Chapter 6
PASTREM: Proactive Ontology Based Recommendations for Information Workers

Benedikt Schmidt, Eicke Godehardt, and Heiko Paulheim

Abstract Information work involves the frequent (re)use of information objects (e.g. files, web sites, emails) for different tasks. Information reuse is complicated by the scattered organization of information among different locations. Therefore, access support based on recommendations is beneficial. Still, support needs to consider the ad-hoc nature of information work and the resulting uncertainty of information requirements. We present PASTREM, an ontology-based recommender system which proactively proposes information objects for reuse while a user interacts with a computer. PASTREM reflects the ad-hoc nature of information work and allows users to switch seamlessly between recommendations for more multitasking oriented or more focused work. This chapter describes the PASTREM recommender, the used data foundation of interaction histories, data storage in an ontology and the process of recommendation elicitation. PASTREM is evaluated in comparison with other, activity related recommendation approaches for information reuse, namely last recently used, most often used, longest used and semantically related. We report on strength and weaknesses of the approaches and show the benefits of PASTREM as recommender which considers the difference between single task focused and multitasking oriented recommendations.

6.1 Introduction

Information is a resource for as well as a product of information work. Within the daily work process, numerous information objects (e.g. files, web sites) are created, modified or consumed using different applications. The sheer amount of accessed

B. Schmidt (✉) · E. Godehardt
SAP Research, Darmstadt, Germany
e-mail: benedikt.schmidt@sap.com

E. Godehardt
e-mail: eicke.godehardt@sap.com

H. Paulheim
University of Mannheim, Mannheim, Germany
e-mail: heiko@informatik.uni-mannheim.de

T. Hussein et al. (eds.), *Semantic Models for Adaptive Interactive Systems*,
Human–Computer Interaction Series, DOI 10.1007/978-1-4471-5301-6_6,

information and the difficulty or impossibility of managing the information results in information overload (stated in a study among 124 managers) (Farhoomand and Drury 2002), threatening the effective and efficient use of information to execute work. As an effect, information retrieval and access present themselves as dominant activities of the workday (Jensen et al. 2010). Interestingly enough, the same information object might be searched for several times. Information objects are reused when an interrupted task is resumed, when they seem appropriate in the context of another task or as a template or data provider for other information objects (Jensen et al. 2010). Each time, the location of an information object is forgotten a duplication of earlier retrieval activities follows.

Different tasks and different state of knowledge of the worker foster different information needs and result in an uncertainty of an information workers' information requirements throughout the work day. An uncertainty which complicates the support of retrieval and access activities.

The reuse of already accessed information objects is supported by features like histories, recently used file lists or manually maintained favorite lists (Bergman et al. 2009). Histories and recently used file lists show a list of previously accessed information object (e.g. the last 10 accessed documents). Favorite lists are manually maintained lists used to structure frequently used information objects. One limitation of these approaches is the scope: generally they are limited to one specific application (e.g. history of a web browser, recently used files of a text processor). Additionally, the size of the lists is frequently restricted to maintain readability (a list of more than 10 items is hard to read). Due to the limitations, other retrieval techniques—not considering the reuse characteristic—like information search are frequently applied for reused objects.

In this chapter, the access of previously used information objects is supported by a recommender approach named PASTREM.[1] Recommenders are generally used to help users to explore information collections under uncertainty. This is achieved based on rating the suitability of items for a user by identifying preference information (Adomavicius and Tuzhilin 2005). Preference information results from observed activities (e.g. which products were watched and which were bought in an online store). The reuse of information in information work can benefit from a similar approach. We consider user activities at the computer desktop as criteria for the recommender system and previously accessed information objects as data source to develop the PASTREM recommender. PASTREM uses a topic model based approach, resulting in the unsupervised recommendation of information objects based on recent activities.

PASTREM has been developed in the context of a toolset to support information work based on data collected from user activities (Schmidt et al. 2011a, 2012). The existing architecture creates an ontology that contains detailed information about the activities of the user, including accessed information objects, time spent with the information objects, the respective content and the activities performed on the

[1]PASTREM refers to the supported process: the REMembering of useful information objects which already have been used in the PAST.

information objects. PASTREM uses this data, extends it by a representation of topics, relevant for the user, created by a topic model approach. The identified topics are composed of words and are linked to information objects. While the user is working, PASTREM basically identifies active topics based on the content of the information objects in the interaction stream of the user. Within the ontology, the active topics link to information objects which are ranked and which are proactively proposed to the user.

The remainder of this chapter is structured as follows. First, information work is discussed to underpin the claim that uncertainty with respect to the information requirement exists, and to highlight the relevance of user activities to derive information requirements. Second, existing recommender approaches in the domain of information work are presented and claims for further research on recommender approaches are derived. Third, the ContAct monitor is presented which is the core component of the toolset PASTREM belongs to. The description of ContAct helps to understand the data used by PASTREM to create recommendations. Core element of ContAct is the computer work ontology (CWO) which formalizes identified user activities. Fourth, the PASTREM recommender approach is described and evaluated. PASTREM is evaluated by comparing the performance of PASTREM to other recommendation approaches, namely last recently user, most often used, longest used and semantically related. All recommenders are compared by measuring the recommendation quality on two existing interaction history data sets of 24 work days. Summary and outlook conclude the chapter.

6.2 Information Work

This section provides a basic understanding of information work. The relevance of information within information work is shown while specific consideration is given to the unpredictability of the information demand due to the dynamicity of work execution. This is the foundations for the later review of existing recommender systems for information work within this chapter.

6.2.1 Multitasking Coordinated by Interruptions

Information workers frequently have a set of different tasks they have to work on. The ad-hoc nature of the information work process results from the way information workers deal with those multiple tasks. Notably, tasks are not processed sequentially, finalizing one task after another. To address constraints (e.g. time) or to react on events, information workers switch tasks, which means that a task is set on hold before it is finalized to start or continue working on a different tasks. Thus, tasks are processed in parallel or in rapid succession (Link et al. 2005), coordinated by interruptions.

Two general types of interruptions can be distinguished (González and Mark 2004; Salvucci and Taatgen 2008): internal and external interruptions. Internal interruptions result from the information worker himself. The information worker decides to switch tasks because of internal stimuli. External interruptions result from events in the environment, external stimuli. Different studies have shown independently that interruptions are evenly distributed among internal and external interruptions (González and Mark (2004) talks about 50 %, Czerwinski et al. (2004) talks about 40 % self-initiated interruptions).

Interruptions at the computer workplace have become increasingly relevant with the computer becoming a multi-task machine (Salvucci and Taatgen 2008). One-task computers discouraged multitasking, whereas the ability to start multiple programs at the same time and access multiple information objects at a time encourages the described multitasking.

A study among Fortune 100 companies (Gallup and San Jose State University and Park, Institute for the Future in Menlo 1999) showed that eighty-four percent of the staffers are interrupted at least three times per hour by messages. In this group, 51 % are interrupted six or more times per hour. Seventy-one percent feel overwhelmed by the message traffic. Czerwinski et al. (2004) reports on an average of 50 goal shifts over a week that were relevant to realize complex goals. Most shifts were triggered by interruptions. Apart from coordinative interruptions, interruptions may as well provide necessary information that is required to realize a goal (González and Mark 2004; Morteo et al. 2004). In this sense, interruptions may even be a core characteristic of work, as Sproull identifies for managers (Sproull 1984).

6.2.2 Uncertainty of Information Requirements

Information is outcome as well as raw material of information work (Aral and Brynjolfsson 2007). First, information work produces information as instrument for illocutionary and perlocutionary acts in Austin's sense (Austin 1962). The individual executes an act by creating a certain piece of information (illocution)[2] or the individual disseminates information (which can also be the modification of symbols in computers) to have a following effect in the real world (perlocution). Second, the work execution itself builds on information, external information accessed and transformed within the work process as well as information which is internalized in the individual (Polyanyi 1966).

Uncertainty with respect to the actual information requirements within information work processes follows. Due to the lack of predefined work processes, the overall information requirement for a work task is unknown. Even if the information requirement can be derived, the fragment of internalized information of the information worker is unknown. Only the activities performed by the information worker at least indicates the overall work domain and possible information requirements.

[2] An example is a priest who contracts a marriage.

6.2.3 Information Reuse

Each task switch modifies the information requirement of the information worker and triggers processes of information retrieval and information access to find the relevant information for the task the information worker switched to. If a task is resumed, the search and access activities are duplications of earlier efforts. When a task was tackled earlier, the subject already identified relevant information but probably needs to identify this information again, once the task is resumed after an interruption.

Barreau and Nardi (1995) classified information reuse as (1) ephemeral information, (2) working information and (3) archived information. Ephemeral information is information which has only a short lifespan, e.g. like some emails or a todo list. Reuse of ephemeral information is unlikely. Working information is the information which is actively produced or modified by the information worker over a longer period of time. Reuse of working information is simple, as the information worker generally spends much time with it and knows where he put it. Archived information is information which is used in the work process and has relevance over a long or very long period of time (e.g. weeks and month). Studies show that the reuse of ephemeral and archived information is complex (Barreau and Nardi 1995). The short time span and the large amount of different information types complicates the access of this type of information.

Although the study by Barreau et al. is from 1995, the results seem not outdated. Techniques which support the quick and simple retrieval of earlier accessed information objects without requiring substantial manual maintenance effort are required. Next to the already mentioned software features of recently used lists and favorites additionally, different personal organization techniques may be applied. Examples of personal organization techniques are tags as categorization system, folder structures as classifying system, post-its etc. All techniques are frequently used and require substantial manual maintenance effort while an increasing complexity of the technique additionally complicates the retrieval (e.g. folder structure depth and size positively correlate with retrieval time; Bergman and Whittaker 2012).

6.3 Related Work

Various recommender approaches exist to support information reuse. One way to address the uncertainty of information requirements is the use of interaction histories. Interaction histories are logs of user system interactions generated by software sensors (Kaptelinin 2003). This basic representation of activities gives an understanding of the information relevant for the information worker at that specific moment and to derive potential information requirements. The main differences of the systems exist with respect to a limitation of recommended information types (e.g. only web-sites) and the data source of information to be used for support.

6.3.1 Overview of Approaches

The Dyonipos system (Makolm 2008; Rath and Weber 2008) uses the interaction history to recommend documents, people and locations from the users' personal and the organizational information stores. The recommendations are based on classifiers trained during design time for a set of tasks. The APOSDLE system analyzes user work and identifies documents related to the activities of the user based on a distinction of navigational goals, information goals and transactional goal (Lokaiczyk et al. 2007). Like Dyonipos, the APOSDLE system recommends based on trained classifiers. A limitation of the Dyonipos and the APOSDLE approach is the need to know about existing tasks and information requirements at design time of the system. The TaskTracer system, a personal information management system, uses an extension called TaskPredictor to train classifiers during work execution (Shen et al. 2006). Thus, the limited information about work tasks that occurs in information work is addressed.

Dyonipos, APOSDLE and TaskTracer encapsulate the recommendation logic and sometimes even the used data foundation in the trained model. The black box characteristic of trained models complicates system maintenance and extension. An open and transparent formalization of the used data source and the recommendation logic in form of an ontology is an alternative approach. Middleton et al. developed the Quickstep and the Foxtrot system (Middleton et al. 2004). The system creates interaction histories for the access of research papers and uses the IBk (Aha et al. 1991) classifier to determines a paper class, a research paper belongs to which is added to an ontology. The ontology is used to create recommendations based on the types of research papers accessed over a day and additionally considers explicit user feedback on paper types of interest.

The SPREADR system uses features of user history, location and local time to create recommendations in a spreading activation network which is built based on ontologies (Hussein et al. 2007). Activated features spread the activation among the network. SPREADR has been used to recommend events and artists in an adaptive music portal web-site.

While approaches like Dyonipos,[3] APOSDLE and TaskTracer address all types of information work at the computer workplace, a very straight forward method of training recommendation is used, which requires training effort, during design time, while later maintenance and extension is complex. The recommendations have a short lifespan and are updated frequently. The approaches that use ontologies have been used for specific domains like research papers or a music portal, considering recommendations with a long lifespan.

To address information work based on recommender system that use ontologies, respective domain ontologies are required. Two examples and important results of

[3]Dyonipos uses ontologies only to capture events in interaction histories. The classifiers do not extend the ontology.

this research have been developed in the context of social semantic desktops, within the Nepomuk[4] and the Calo project (Cheyer et al. 2005). Both projects provide an initial ontology which allows a basic classification of things which may have relevance in different information work scenarios, including elements like files, locations and tasks. The ontology of the Nepomuk project is a RDF-S ontology named PIMO (Personal Information Model). Similarly, IRIS provides a personal topic map based on OWL ontologies. After crawling information stores, both ontologies provide a rich presentation of data users are working with. The main use of the data is browsing of the personal information structure. We have developed a comparable ontology, named computer work ontology (CWO) (Schmidt et al. 2011b). The computer work ontologies is capable of managing very different types of information objects which may be used in information work. It has been designed to be used by tools to collect and process interaction histories.

6.3.2 Requirements for Information Reuse Support

Based on the reviewed recommender approaches, requirements for further research in the domain of recommender for information reuse can be identified:

1. *Characteristic*: During design time, there is a lack of knowledge which types of user tasks will be executed and which information requirements may occur when the tool is used.
   Requirement: Recommendation models need to derive recommendations based on data which emerges when a recommender is used, not based on design time assumptions.
2. *Characteristic*: Every required user input, e.g. the maintenance of models or the supervision of a training is a potential interruption.
   Requirement: The creation of recommendation models should require no, or minimal user input.
3. *Characteristic*: System requirements may change over times, requiring maintenance or extension.
   Requirement: Recommender approaches should structure the trained data and the used data source in a way which is open to access, to increase maintainability and extendibility.
4. *Requirement*: Information requirements change frequently during the work time due to multitasking.
   Characteristics: A recommender approach needs to monitor indicators of information requirements closely to align the recommendations, especially if task switches occur.

[4] http://nepomuk.semanticdesktop.org/nepomuk/.

6.4 ContAct Monitor and the Computer Work Ontology

This section presents the ContAct monitor. The ContAct monitor collects interaction histories, processes the interaction data and creates a formal representation of the information workers' work process. The PASTREM recommender approach presented later in this chapter builds on the output of the ContAct monitor.

Basically, the ContAct monitor realizes an interaction history management process, composed of the steps (1) data collection, (2) data processing and (3) data organization. A detailed overview of these steps is given in Schmidt and Godehardt (2011). In this chapter, we give a summary of the involved components with a focus on the computer work ontology, used to formalize the work process.

6.4.1 Data Collection

Data collection in the ContAct monitor is realized with software sensors to store an interaction histories. The existing implementation of the ContAct monitor can be used for Windows 7 and Windows 8. Each time the foreground process changes or the user interacts with the computer, an event is generated which specifies the foreground process, the information object accessed (if available) and the textual content displayed by the object (if available). This data gives a detailed overview of the sequence of the work process with detailed information about the type of information, the user interacts with.

6.4.2 Data Processing

The data processing step enriches the interaction history and derives additional information from the history. The output is a classification of the user activities and an aggregation of activities which were repeated during execution. For example, in an interaction history, multiple switches to a word processor with a similar open document may exist, always accompanied by multiple keyboard inputs. The data processing classifies this as authoring of the respective document and aggregates all respective events.

6.4.3 Data Organization

The work process data that results from the data collection and data processing is stored in the computer work ontology (CWO). The CWO offers a vocabulary of user system interactions based on the DOLCE foundation ontology (Gangemi et al. 2002). This brief presentation follows a detailed discussion of CWO in Schmidt et al. (2011b). In the following, the specific characteristics of DOLCE and CWO are provided.

DOLCE

DOLCE, the "descriptive ontology for linguistic and cognitive engineering" (Gangemi et al. 2002), is a foundational ontology with its roots in cognitive science and linguistics. It provides a top level of categories in which entities can be classified. Notably, the top level category is "particular"—where a particular is something which cannot have direct instances, whereas a "universal" is something which *can* have direct instances. For example, the Eiffel Tower is a universal, since there is a direct instance of it. A building, on the other hand, is a particular, since there is nothing that would be denoted as *the building*. Universals are members of the sets defined by particulars (Masolo et al. 2001).

The top level of DOLCE is composed of four basic categories: ENDURANT, PERDURANT, QUALITY, and ABSTRACT. An endurant is something whose parts a fully present at a given point in time (like a car), while a perdurant is something whose parts are not fully present at a given point in time (like the process of driving with a car). As a consequence, the parthood relation for endurants is only fully defined when adding a time span (e.g. "Alan Wilder was a member of Depeche Mode from 1982 to 1995"), while the parthood relation for perdurants does not require such a time span (e.g. "the 1980s were part of the 20th century"), as explained by Masolo et al. (2001).

Typically, endurants *participate* in perdurants (like a car participating in the driving of that very car). Important distinctions of endurants encompass physical vs. non-physical and agentive vs. non-agentive endurants.

Qualities are entities that can be perceived or measured, like the color and the prize of a car. Every entity may have a set of qualities that exist as long as the entity exists. DOLCE distinguishes physical qualities (such as size or color), temporal qualities (like the duration of a process), and abstract qualities (such as a prize).

Abstracts are entities that neither have any qualities nor are qualities by themselves. A typical abstract is a spatial region or a time interval.

Several extensions to DOLCE exist (see Fig. 6.1). One of the most frequently used is the DOLCE DNS (Descriptions and Situations) module, which is used to formalize communication scenarios. The DNS ontology provides useful concepts for describing such interoperations, such as parameters, functional roles, and communication methods. Due to its wide usage, DOLCE and DOLCE DnS are bundled together in one ontology as DOLCE-Lite. DOLCE-Lite consists of 37 classes, 70 object properties, and 349 axioms.

Based on the DnS extension, two other extensions to DOLCE have been proposed, which are useful foundations for using ontologies in the field of software engineering. The *DDPO* (Dolce and DnS Plan Ontology) (Gangemi et al. 2004), which defines categories such as tasks and goals, as well as constructs needed to account for the temporal relations, such as preconditions and postconditions. The *information object ontology* (Gangemi et al. 2004) defines information objects (such as printed or digital documents) and their relations to actors and real world entities. Based on these foundations, Oberle et al. (2006) have defined ontologies of software and software components.

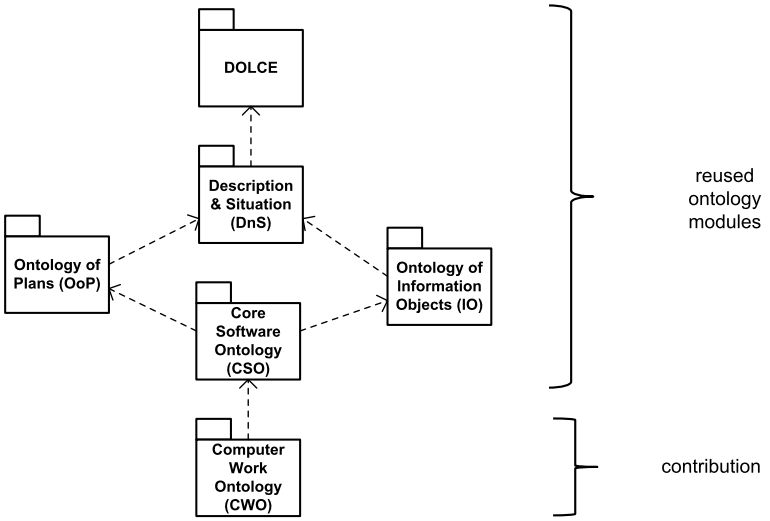

Fig. 6.1 Overview of the ontologies. *Dotted lines* represent dependencies between ontologies. An ontology O_1 depends on O_2 if it specializes concepts of O_2, has associations with domains and ranges to O_2 or reuses its axioms

CWO Modeling Computer Work

The CWO is modeled by considering the computer workplace as an environment that offers functionalities of generating, displaying and transforming data which can be consumed as information. The functionalities and the available information define a possibility-space for the execution of work. Functionalities are encapsulated in software tools and information is stored in files.

Aspects related to the computer like data are modeled based on the CSO ontology (Oberle et al. 2006). Files realize a connection between meaningful information and software as data in a digital encoded representation.

First, we describe the representation of information by files (see Fig. 6.2). We model a CWO:FILE[5] as a role played-by only CSO:DATA. As CSO:SOFTWARE is a subclass of CSO:DATA, we cover software as files (see Fig. 6.2). CSO:Abstract Data is another subclass of CSO:DATA, containing data that identifies something different from itself, e.g. the word *tree* that stands for a mental image of a real tree. As a file may be abstract data or software, two aspects of files are supported: (1) being a static information object, and (2) being an information object for execution to make plans accessible in a runtime representation. A file as a static information object is modeled by relating the file as CSO:DATA by DNS:ABOUT with a DNS:DESCRIPTION. A file as an executable information object relates

[5]From now on and throughout the paper entities that belong to CWO are given without prefix. For all other entities, the respective prefix is given.

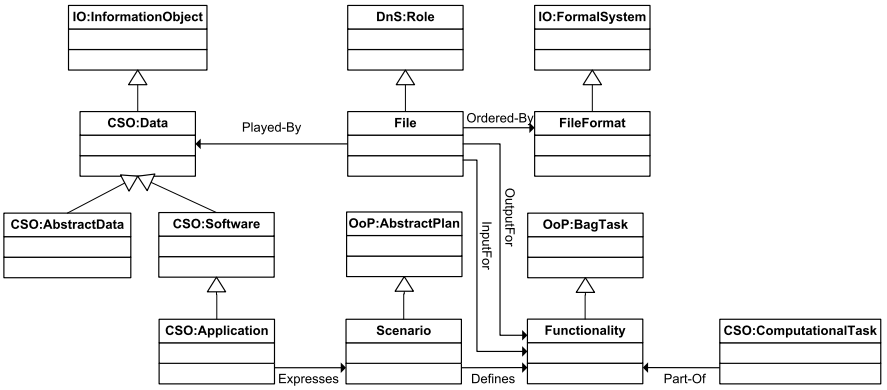

Fig. 6.2 The classification of software with scenarios, functionalities, and files. Concepts taken from DOLCE and accompanying ontologies are labeled with the respective name space

CSO:SOFTWARE with OOP:PLAN by the DNS:EXPRESSES relation. This is given by the following definitions:

(D1) File-Format(x) → IO:Formal-System(x)
(D2) specializes(x, y) ∧ File-Format(x) → File-Format(y)
(D3) uses(x, y) ∧ File-Format(x) → File-Format(y)
(D4) File(x) $=_{def}$ DnS:Role(x) ∧ ∃y(ordered-by(x, y) ∧ File-Format(y)) ∧ ∃z (played-by(z, x) ∧ (AbstractData(z) ∨ Software(z))) ∧ ∀f(inputFor(x, f) → Functionality(f)) ∧ ∀g(outputFor(x, g → Functionality(g)))

A CWO:FILE is DNS:ORDERED-BY a CWO:FILE-FORMAT. A CWO:FILE with specific CWO:FILE-FORMATS can be input for CWO:FUNCTIONALITY. This connection organizes the file access by functionalities, which may range from opening the file to displaying content in a work processor or to the interpretation of a web page by a web browser.

To express content extracted from a file, a DNS:ABOUT relation between CSO:ABSTRACTDATA and the respective entity is created.

By modeling files in a way that they can stand for software, a file which represents a website can capture a service. CSO:SOFTWARE is IO:REALIZEDBY a CSO:COMPUTATIONALOBJECT. Services use functionalities to express scenarios. This is given with the following definitions:

(D5) CSO:Functionality(x) $=_{def}$ OoP:BagTask(x) ∧ ∃y(DOLCE: part-of(y, x) ∧ ComputationalTask(y))
(D6) Scenario(x) $=_{def}$ OoP:Abstract-Plan(x) ∧ ∀y(DnS: defines(x, y) → Functionality(y))
(D7) CSO:Application(x) $=_{def}$ CSO:Software(x) ∧ ∃y(IO: realizedBy(x, y) ∧ CSO: ComputationalObjects(y)) ∧ ∀z(IO: expresses(x, z) → Scenario(z))

The described aspects allow the use of the CWO ontology to create personal information models comparable to those given with PIMO and the IRIS ontologies.

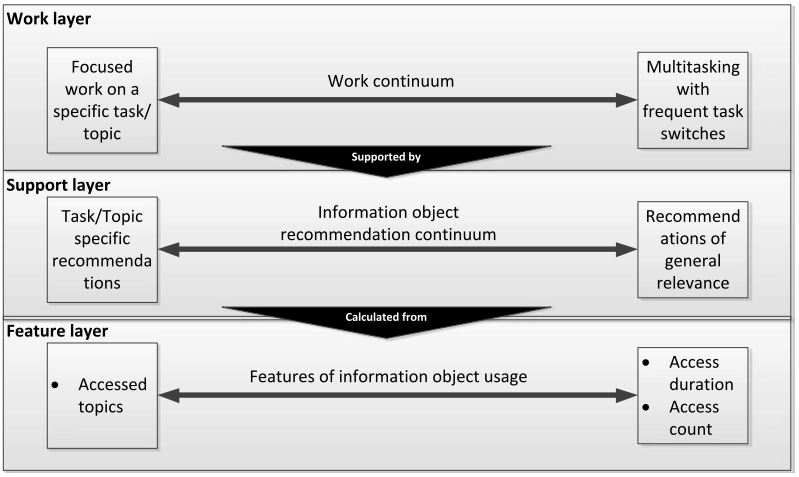

Fig. 6.3 Work continuum, related recommendation continuum and influence of features

The output of the organization step of the ContAct monitor is a CWO representation of the user work. This data can be stored to create an archive of user system interactions, or it can be directly forwarded to subscribed applications. In the following, both types will be used. The stored history is used to get an understanding of the general history of the user. The direct forwarding helps to understand the short term activities of the user which describe his situation and hint to existing information requirements.

6.5 PASTREM Recommender

This section presents the PASTREM recommender approach. The PASTREM recommender builds on the CWO instance data created by the ContAct monitor and extends it. The PASTREM recommender approach supports information reuse for information workers for a more focused or a more multitasking oriented work. The approach especially tackles the requirements of (1) creating models for the recommender based on and within the actual work process, (2) limiting the required user input for the recommender system (3), structuring recommendation data in an easily accessible way to improve maintainability, and (4) respecting the dynamism of information work.

6.5.1 PASTREM Recommendation Continuum

PASTREM builds recommendations for information object reuse with respect to a work continuum which goes from an extremely focused, single task work to multitasking with frequent task switches (see Fig. 6.3). The assumption is that the actual

useful recommendations differ. A very focused work may be supported by information objects which are closely related to the task, considering even information objects which have been accessed very few times until that moment. In contrast, a multitasking oriented work requires recommendations which support the task switches by providing information objects as anchor points for upcoming tasks. An anchor point is an information object of high relevance which helps the user to quickly recall conditions and requirements of a task, like a memory cue that supports a task switch. Therefore, multitasking oriented work would probably be supported best by information objects of general importance. Thus, the work continuum triggers a continuum of recommendations, focusing more or less on focused or multitasking work respectively.

For PASTREM three activity features are used: user topics, access count and access duration. Topics capture an abstract representation of information requirements of the user generally related to the task a user works on. A latest time segment of user interaction is used to identify relevant topics which hint to related information objects in the interaction history of the user captured by the CWO. Topics can be understood as an information requirement following the assumption that a user continues to work on a focused task. Thus, topic related recommendations help users to focus on specific topics. Access count and overall access duration are global characteristics, not related to the given focus task. Therefore, access count and access duration support task switches as they result in information object recommendations of general high relevance, possibly unrelated to an active task but serving as memory cues for task switches.

In the following, information about topic modeling and the integration of topics into the CWO is provided. Then, the overall process of PASTREM is presented, including data preparation and recommendation elicitation (see steps in Fig. 6.4).

6.5.2 Topic Modeling for CWO

Topic modeling stands for a group of approaches which use Bayesian parameter estimation on multinomial distributions frequently used to derive the latent semantics of a text corpus. PASTREM uses the Latent Dirichlet Allocation (LDA) (Blei et al. 2003) to derive topics as latent semantics from a user interaction history as text corpus. In the following, a brief description of LDA is provided and the integration of topics, extracted from interaction histories, into the CWO is described.

The model assumption of LDA is that documents are composed of topics, while each topic is a set of words. Creating a document means choosing the required topics, their relevance for the document and sampling the words from the set of topics. LDA reverts this process and extracts a generative probabilistic model from a text corpus using Bayesian methods (for a good introduction, see Heinrich 2009). The model describes the probability that a word is part of a topic and the probability that a topic was used to generate a document.

Input for LDA is a bag of words representation of documents, i.e. the words used in the corpus are enumerated and for each document the count of each word is noted.

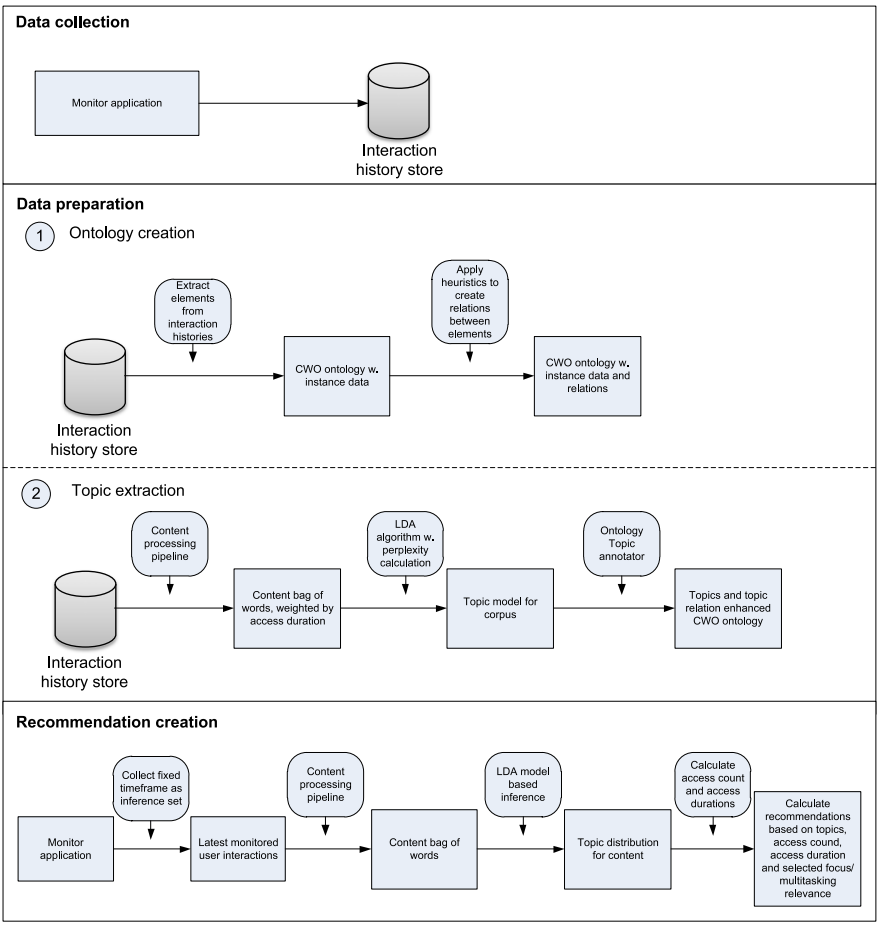

Fig. 6.4 Processes involved in the recommendation creation

6.5.3 Adding Topics, Access Count and Access Duration to CWO

The extended information object design pattern (Gangemi et al. 2004) describes the modeling of an information object. An information object can be realized by any sort of entity and can be about any sort of entity. To express that a file has a content which stands for different topics the following model applies: the file plays the role of abstract data, as discussed above, and the abstract data expresses a topic which is modeled using the subject entity. As the topic extraction identifies a value which stands for the relatedness of the data to the topic we have applied reification.

An IO:SUBJECT gets connected to a CSO:MEASUREMENT unit with a property of type DNS:REFERENCES. The CSO:MEASUREMENTUNIT is again connected to an IO:INFORMATIONOBJECT. The measurement unit contains the relatedness value.

For access count and access duration, the extraction is simpler. They can be derived from the CWO based on the logged work situations which refer to information objects. The situation number for each information object needs to be counted to get the access count while the access duration is provided by the sum of the situation durations for each information object.

6.5.4 Data Preparation

The data preparation described in the following especially focuses on the extraction of topics from the interaction which requires most effort within the recommendation process. Data preparation creates two artifacts which are used in the recommendation process. On the one hand, an instance of the CWO ontology is created and annotated with information about topics and the relatedness values for information objects. On the other hand, a model of the user topics is created, which is later used to infer topic distributions of new documents.

Data preparation is a time consuming task which needs to be performed on a regular basis (e.g. daily):

1. *Ontology creation*: First, the CWO ontology is filled with instance data about the elements the user interacts with. Based on the classification of information objects and additional heuristics, CWO instances are extracted. The resulting CWO ontology links information about the information objects, services and applications a user interacted with. The CWO also includes information about work episodes, thus providing data about access count and access duration of the information objects. This is the output of the ContAct monitor.
2. *Topic model creation and ontology enrichment*: Second, the content of the interaction history is used to identify topics of the accessed content. This is done using LDA, which requires a bag of words representation of the content as input. The bag of words is created in a document processing pipeline, as it is frequently used in natural language processing tasks (Nadkarni et al. 2011). The pipeline contains the following elements: tokenizer, language detection based on n-grams, part of speech tagging and stopword detection. Stopwords are deleted and only nouns and verbs are processed further.
   The pipeline creates content representations as bags of words: lists of words with the number of occurrences.
   The corpus represented by sets of bag of words is input to LDA. The LDA algorithm creates two distributions: a distribution of words to topics and a distribution of topics to documents. The LDA algorithm requires the input of topics before the algorithm runs. As the amount of useful topics generally is not known, a workaround can be used. The perplexity "is monotonically decreasing in the likelihood of the test data, and is algebraically equivalent to the inverse of the geometric mean per-word likelihood" (Blei et al. 2003). The lower the perplexity score, the better the generalization performance. If LDA is executed several

times for different amounts of topics, the perplexity indicates the topic amount with the best generalization performance.
The ontology created in the previous step is enriched by the new data. Each topic is added as a topic entity represented by IO:Subject to the ontology. As described in the previous section, a CSO:MEASUREMENT unit connected with DNS:REALIZES connects CSO:ABSTRACTDATA played by the file and the IO:Subject.
The output of the step is not only the ontology enriched with the topic and topic relatedness data. The second output is the model of document, word and topics created by the LDA algorithm which is used later for inference.

6.5.5 Recommendation Creation

Recommendations are proactively generated while the user is working. While access count and access duration are directly available, the relevant topics are derived from the latest interaction history. Therefore, the most recent segment of the users' interaction history is used as inference set to identify the relevant topics.

The textual content of the interaction history fragment is used to identify recommendations based on the CWO ontology. To create recommendations, first a bag of word representation of the content is created using the document processing pipeline mentioned. The access date has no influence on the recommendation creation. The topic distribution for the content is inferred based on the model of document, word and topics created in the previous step. As a result a numerical representation of the topic relevance for the work in the considered latest time frame is created. The information object relevance ($IOTOPIC_{Rel}$) value is composed of the accumulated relatedness of the inference set to the topics and of the topics to the information objects: $IOTOPIC_{Rel} = (\sum_{t=1}^{T}(IS_t + \sum_{i=1}^{I} IO_{it}))$ with $T =$ number of topics, $I =$ number of information objects, $IS_t =$ relatedness of Inference set to topic t, IO_{it} as relatedness of information object i to topic t. Thus, the relevance of a topic for the latest time segment adds to the relevance of all information objects for the topic.

For each information object, the relevance (IO_{Rel}) for the recommendation is calculated as a product of the topic relevance, the access count and the access duration weighted by factors to increase or decrease the relevance of focused or multitasking work respectively: $IO_{Rel} = IOTOPIC_{Rel}^{\beta} * ac^{\alpha} * ad^{\alpha}$ with ac as access count, ad as access duration in minutes and α and $beta$ to trigger the relevance of topics for focused work and of ac and ad for multitasking oriented work.

6.5.6 PASTREM Discussion

PASTREM addresses the needs identified for recommendation approaches for information reuse based on topic extraction on the long term interaction history and

topic inference on the short term history. The specific demands are tackled by this approach in the following way:

1. *Requirement*: Creating models for the recommender based on and within the actual work process.
   Addressed: The topic model created by LDA is the model used to generate recommendations based on the interaction history of a user.
2. *Requirement*: Limiting the required user input for the recommender system.
   Addressed: LDA is an unsupervised algorithm which only requires the work process information provided by the ContAct monitor and captured in the CWO. Access count and access can be calculated from the interaction history.
3. *Requirement*: Structuring recommendation data in an easily accessible way to improve maintainability.
   Addressed: The use of the CWO to capture an abstract representation of the computer work, accessed information objects, topics and the relatedness of topics to information objects provides simple access to the data used for recommendation elicitation. Extension of CWO to other types of accessed information is simple, as long as a textual representation of the information is given.
4. *Requirement*: Respecting the dynamism of information work.
   Addressed: The frequent creation of recommendations based on the most recent interaction history segment helps to consider the latest topic of interest which may change the information requirement quickly within the recommendation. The ability to increase or decrease the relevance of topics on the one hand and access count/access duration on the other hand helps to increase or decrease the relevance of focused or multitasking oriented work episodes.

6.6 Evaluation

In the following, the PASTREM recommender is evaluated and compared to the results of other activity related recommenders: last recently used (LRU), semantic relatedness (TR), most often used (MOU) and longest used (LOU). LRU, MOU and LOU are-self explaining. The TR algorithms recommends only based on the relatedness of the topic of the considered time segment to stored topic models with related information objects. Especially, MOU and LRU are frequently used recommender types used in applications (often referred to as *recently used lists* or *histories*).

The evaluation is conducted in an ex post manner. Two interaction history data sets are used to identify the number of correct recommendations at a given position in the history by checking whether the elements actually accessed by the user would have been recommended. This results in a binary decision whether a used resource was recommended or not with a hit percentage.

The evaluation process is described in the following. Information objects are identified which have been used in a real use time segment after a randomly selected starting point (see Fig. 6.5, start point) in the interaction history and which were used earlier. The information objects of the real use time segment are compared to

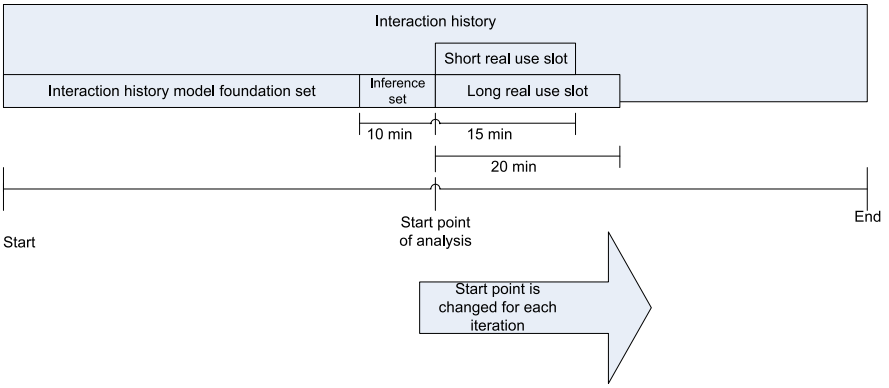

Fig. 6.5 Timeframes relevant for recommendation analysis for a given starting point

the recommendations generated by the recommender approaches, i.e. it is checked how many of the reused information objects in the real use slot are recommended by the algorithms (see Fig. 6.5, use slots).

The events before the start position are used to create recommendations. Therefore, they are separated in two sets: (1) Model foundation set (2) Inference set. To ensure a sufficiently large number of events to build the model, it was enforced that the start position was in the middle or later of the interaction history. The recommendation inference set is a time segment of 10 minutes before the selected position. This time segment is used for the recommendation creation. All events that occurred before than the recommendation inference set are used to build the ontology and to perform topic extraction (see Fig. 6.5, model foundation and inference set).

6.6.1 Evaluation Configuration

The performance of PASTREM as well as the performance of LRU is scaled by the amount of elements included in the recommendation list. If both propose a list of all elements the user ever interacted with, both have the best possible recall but a low precision. This has practical relevance for the user interface of the recommender. A longer list of recommendations complicates user interactions due to limited cognitive capabilities. Therefore, the number of recommended elements is of high importance: the lower the number of recommendations required to make a valid recommendation, the better.

To address this, different recommendation set sizes have been compared: 10, 15, and 20 information objects. The ranking was performed as follows. For LRU the last n elements which were used directly before the begin of the inference set have been used. MOU uses the n most often used elements and LOU uses those n information objects used for the longest amount of time. TR calculates the relatedness of the inference set to topics of the model and the relatedness of the topics to the

information objects (actually the calculation of $IOTOPIC_{Rel}$ described in the previous section). Based on the resulting values, TR recommends the n elements with the highest relatedness. In all cases, elements from the inference set were excluded from the list of potential recommendations, as they are already used.

Another influence factor is the length of the real use slot. The longer the slot, the higher the probability that a recommendation might fit. This has been addressed by considering two different real use slot lengths: 15 and 20 minutes.

A third influence factor is the temporal length of the inference set. Based on experience, we have set the length to 10 minutes. This value has not been changed in the study, although it is worth to investigate it further. The assumption is that the length of a useful inference time segment length depends on the homogeneity of work as measure for multitasking. An inhomogeneous work probably requires smaller inference time segments than homogeneous work.

Two interaction history data sets have been analyzed, using the described process. The α and β value were both set to one, to balance between task-focus and multitask orientation.

6.6.2 Evaluation Process

The interaction history data sets were created by researchers at an IT company. Data set 1 contains 15 363 interaction events (e.g., mouse clicks, window focus, etc.) for a period of 9 work days. Data set 2 contains 18 311 interaction events for 4 work days. Information objects were only considered, if they were at least 10 seconds focused. The data sets represent the normal working day of the two people, starting emails, browsing the internet, reading emails, etc.

For data sets 1 (100 data points) and for data set 2 (80 data points) were chosen randomly with the constraint that at least one third of the overall event number was recorded before the selected event as starting point. The constraint assured that enough information objects and data for reasonable recommendations and topic model creation existed.

Data set 1 contains 620 different information object accesses in all 100 real use time segments for a 15 minutes time segment (elements not included in the inference set). Of those 620 elements, 384 elements had not been used earlier, while 272 elements were reused. For all 20 minute real use slots, overall 765 information objects were used, 436 had not used been before, while 329 were reused. The average number of reused information objects for a 15 minutes real use time segment was 2.7 and 3.2 for a 20 minutes real use time segment. Only three real use slots for 15 minutes as well as for 20 minutes reused more than 20 information objects which means that only for these three elements the largest recommendation set would be insufficient to recommend all items.

Data set 2 contained 287 different information objects accessed in all 80 real use time segments of 15 minutes length. The 287 elements contained 237 elements not used before and 50 reused elements. Within the 20 minute time segments 336

elements were accessed, 267 were unknown before and 69 were reused. An average number of 0.6 elements were reused within 15 minutes, 0.86 were reused within 20 minutes. No slot for 15 or 20 minutes contained more than 20 information objects, thus the recommendations could have been sufficient to recommend all actually used information objects.

The numbers already hint to different work styles captured by the data sets. In the following evaluation, we will see that data set 1 is more multitasking oriented while data set 2 stands for work with less multitasking which has effects on the different assessed recommender algorithms.

6.6.3 Evaluation Results

The accuracy of recommended information objects for PASTREM, LRU, MOU, LOU and TR for data set 1 is given in Table 6.1 and for data set 2 in 6.2. PASTREM shows a good performance on both data sets, as up to 67.2 % and 71.0 % (15 min) of accuracy is reached for a list of 20 recommendation elements and a 15 minutes time segment. For 10 elements 58.1 % (data set 1), 54.7 % (data set 2) and for 10 elements 42.6 % (data set 1), 40.4 % (data set 2) of all information objects used in a 15 minutes segment have been actually recommended.

Interesting results is the performance of MOU for data set 1 compared to the MOU performance for data set 2. While data set 1 reaches 69.3 % of accuracy for 20 minutes length and 20 recommendations, data set 2 only shows an accuracy of 44.7 %. A similar peculiarity is the performance of LRU which shows a good performance on data set 2 reaching an accuracy of 63.6 % for 15 minutes and 20 recommendations while for data set 1 only 49.6 % of accuracy are reached for the same value. The overall weak performance of TR (23.5 % is the highest reached accuracy value) is another notable result. The different performances and especially the peculiarities with respect to the specific characteristics of the data sets are discussed in the following.

6.6.4 Evaluation Discussion

The evaluation showed a good performance of PASTREM for both data sets. The only algorithm with comparable results for data set 1 is MOU which shows a less good performance on data set 2.

Discussion of LOU and TR: LOU shows stable results between 24 and 50 % recommendation successes which show that the usage duration indicates relevance while it is not very useful on its own. The TR recommender shows exceptionally weak results. The assumption is that considering topic relatedness fails to rank the information objects which belong to the relevant topics. Additional relevance indicators are required to rank the information objects of one topic, e.g. frequently

Table 6.1 Data set 1: Accuracy of recommendations for PASTREM, LRU, MOU, LOU, TR for a short (15 min) and longer (20 min) real use time segment of recommendation validity with lists of 10, 15 and 20 elements

	Number of recommendations		
	10	15	20
PASTREM 15 minutes	42.6 %	58.1 %	67.2 %
PASTREM 20 minutes	35.6 %	39.2 %	68.1 %
LRU 15 minutes	41.5 %	42.2 %	49.6 %
LRU 20 minutes	40.1 %	41.3 %	49.2 %
MOU 15 minutes	43.7 %	64.7 %	69.1 %
MOU 20 minutes	43.2 %	64.7 %	69.3 %
LOU 15 minutes	24.2 %	37.5 %	54.0 %
LOU 20 minutes	24.3 %	37.1 %	54.7 %
TR 15 minutes	13.6 %	17.2 %	23.5 %
TR 20 minutes	12.7 %	16.5 %	22.4 %

Table 6.2 Data set 2: Accuracy of recommendations for PASTREM, LRU, MOU, LOU, TR for a short (15 min) and longer (20 min) real use time segment of recommendation validity with lists of 10, 15 and 20 elements

	Number of recommendations		
	10	15	20
PASTREM 15 minutes	40.4 %	54.7 %	71.0 %
PASTREM 20 minutes	36.0 %	47.5 %	59.6 %
LRU 15 minutes	29.5 %	47.7 %	63.6 %
LRU 20 minutes	25.3 %	44.4 %	60.3 %
MOU 15 minutes	31.7 %	41.5 %	44.7 %
MOU 20 minutes	28.3 %	38.3 %	40.3 %
LOU 15 minutes	30.0 %	40.0 %	48.0 %
LOU 20 minutes	27.5 %	37.7 %	44.9 %
TR 15 minutes	16.0 %	20.0 %	20.0 %
TR 20 minutes	14.5 %	18.8 %	18.8 %

used for longer periods of time should be ranked higher than a resource which is only infrequently used for a short time. This is considered in PASTREM based on the integration of additional relevance factors which always influence the semantic relatedness based on an overall relevance (ac and ad are always bigger than 1).

PASTREM, MOU and LRU: A closer investigation of data set 1 showed a strong tendency of the user to switch between tasks. The good performance of MOU most likely results from the frequent task switches which are best supported by recommending resources of an overall relevance without paying much attention to the topic which will change only minutes later. The second data set shows a more fo-

cused work type, even including phases of several minutes without any switch of the focus application. The good performance of LRU results from the stable work provided with data set 2 which creates strong local contexts of a high return probability to earlier used resources. For PASTREM, this data set benefits from topic specific recommendations ranked by access count and access duration.

Overall, the combination of semantic relatedness and relevance within PASTREM shows promising results. Next to the accuracy, the type of recommendations is of relevance. LRU and MOU tend to propose elements which were recently and often used, therefore it is likely that the subject remembers those resources and the respective locations without help. In contrast, a review of the PASTREM recommendations showed that often elements not used for a longer period of time or with a medium access count (not the top 4 and not the last 4) were recommended. Those elements probably represent archived and ephemeral elements which is of specific benefit, as the recall of those elements is complex.

6.7 Conclusion

We have presented PASTREM, a recommender system to support information reuse in information work. PASTREM extends existing work on recommender systems for information work in several respects. The approach covers a broad range of different data types, is completely unsupervised and requires few user input. The use of the CWO ontology to structure the data integrates PASTREM into an existing infrastructure for information work support. A specific benefit of PASTREM is the modification of the algorithm for a more focused or a more multitasking oriented work execution. As the respective calculation is a "cheap" reordering of a list, this modification of recommendations can be triggered by the user during runtime. Another aspect of PASTREM is that it provides an entry point to an ontology based on the topic. The abstract nature of topics seem to be a valuable entry point for browsing and extension of the recommender by other, topic related elements.

PASTREM, TR, LRU, MOU and LOU were evaluated by comparing the recommendations to real information object usages in two collected interaction histories. PASTREM showed better results for both data sets, with a balanced influence of topic relatedness to duration and access count.

Future work will investigate into a user interface for PASTREM. A first implementation makes use of the jumplist in Windows 7. Further research will try to improve the accuracy and consider the automatic calibration of the algorithm to the preferred work style of the user. A calibration which is feasible by applying the technique used to evaluate the recommender performance.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *33*(6), 81–749.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37–66.

Aral, S., & Brynjolfsson, E. (2007). *Information, technology and information worker productivity: task level evidence*. Cambridge: National Bureau of Economic Research.

Austin, J. L. (1962). *How to do things with words*. Cambridge: Harvard University Press.

Barreau, D., & Nardi, B. (1995). Finding and reminding: file organization from the desktop. *ACM SIGCHI Bulletin*, *27*(3), 39–43.

Bergman, O., & Whittaker, S. (2012). How do we find personal files?: the effect of OS, presentation & depth on file navigation. In *Proceedings of the 2012 ACM annual conference on human factors in computing systems*.

Bergman, O., Tucker, S., Beyth-marom, R., Cutrell, E., & Whittaker, S. (2009). *It's not that important: demoting personal information of low subjective importance using GrayArea*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*(4–5), 993–1022.

Cheyer, A., Park, J., & Giuli, R. (2005). *IRIS: integrate, relate. infer. share.* DTIC Document.

Czerwinski, M., Horvitz, E., & Wilhite, S. (2004). A diary study of task switching and interruptions. In *Proceedings of the SIGCHI*.

Farhoomand, B. A. F., & Drury, D. H. (2002). Managerial information overload. *Communications of the ACM*, *45*(10), 127–131.

Gallup and San Jose State University and Park, Institute for the Future in Menlo (1999). Managing corporate communications. In *The information age*. Stamford: Pitney Bowes.

Gangemi, A., Borgo, S., & Catenacci, C. (2004). Task taxonomies for knowledge content. *METOKIS deliverable D*.

Gangemi, A., Guarino, N., & Masolo, C. (2002). Sweetening ontologies with DOLCE. In *Knowledge engineering and knowledge management: ontologies and the semantic web* (pp. 223–233). Berlin: Springer.

González, V. M., & Mark, G. (2004). Constant, constant, multi-tasking craziness: managing multiple working spheres. In *Proceedings of the SIGCHI conference on human factors in computing systems* (Vol. 6, pp. 113–120). New York: ACM.

Heinrich, G. (2009). *Parameter estimation for text analysis* (Fraunhofer Technology Report).

Hussein, T., Westheide, D., & Ziegler, J. (2007). Context-adaptation based on ontologies and spreading activation. In *Proceedings of ABIS '07: 15th workshop on adaptivity and user modeling in interactive systems*.

Jensen, C., Lonsdale, H., Wynn, E., & Cao, J. (2010). The life and times of files and information: a study of desktop provenance. In *Proceedings of the 28th CHI* (pp. 767–776). New York: ACM.

Kaptelinin, V. (2003). UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on human factors in computing systems* (Vol. 5, pp. 353–360). New York: ACM.

Link, H., Lane, T., & Magliano, J. (2005). Models and model biases for automatically learning task switching behavior. In *Foundations of augmented cognition* (Vol. *5*, pp. 510–519). Hillsdale: Erlbaum.

Lokaiczyk, R., Faatz, A., Beckhaus, A., & Goertz, M. (2007). Enhancing just-in-time E-learning through machine learning on desktop context sensors. In *Modeling and using context* (pp. 330–341). Berlin: Springer.

Makolm, J. (2008). DYONIPOS: proactive knowledge management. In *BLED 2008 proceedings* (pp. 475–482).

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., & Horrocks, I. (2001). *WonderWeb deliverable D18. Ontology library* (*final*). WonderWeb project.

Middleton, S. E., Shadbolt, N. R., & Roure, D. C. D. E. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, *22*(1), 54–88.

Morteo, R., Gonzalez, V. M., Favela, J., & Mark, G. (2004). Sphere juggler: fast context retrieval in support of working spheres. In *Proceedings of the fifth Mexican international conference in computer science, 2004. ENC 2004* (pp. 361–367).

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544–551.

Oberle, D., Lamparter, S., Grimm, S., & Vrande, D. (2006). Towards ontologies for formalizing modularization and communication in large software systems. In *Handbook on ontologies*. Berlin: Springer.

Polyanyi, M. (1966). *The tacit dimension*. London: Routledge & Kegan Paul.

Rath, A., & Weber, N. (2008). Context-aware knowledge services. In *Personal Information Management*: *PIM* (pp. 1–11).

Salvucci, D., & Taatgen, N. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological Review*, *115*(1), 101–130.

Schmidt, B., & Godehardt, E. (2011). Interaction data management. In *Knowledge-based and intelligent information and engineering systems*. Berlin: Springer.

Schmidt, B., Kastl, J., Stoitsev, T., & Mühlhäuser, M. (2011a). Hierarchical task instance mining in interaction histories. In *Proceedings of the 29th annual international conference on design of communication (SIGDOC)*. New York: ACM.

Schmidt, B., Paulheim, H., Stoitsev, T., & Mühlhäuser, M. (2011b). Towards a formalization of individual work execution at computer workplaces. In *Lecture notes in artificial intelligence*. *Conceptual structures for discovering knowledge* (pp. 270–284). Berlin: Springer.

Schmidt, B., Godehardt, E., & Pantel, B. (2012). Visualizing the work process—situation awareness for the knowledge worker. In *3rd IUI workshop on semantic models for adaptive interactive systems (SEMAIS 2012)*.

Shen, J., Li, L., Dietterich, T. G., & Herlocker, J. L. (2006). A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th international conference on intelligent user interfaces—IUI '06* (pp. 86–92).

Sproull, L. S. (1984). The nature of managerial attention. In *Advances in information processing in organizations* (pp. 9–27). London: JAI Press.