system TaskNavigator to cope with requirements from AKW. The main goal of our research is to find means to combine advantages of light- and heavy- weight PID, i.e., low modeling effort and high delivery precision.

In this paper we describe concepts of lightweight PID and task tagging (section 2). Using concept maps to improve PID is discussed in section 3. Evaluation results of our approach are depicted in section 4. Conclusions are provided in section 5.

## 2 PID IN TASKNAVIGATOR

TaskNavigator is a web-based TLM system providing support for knowledge intensive business processes (Holz, H. et al, 2006). By the mechanism of task delegation, task comments and notification as well as flexible task structure management implied

by work breakdown structure (WBS), TaskNavigator becomes a powerful tool for work coordination and collaboration in distributed teams.

## 2.1 Lightweight PID

The main idea of PID in TaskNavigator is to proactively deliver task-relevant information e.g., documents, e-mails, web-history, wiki pages related to the task without explicit user request. The principle of lightweight PID is based on the assumption that a task can be described sufficiently by task title, description and comments as well as documents attached to the task. The PID module of TaskNavigator generates a keyword-based query from the current task context represented by task description and its attachments and sends request to external information retrieval (IR) systems automatically to get task-relevant information. Results from the IR systems are sorted by their relevance to the query and presented to the user.

The core advantage of lightweight PID is the low level of human effort needed to make it work: the user just types a task name in TaskNavigator to get first PID results. A formal model of task information needs is not required.

The main problem (**P1**) of lightweight PID is that statistics-based query generation used in lightweight PID can cause unsatisfactory quality of generated queries or search results:

(**P1.1**), TF/IDF algorithm used for a query generation has limitations, e.g. for the task “Introduce new employee” the keyword “new” is regarded as a stop word and removed from the query, although it is essential in the given situation.

(**P1.2**), Compound search terms: Even if the algorithm could identify the importance of the keyword “new” for the task, the keyword does not have sense as a query term without considering it in the combination with the keyword “employee”.

(**P1.3**), Verbose task descriptions can spoil automatically generated query, e.g., for the task “Create new DB for TouchMap weblog” with the description “To install a new wordpress blog we need a separate database on our mysql server” would generate the query “create, db, touchmap, weblog, wordpress, install, separate, database ...” that would result in delivery of no or too many documents.

## 2.2 Task Tagging Improves PID

The objective of the TaskNavigator project was to find an optimal solution that requires a minimally possible modeling effort to achieve acceptable PID

results. Our claim here is that bottom up task modeling realized by collaborative task tagging is feasible and can improve PID results (**C1**).

Tagging is a wide-spread technology for lightweight annotation of electronic resources by manually or automatically assigning keywords to them (Golder and Huberman, 2006). Considering tasks in TaskNavigator as resources that are annotated collaboratively by tags, we decompose C1 into the following sub-claims:

(**CI.1**) Task tags can be used as keywords to refine a search query for task-related PID. Keywords defined by users do not cause problems P1.1 and P1.2 (if multi-word tags are allowed). The implicit semantics behind task tags given by humans will highlight the most important task aspects suppressing the problem P1.3.

(**CI.2**) Provided the bag tagging model is used in TaskNavigator, where different users can tag tasks multiple times with the same tag, the popularity of task-related tags can be used to specify weights of single terms comprising a PID query. A weighted query expresses the importance of each term thus better specifying the task semantics (see P1.3).

(**CI.3**) Provided a list of tags of the parent task is easy available in the current task details, the parent task tags will ease the effort on current task tagging.

In order to implement this new vision on PID, the process of the task-specific information delivery will be extended as follows: i) Propose possible tags to the user proactively; ii) User accepts/rejects tag proposals or tags tasks manually (compound tags are allowed); iii) In collaborative task management environment, users can vote for or against task tags assigned by themselves or by colleagues. iv) A new PID query is generated by TaskNavigator considering tags and tag votes as (compound) search terms and their weights in the query.

Although task tagging can solve problems of lightweight PID, there are severe problems going along with tagging such as synonymy (**P2.1**), homonymy (**P2.2**), polysemy (**P2.3**) - see (Goldman 06). In respect to the information retrieval, the problem **P2.1** (includes synonyms, misspelling, different writing styles and different languages) is the most critical. Provided the user tagged the task with “digitalpaper”, documents containing “digital paper” or “digitales Papier” (Ger.) will not be found by the IR engine. The problem of homonymy can emerge, for example, if the user tagged a task with “SME” assuming “subject matter expert” but received documents about “small and medium enterprises”. The problem of polysemy is sometimes difficult to recognize but it can spoil the IR results:

while expecting a description of the TouchMap system, the user receives documents about the TouchMap project which are related but not same.

### 3 TASK MODELING WITH CONCEPT MAPS

A standard way of solving the problem of synonymy and misspelling is to use mechanisms supporting controlled vocabularies during the tagging process. To solve homonymy and polysemy problems, more sophisticated ontological modeling of the task-relevant domains can be done. However, a sound modeling of task context is practically impossible for every task in TaskNavigator as they are mostly ad-hoc in nature. Instead, we propose a method for lightweight tasks modeling that realizes function of the vocabulary control. The proposed solution is based on the idea of concept maps.

LeCoOnt (<http://lecoont.opendfki.de>) is a web-based tool for collaborative concept mapping developed at DFKI (figure 1). It is aimed to combine the graphical expressiveness and intuitiveness of concept maps, a simple but well-defined information model as well as vocabulary control to provide a universal platform for lightweight knowledge modeling using the concept map paradigm.

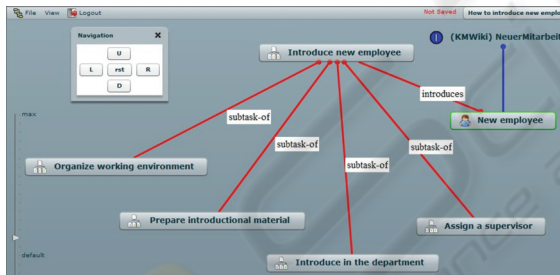


Figure 1: Task and domain modeling with LeCoOnt.

TaskNavigator integrates LeCoOnt as means to control the vocabulary used for task tagging: the LeCoOnt service realizes the auto-completion function for manual tag input. The user can select proposed concepts as task tags or create new tags. Newly created tags are stored in the LeCoOnt database as an unbound concept that can be later used for domain modeling in LeCoOnt.

#### 3.1 Concept-Based PID

By introducing the controlled vocabulary for task tags, we are able to use it to identify concepts from the knowledge base matching the current task and

thus not to rely on results of statistics-based keyword extraction (P1.1-3). Figure 2 shows a recommendation to add the concept “New employee” as a tag for the task “Introduce John Smith”. The user can tag the current task with proposed concepts or attach concept information items to the task. Figure 2 (left middle) illustrates tags accepted by the user and attached to the task “Introduce John Smith”.

The labels of attached concepts together with their alternative labels will be used by the PID engine to generate new queries. A simple query expansion realized by using concept alternative labels will ease the problem of synonymy (P2.1). Furthermore, relations of the concept used for tagging to other concepts in the knowledge base can be exploited to disambiguate meanings of keywords presented by tags and filter delivered documents.

#### 3.2 Conceptual Task Modeling

Whereas the task tagging represents a bottom-up approach to task modeling, the LeCoOnt tool can be used as means to lightweight top-down task modeling. In figure 1 an informal process model “Introduce a new employee” created in LeCoOnt is shown. Having attached the concept “Introduce a new employee” as a task tag, a TaskNavigator user can decompose the task into subtasks according to the task model defined in the concept map. Created subtasks will be automatically tagged by corresponding concepts from the concept map and inherit information items attached to the concepts.

## 4 EVALUATION

In order to show the feasibility of the approach, a case study was conducted at the DFKI that lasted for 3 months. Totally, 11 subjects took part at the experiment: 4 students, 9 researchers and 2 consultants. During the case study, users created 376 tasks as well as attached 624 documents and 164 comments to their tasks. We classified users in two groups: 7 users those who used TaskNavigator for part of their work and initiated 97% of the tasks; and ii) the rest with rather short usage period small number of own created tasks. The type of tasks conducted with TaskNavigator ranged from personal tasks such as workshop preparation or writing publications to project tasks such as project organization or customer relations.

Over the case study period, 458 tags were added to tasks. During task tagging, 70 new concepts were

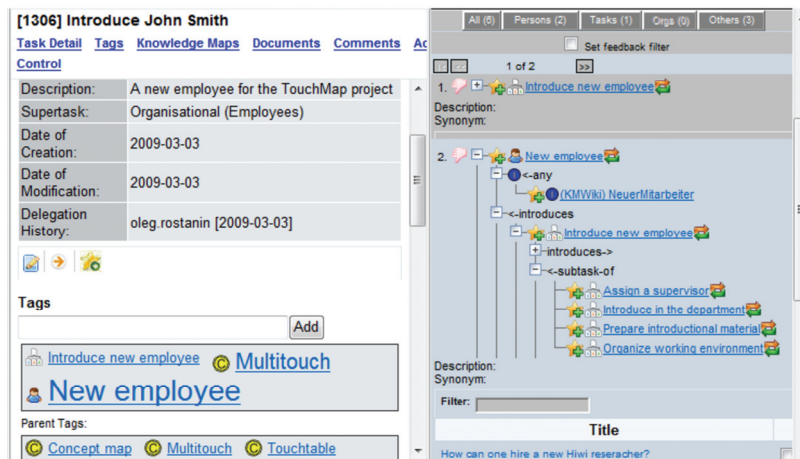


Figure 2: Concept-based PID.

created. Considering both numbers of tasks and given tags, each task got enriched description by 1.2 tags in average.

Over 80% of tags were reused by some means, which means a number of tags being used in the system is fairly maintained to reduce risks introduced with tagging. Over half (54%) of the tags were automatically provided by the system. Finally 24% of the tags were proposed by the concept-based PID and added to tasks by users. For the controlled tasks, the subjects compared the query generated from the task's textual context to the query generated from the concepts attached to the tasks. Once tags were available, usually the tag-based query terms were rated better. The overall impression of the subjects was, that both, lightweight and tag-based PID compliment each other, therefore, they should be used in combination.

## 5 CONCLUSIONS

The uniqueness of the TaskNavigator approach of concept-based PID is in using lightweight concept maps instead of formal ontologies to describe knowledge domains and support task tagging. According to our case study, a bearable user effort spent for task tagging, either manual or supported by the system, allows to improve results of PID as well as to develop the corporative knowledge base.

As a feasibility test with real users showed, both lightweight and extended PID approaches complement each other and should be used together. Whereas the concept based PID solves many problems of lightweight one, lightweight PID can help to solve the problem of a system cold start

specific to tag-based PID: if there are few concepts available in the knowledge base, lightweight PID keyword proposals can be used to initialize it.

Some conceptual aspects could not be tackled in the project's time frame: e.g., the PID engine used in this work considers neither different user skill and knowledge levels. Another critical issue is a seamless integration into the user's workspace.

## REFERENCES

- Elst, L.v., Aschoff, F.-R., Bernardi, A., Maus, H. and Schwarz, S. (2003). Weakly-structured workflows for knowledge-intensive tasks: An experimental evaluation. IEEE WETICE Workshop on Knowledge Management for Distributed Agile Processes (KMDAP03). IEEE Comp. Press.
- Golder, S.A. and Huberman, B.A. (2006). Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32, 198-208.  
 Url: <http://jis.sagepub.com/cgi/reprint/32/2/198>
- Holz, H., Maus, H., Bernardi, A. and Rostanin, O. (2005). From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management. *Journal of Universal Knowledge Management*, 2, 2005, pp. 101-127.
- Holz, H., Rostanin, O., Dengel, A., Suzuki, T., Maeda, K., Kanasaki, K. (2006). Task-Based Process Know-how Reuse and Proactive Information Delivery in TaskNavigator. In *Proc of 15<sup>th</sup> ACM Conference on Information and Knowledge Management CIKM '06*. Arlington, USA.
- Rostanin, O., Holz, H., Ullrich, C. and Song, Sh. (2006). Project TEAL: Add Adaptive e-Learning to your Workflows. In Tochtermann, K., Maurer, H. (Eds.) *Proc. of 6<sup>th</sup> International Conference on Knowledge Management, I-KNOW'06*. Graz, Austria.