Complex Systems Informatics and Modeling Quarterly
CSIMQ, Issue 2, April, 2015, Pages 31-45
Published online by RTU Press, https://csimq-journals.rtu.lv
http://dx.doi.org/10.7250/csimq.2015-2.03
ISSN: 2255-9922 online

Technical Perspective on Using Information Demand Pattern in a Collaborative Recommendation System for Improving E-Mail Communication[a]

Dirk Stamer[1], Andrew Ponomarev[2], Nikolay Shilov[2], Alexander Smirnov[2,3]

[1] University of Rostock, Chair Business Information Systems,
Albert-Einstein-Str. 22, 18059, Rostock, Germany,
[2] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS),
39, 14 line, 199178, St. Petersburg, Russia,
[3] ITMO University,
Kronverkskiy pr., 49, St. Petersburg, Russia

Dirk.Stamer@uni-rostock.de, {ponomarev, nick, smir}@iias.spb.su

Abstract. Today e-mail communication information is widely used in organizations to distribute information. The increasing volume of received e-mails hinders an efficient work. It becomes more and more difficult to identify relevant e-mails inside this enormous volume of information. This work presents a solution in a multi-user environment by improving an established e-mail client extension based on information demand pattern with a recommendation system. The contributions of this work are (1) the concept and implementation of the solution for a single-user environment using information demand pattern, (2) the concept and architecture to use the solution in a multi-user environment (3) a detailed technical description about the proposed solutions.
Keywords: Information logistics, information demand pattern, recommendation systems, e-mail.

1 Introduction

Our modern economy is affected by the creation, processing and sharing of information and knowledge as Castells [2] pointed out. The main advantage for an enterprise is no longer just the access to information, but more the access to an adequate information management. A central issue for this is the phenomenon of information overload [3].

In spite of many isolated applications information overload is still a problem in our modern working environment. In order to face this problem not only a single application but also the integration of well-established solutions is the key. Every day, a wide variety of information is produced. This emergence has greatly increased in recent years. The society has changed to an information society. The availability of information can be viewed as an obstacle for a demand-oriented information supply, but finding the information – by the person who needs them. This development does not stop when companies are concerned, so that they see themselves confronted with an ever-increasing amount of information, as studies have shown in [4].

[a] This work extends an already published work, which describes how to improve e-mail communication by using information demand pattern and recommendation systems. Especially detailed technical descriptions were added. For the original work see [1].

Information logistics as an application field of business information systems takes care of this problem and has the goal to achieve the improvement of the information flow in organizations by providing a demand-oriented information supply. Information logistics offers methods, concepts and tools to achieve the improvement. Under the concept of demand-oriented information supply the right information at the right time, in the right quality, in the right form and at the right place for the seeking person is understood [5].

For a demand-oriented information supply in addition to the lack of information, especially the flood of information in companies is regarded as an obstacle. Öhgren and Sandkuhl [6] showed in an empirical study that around two-thirds of the top management and middle management of the Swedish Manager suffer from information overload. In that survey 37% of the participants answered that they were receiving "far too much information" and another 29% were receiving "too much information" [6]. The phenomenon of information overload is often equated with too much information. Speier et al. [7], however, point out that the amount of information, as well as the complexity and the time available for the working person are related to each other.

In companies e-mail has established itself as an important communication medium. Also here a flood of e-mails can be observed which Volnhals and Hirsch [8] have shown in a study. This has negative effect on the quality of decision-making by managers and can lead to economic consequences as well. It is pointed out that the information overload can be defined as the amount of information that exceeds cognitive processing capacity.

Soucek and Moser [9], however, have identified in a survey three facets of information overload through e-mail. These are the mass of incoming e-mails, inefficient workflow and poor quality of communication.

To improve the demand-oriented information supply within e-mail communication an extension of an e-mail client was proposed by Stamer [10]. This extension uses information demand pattern (IDP) as input about the information a worker needs. The extension filters the e-mails according to the user's needs and provides better information supply. This solution fits well the single-user work environment. But normally workers within a team interact with each other during their work. Therefore, integrating the recommendation system in order to enable interaction between the local extension installations enhances this solution.

The contributions of this work are: (1) the concept and implementation of the solution to filter e-mails in a single user environment by using information demand pattern; (2) the architecture and concept to establish the aforementioned solution in a multi-user environment such as a workgroup using recommendation systems technologies; (3) detailed technical descriptions about the proposed solution concerning programming languages, interfaces and data structures.

This work is structured as follows: Section 2 gives an overview about the recommendation systems and information demand patterns in general. Section 3 describes the architecture and functionality of the proposed IDP-based recommendation system. Section 4 summarizes the results, gives some conclusions and the outlook on further research.

2 State of the Art

Due to a better understanding of the solution presented in Section 3, this section introduces the key principles of recommendation systems and gives an overview about information demand patterns.

2.1 Recommendation Systems

Recommendation systems have become widely used nowadays as they help to mitigate information overflow of current life [11], [12]. Any recommendation system involves two entities: users and items. The user is a person interested in interacting with items of a certain kind. On the other hand products, services, web pages, blogs etc. may represent items. There are two crucial points that justify the development of the recommendation system in some domain: a) some items are more interesting (or useful) for a particular user than others; b) there are plenty of items, and the user has no chance to examine them all in order to find the most useful ones. Recommendation systems have much in common with search engines, but then differ in a sense that the user must query the search engine, whereas the recommendation system acts more in a proactive way offering the user items that might be useful without explicit requests. Classical examples are movies, books, and music recommendation systems.

There are three basic approaches to recommendation systems development (not to mention hybrid recommendation systems, which usually employ some ensemble of basic approaches) [13]:

- Content-based recommendation systems (CB);
- Collaborative filtering systems (CF);
- Knowledge-based recommendation systems (KB), also known as constraint-based recommendation systems.

These approaches differ in rationale that is behind the recommendation process, information used, as well as information and mathematical models of users and items.

Content-based recommendation systems [14] are based on the premise that if a user likes some item he or she will probably like similar items. So, there are two pieces of information that make this kind of systems viable: the information about the items the user likes and the pairwise similarity between other items. The former is usually collected during the user's interaction with the system and the latter requires some domain-specific analysis of the item's properties and characteristics. Similarity measures for movies, blogs and books are quite different. Typical pitfall of this kind of systems is the lack of diversity.

Collaborative filtering systems [15] are based on the premise that if two users share a significant part of their interests, other interests may also be common. For example, if it is known that two users assigned high rating to the movies "Green Mile" and "Apollo 13" and one of these users had not watched "Forrest Gump" the system may infer that "Forrest Gump" may be of some interest for that user. It is important to note that this inference has nothing about the fact that Tom Hanks starred in all the mentioned movies, but it is based solely on the fact that the users similarly rated some movies. So, the only information that is used by this kind of systems is users' attitude to various items. This attitude usually takes the form of ratings assigned to items by the users, but also may be derived from some user's behavior peculiarities. As this approach does not rely on item's properties it is rather universal and can be applied to almost any domain. The significant drawback, however, is that without a significant number of ratings the statistical inference becomes unreliable.

Knowledge-based systems [16]–[18] are powered by a set of rules that connect users, context and items. Recommendations here are provided as a result of logical inference and/or constraints resolution. Such systems can also be seen as a kind of expert systems. The development of the knowledge-based system requires significant effort, as all the rules and trends that are automatically inferred (and updated during the system lifetime) by recommendation systems of the other approaches must be formalized and manually represented in some machine-readable form by knowledge engineers. Hence, KB recommendation systems are usually developed in domains where there are experts that can provide comprehensive formalization of item space, for example, for browsing product/services catalogues of some company.

2.2 Information Demand Patterns

The concept of information demand pattern originates from the work in the research and development project Information Logistics for SME (small and medium-sized enterprises) (infoFLOW). The project included seven partners from automotive supplier industries, IT industry and academia. The objectives were to develop a method for information demand analysis [19] and to identify recurring elements in information demand, i.e., patterns of information demand.

Lundqvist [19] has shown in a study in companies that the information demand of an employee depends mainly on the role in the organization that he or she fulfills. The structured collection of this information, which is necessary for the processing of work tasks, was underpinned by the development of the methodology for information demand analysis and has been validated.

After detecting the information needs of roles in a company, Sandkuhl presented the concept of information demand pattern. As with patterns in other disciplines of computer science, these patterns have the purpose to detect a proven solution to a problem in order to reuse it in other application scenarios. With information demand patterns, the identified organizational knowledge is collected in a structured and reusable way. The term 'information demand pattern' is defined according to [20] as follows:

An information demand pattern addresses a recurring information flow problem that arises for specific roles and work situations in an enterprise, and presents a conceptual solution to it.

Information demand pattern consists of five integral components:

- *name of the pattern,*
- *organizational context,*
- *problems,*
- *conceptual solution,*
- *effects.*

The *name* is used to identify the pattern. This is usually the name of the role, which the pattern describes.

The *organizational context* explains the application context in which the pattern can be applied. This can be departments, functions or domains.

Problems represent the difficulties and challenges that the person is facing in performing his/her role in the company. In addition, the duties and responsibilities of the role are subsumed under this point also.

How the described problems of the role can be solved is shown in the section *conceptual solution*. It is divided into three areas to consider: information demand, quality criteria and timeline. Information demand describes the information that is necessary to fulfill the duties and responsibilities of the role. The quality criteria describe the quality in which the information must be available, such as the general importance of the accuracy, the time and the completeness of the information. The timeline represents the time at which the required information must be available at the latest. Figure 1 illustrates the quality criteria for the information demand.

The *effects* part describes effects that may occur if the information is not available or not on time. The occurring of possible effects may be associated with the following dimensions: economic effects, time and efficiency, quality of work, motivation, learning and experience and customer [20].

The concept of information demand pattern has been studied in several other works and its applicability has been validated [21]–[24].

Information Demand	General Importance	Accurate	In Real-time	Complete
Customer Change Requests	Decisive	Decisive	High	Decisive
Policy, Law, Regulation Changes	High	High	High	Decisive
Complaints From Own Production	Decisive	High	Decisive	High
Customer Complaints	Decisive	Decisive	High	Decisive
Test Results	High	High	High	High
Changes on Supplier Side	High	High	High	High
Complementary Information	Nice to Have	Nice to Have	Nice to Have	Nice to Have

Figure 1. Quality criteria for the information demand of the role 'material specification responsible' [20]

3 Architecture and Functionality of IDP-based Recommendation Systems

The overall architecture of the proposed contribution is described in this section first. It will be shown how an e-mail client extension can work together with the recommendation system in order to provide the demand-oriented information by using the e-mail filtering technologies. It will be shown how the solution can be implemented in organizations. The last subsection gives some detailed information about technical aspects.

3.1 Architecture in Enterprises and Organizations

Demand-oriented provision of information is important to avoid disadvantages caused by information overload. Therefore the extension of e-mail clients described in Subsection 3.2 can be used to present to a user the right information at the right time as information logistics proposes. The quality of the results of this solution for a single person depends on the quality of the given input – here information demand pattern. On the one hand this solution does not consider the changing demands of the user depending on their current working context. On the other hand there are no connections between the users to benefit from the continuous information exchange. Therefore the solution will be enriched with the recommendation system, which will be described in detail in the following section.

Figure 2 shows in a conceptual way how our proposed solution can be implemented in an organizational context. Inside this organizational context there are several user contexts. These user contexts represent users in their working environments, who fulfill a role inside the enterprise. During carrying out their duties, roles in organizations have a typical information demand, which is determined by their tasks and responsibilities. This information demand can be gathered with an information demand analysis and described with an information demand pattern. This information demand is somehow abstract, which enables the pattern to be used in different contexts, like in other enterprises. Therefore it is necessary to specialize or to tailor the information demand to the exact user context. Due to the fact, that information demand patterns are right now only textual descriptions, these patterns will be transformed into a machine-readable und machine-interpretable format by using technologies like indexing. This indexed information demand pattern is used by the next step as input. After preparing the information demand pattern for use in the organizational structure the specialized information demand of a role helps to provide a demand-oriented user support, which helps to increase, e.g., the efficiency of the role while accomplishing his or her tasks. From the technology perspective this will be implemented by extending the e-mail clients of the users with a plug-in. This plug-in will do local filtering of e-mails to offer the user e-mails he or she needs at the moment to accomplish his/her tasks. Due to the fact, that most of the employees in organizations use e-mails to communicate with internal or external partners, especially e-mail communication is on the one hand an important medium and on the other hand a medium, which is affected by information overload as already pointed out.

Since the above described user contexts are somehow isolated from each other, we propose the context-based recommendations for organizational support. This means that there should be interaction in between the different user context in order to generate benefits through information exchange. Therefore we propose to use the recommendation system. The previous installed plug-in provides feedback about the user behavior to the recommendation system. The recommendation system itself responds to all installed e-mail plug-ins in order to adapt the e-mail filtering with the newly gained information by other users.

To be able to have advantages of economies of scale we propose to manage all available information demand patterns on a central server. The server itself distributes the actual needed information demand pattern to the local installed plug-in. The central administration will reduce the effort of maintaining and distributing the information demand pattern. This reasonable procedure helps to keep away additional work from each employee. Information demand pattern can be changed or maintained easier, if it is stored at the central server. Figure 2 shows this in detail.

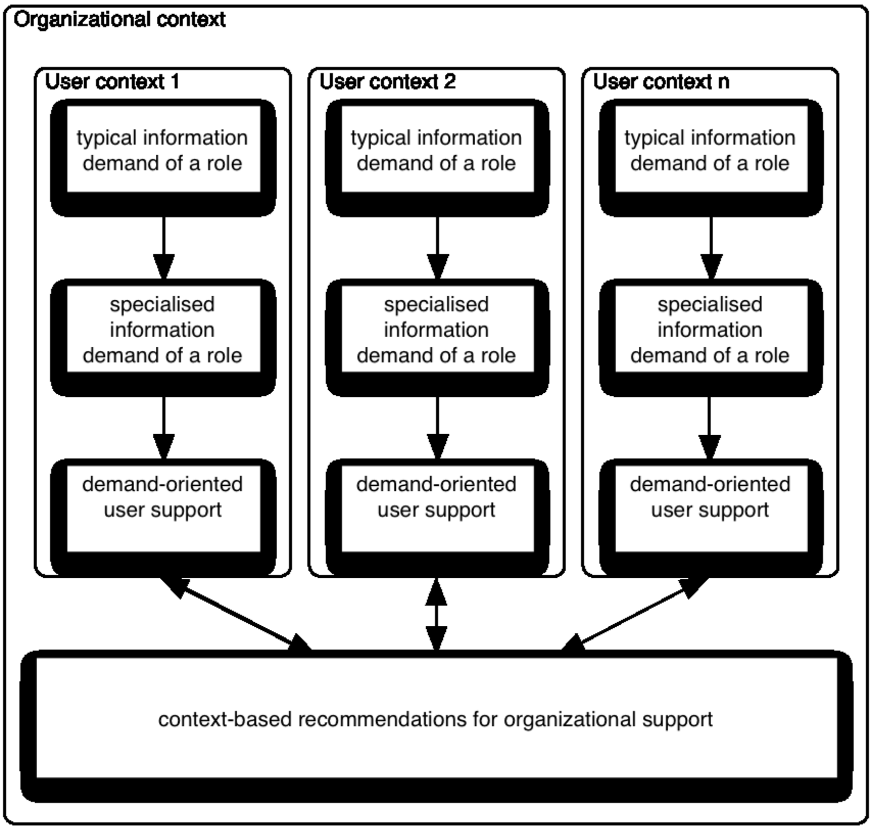

Figure 2. Implementation in an organizational context

3.2 E-Mail Client Extension

Information demand patterns were presented as a way to capture the information demand of roles in organizations in a structured way to make them reusable. These collected information demands associated with tasks and responsibilities, should continue to be used now to offer a solution to reduce the problem of information overload while communicating via e-mail.

The aim is to support a person, who performs the defined tasks, through an automated provision of information at the time when the information is needed. The field of application of the concept can be a company or an organization, which uses e-mail as a communication medium and is suffering from information overload. The user will have the completion of his duties necessary e-mails in an appropriate manner at the appropriate time. Interpreting the information within the e-mail will remain with the user. The presence of an existing information demand pattern, describing the supported role, is adopted as a precondition for the application of the concept.

Information demand patterns are described so far in a semi-structured textual form. This form is neither readable nor interpretable by computers. Therefore, it is first necessary to make the information demand patterns machine-readable. It is proposed here to use the Extensible Markup Language (XML). The focus is here on a simpler subsequent adjustment of the structure, if needed. Later on it is possible to extend it in the direction of Web Ontology Language (OWL), if needed.

Within information demand patterns the specific information demands are only described linguistically and are not interpretable by computers. In the first step we added semantic by assigning keywords by hand. These keywords describe the information objects in order to make them identifiable. Due to the fact, that this manual procedure is time consuming, automated indexing will be tested. This may help later on to reduce the preparation time. The usage of online dictionaries to easier retrieve synonyms should be discussed as well.

Information has its highest value for business at the time it is needed. Information received late may be worthless. Obtaining it too early has low value either. The information must be presented to the user at the point in time when the user starts to fulfill the assigned tasks. This point emerges from the timeline out of the information demand pattern. The timeline provides the needed deadlines.

While identifying relevant e-mails not only incoming e-mails have to be considered, but also already existing e-mails. The e-mails can be labeled optionally, if a deadline comes closer, in order to attract the user’s attention.

At this point in time e-mails are assigned to the information demand. If the user wants to accomplish the task concerning the information demand, he/she has to access the needed e-mails. Therefore we have to discuss the distribution of the e-mails. We are of the opinion that it is important to interfere as little as possible with the user’s way of working [25]. The use of virtual folders provided by any modern e-mail application can be helpful to present the relevant e-mails to the user unobtrusively. It is to be noted that the relevant e-mails are presented to the user only. The user still coordinates and interprets the content. We aim at reducing the time spent for searching e-mails.

The inclusion of the proposed approach in the e-mail communication can be done locally by the user as well as on the side of the e-mail server. As an advantage of this implementation, broad support from any devices such as PCs, smartphones, Tablet PCs or Web access to e-mails is conceivable because filtering the content of e-mails happens centrally. The contrary is a higher cost to the implementation, as well as the possibility to extend the e-mail server with new software. The concept can be implemented as an extension of the used e-mail program as well. Modern e-mail programs offer interfaces to do this. As a disadvantage, it is here to note that, in this case, no support from other devices is feasible. Due to less expected effort we decided to implement a plug-in for the well-established e-mail client Mozilla Thunderbird. Later on we will port it to Microsoft Outlook and to the server-sided solution.

After the user has started the e-mail client, the plug-in is initialized as well. The start-up phase of the plug-in is straightforward. First the extension itself is loaded. If this is done, the information demand pattern is loaded. Keep in mind, that in order to benefit from the proposed economies of scale, the server in the intranet of the organization may provide the file containing the information demand pattern. After that the plug-in updates the virtual folders to present to the user the e-mails regarding their information demands. Figure 3 illustrates this behavior.

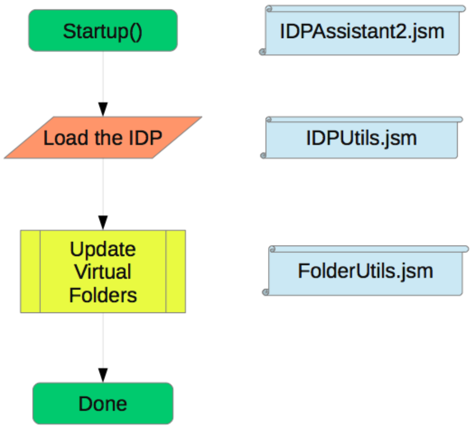

Figure 3. E-mail client plug-in startup sequence

The concept is offered as a way to reduce information overload within the e-mail communication in organizations. It can be expected that the automated provision of e-mails for the user will lead to saving time. As economic benefits, an increase in efficiency and an avoidance of wrong decisions can be stated.

An appropriate and previously specified information demand pattern can be viewed as a limitation of the proposed approach. Also, the collected information demand must be made sufficiently identifiable by keywords. It is likely that the quality of the presented e-mails to the users thereof will be significantly dependent on it. It is conceivable to transfer the duty to create the information demand pattern and the duty to determine keywords to central organization units. Economies of scale can be used if there are several similar roles in the organization.

3.3 Recommendation System Supporting E-Mail Client Extension

Goals of the recommendation system

Goals of the recommendation system:

- To adjust IDP-formed tasks based on actual worker's activity. Initially, all IDPs are created by some authorized entities and reflect general view on worker's specific role or tasks.
- To help to classify pieces of information (e-mail messages) as relevant/irrelevant to some IDP.

Informally, the goal of the recommendation system is to provide dynamic adjustments to the IDP-based structure of the information workflow.

In the context of this work, recommendation system analyzes the interaction between workers and e-mail messages within the scope of each IDP and adjusts the importance of e-mails in other workers' mailboxes based on this interaction.

Generalization and propagation of user-item relations is usually achieved through collaborative filtering which is the main principle of the proposed recommendation system. However, holistic approach to tailoring the IDP structure to the organization workflows goes beyond the traditional collaborative filtering scheme and involves variety of information processing techniques and models (Figure 4).

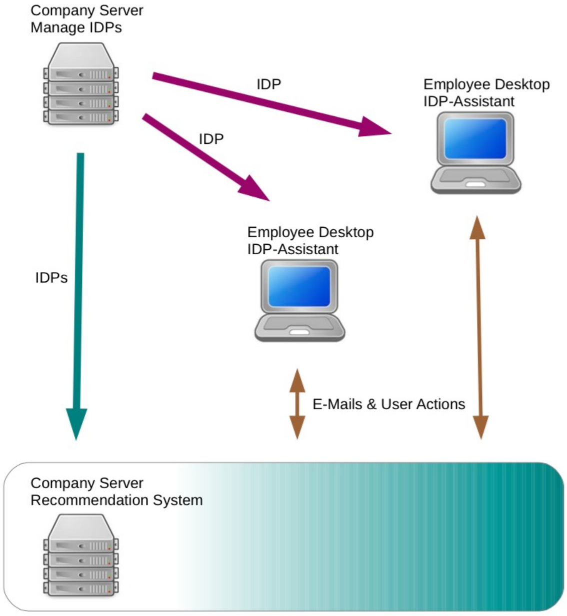

Figure 4. Technological perspective

Recommendation system input data

One of the crucial aspects of recommendation system development is identifying the input data useful for fulfilling the goal of the recommendation system. For each type of input data the rationale that underpins its usage for recommendations should be identified. Input data selection then affects mathematical models and algorithms that are used for making recommendations. For the proposed recommendation system the following types of input data are used:

- Textual content and additional attributes (message id, sender etc.) of e-mail messages;
- Workers' actions on e-mail messages;
- IDP descriptions;
- Workers' profiles.

Each type of input information is discussed in detail below.

Textual content and additional attributes of e-mail messages. To propagate the actions that a user applies to e-mail messages in his/her mailbox to other users' mailboxes the system must relate messages in different mailboxes. It can easily be done with multiple recipient messages as the sender program usually assigns message identifiers for outbound messages and these message identifiers will be the same for each recipient. The message identifier is put into the 'Message-ID' field of the message header (RFC 5322) and can be read by the receiver. Single recipient messages need different approach. So the similarity measure between e-mail messages is introduced for relating messages of different mailboxes. This similarity measure accounts not only for message contents, but also for supplementary message attributes (sender, list of receivers). Furthermore, the textual contents can be used to automate the message classification to IDPs.

Workers' actions on e-mail messages. These actions are interpreted as implicit information about how useful an e-mail message is for the given user in the particular IDP. Some examples for captured actions paired with their interpretation are listed below:

- The user deletes an e-mail message. It means that the e-mail message is irrelevant to this IDP and probably should be ranked lower for other users or even removed from the respective IDP-folders of other users.

- The user ignores (does not open) an e-mail message. It means that the e-mail message is likely to be relevant to this IDP but has been received too early.
- The user opens a low-ranked e-mail message before a high-ranked one. It means that there is a sign of ranking inversion and the opened e-mail probably should be ranked higher for other users.
- The user marks, flags and highlights an e-mail. It means that the e-mail message is important and should be ranked higher for other users.

IDP descriptions are employed to determine the scope of actions propagation. For example, if two workers perform similar tasks (and therefore, follow similar IDPs), there is a chance that they consider the same information as important. So, similarity measure between IDPs is introduced. The similarity measure accounts for information demands, their keywords and structural relations between organization units, performing respective IDPs.

Workers' profiles. The set of roles assigned to a worker and workers' efficiency measures that can be used to share the experience of highly professional employees.

Recommendation system output

Recommendation system produces an output in the form of expected importance of each e-mail message in the mailbox. The expected importance is then used to rank the e-mail messages presented to the user to make sure that the most relevant and actual e-mail messages are placed on the top of the e-mails list, attracting most of the worker's attention.

An important feature of the proposed recommendation system is that there are two principal components affecting e-mail ranking: (a) a set of rules from an IDP description, reflecting the information workflow design; (b) usage-based rules inferred from the practice of information processing by workers. During our research, these components are considered not reducible to one another as they aim at different goals: conceptual description of the IDP vs. tailoring this IDP to actual information workflows and informal information processing patterns of an organization. Therefore, the recommendation system has to merge e-mail rankings produced by either of these components.

As it is shown in Figure 5, recommendation system functionality comprises four blocks discussed in the rest of the current section.

Technological model

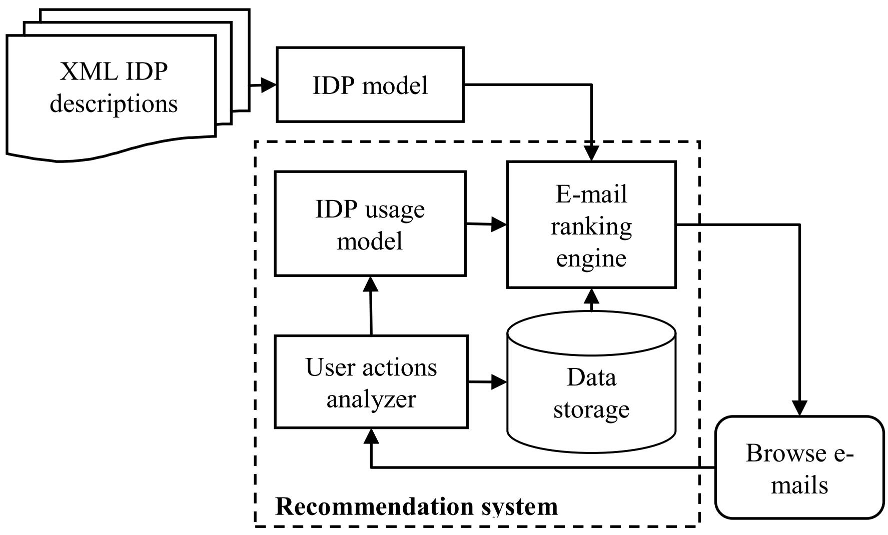

Figure 5. Recommendation system in the context of e-mail client with IDP support

First of all, entire IDP model is divided into two parts: static (or, IDP model) and dynamic (or, IDP usage model). The static part is enforced by organization information workflow engineering and is composed of human-generated rules about how to classify an e-mail to IDP and how to assign importance to e-mails inside some IDP. This part is knowledge-based and pieces of knowledge here are classification rules connecting the message attributes (sender), keywords, and optionally facts, extracted from the message by the context-free pattern analysis algorithm [26], [27], with IDPs and current message importance according to some IDP.

The dynamic part, or *IDP usage model*, is employed to adjust base IDP model to actual information workflows of the organization. This part also has the form of a classifier but it is learned from workers' interaction with e-mails by some machine-learning algorithm.

User actions analyzer. The e-mail client tracks the user actions and passes their descriptions to this component of recommendation system. Each action description includes: action type (removing, flagging, opening an e-mail etc.), IDP instance, user, action time, message browsing context (identifiers of other e-mails that are ranked higher in current users' browsing context). Actions data are used to estimate current e-mail importance in the context of the given IDP. These estimations are saved in recommendation system data storage. Furthermore, user actions and estimated importance are used to build the adjusted IDP models by training classifier that predicts IDP from the message text and attributes.

E-mail ranking engine. It is hybridization of knowledge-based and collaborative approaches. The knowledge-based part is powered by the IDP description provided by the knowledge worker and the processed form of the e-mail message. The collaborative engine looks for similar users and then for each e-mail retrieves estimated importance of similar e-mails from the recommendation system data storage. The knowledge-based and collaborative lists are then merged to provide user with a resulting message list.

3.4 Interface Description

This section describes the interface of the recommendation system, which is used for the communication between the locally installed e-mail plug-in and the recommendation system.

The recommendation system provides an application interface according to the Representational State Transfer programming paradigm (REST) with communication over the Hypertext Transfer Protocol (HTTP) protocol and the encoding in the JavaScript Object Notation (JSON).

There are several reasons for choosing this setting. The concepts are widely used and are actually de-facto standard for web application programming interfaces. This leads to a wide range of available supporting software for servers, clients and libraries. The Hypertext Transfer Protocol is allowed in many networks. This may lead to a broader usage of the implementation of our concept. SSL can easily secure HTTP (HTTPS) and respects the need for security of enterprises and organizations. Due to the fact that JSON libraries are available for almost every programming language, later modifications can be done easily. Another important point is the possibility to push data from the server to the clients. As already described, our e-mail-extension is implemented at the moment for Mozilla Thunderbird, but one of the next steps will be the porting to Microsoft Outlook to support a wide range of modern e-mail-clients.

The recommendation system will support the following operations:

- *POST /idps*: tells the recommendation system about the information demand pattern used in the single e-mail-client-extension. This includes attributes like a pattern identifier (id), the name of the pattern (name) and the information demands (demands). Information demands have attributes. Namely there are the demand attributes 'name', 'dueDate', 'priority' and 'keywords'. The attribute 'name' and 'dueDate' are obvious and indicate the name of the information demand and the last date of delivery. Derived from the quality criteria of each information demand the ‘priority’ attribute represents the ‘general importance’ of the information demand. The ‘keywords’ attribute helps to identify the information demand within an e-mail.

- *GET /idps/<id>*: requests information about an IDP from the recommendation system.
- *POST /mails*: tells the recommendation system about an e-mail. The data structure covers various attributes, e.g., ‘id’, ‘priority’, ‘date’, ‘subject’ or ‘body’. As with information demand pattern the e-mail plug-in does not assign an ‘id’ to the e-mail. This is done by the recommendation system which checks if the e-mail is already known or not, because another e-mail extension in the enterprise may have reported the e-mail before.
- *GET /mails/<id>*: allows a request by the client about an e-mail.
- *PUT /mails/<id>*: tells the recommendation system about the user interaction.
- *GET /users/<user_id>*/events: opens the event-stream for pushing recommendation data to the client.

After the description of the interface of the recommendation system and some descriptions about the used data structures, we examine now the process of operation. On system start up some initialization operations take place. Figure 6 shows these operations using a sequence diagram. The e-mail client extension first has to register itself by sending the used information demand pattern to the recommendation system. All e-mails, which are unknown to the recommendation system, will be sent as well. The extension stores information about the communication in the history to be able to identify the e-mails, which are still unknown to the recommendation system. This avoids communication overhead.

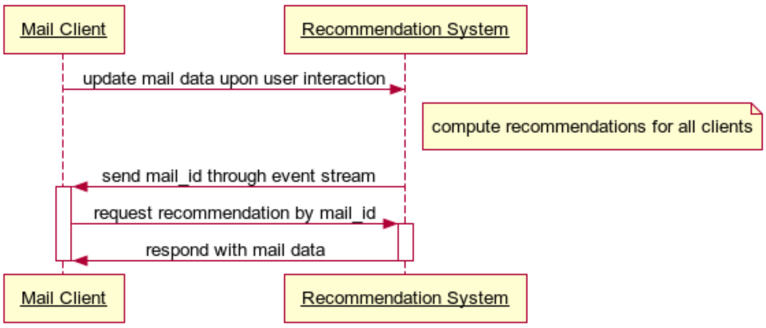

Figure 6. Initialization of communication

During normal operation the e-mail client updates mail data upon user interactions. If a user, for example, deletes a message, the e-mail client extension sends a message to the recommendation system including the ‘id’ of the e-mail message and the performed operation on that e-mail. The recommendation system computes the derived recommendations for all attached clients. Then the recommendation system opens the event stream to the relevant clients and informs the clients about the new recommendation by sending the e-mail ‘id’ to the clients. The clients will then request the recommendation by this e-mail ‘id’. Finally the recommendation system will respond with the requested recommendations. Figure 7 illustrates this in a sequence diagram.

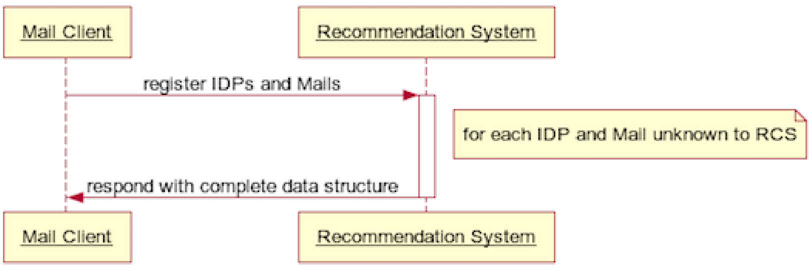

Figure 7. Normal operation

4 Summary and Conclusion

Due to the increasing amount of e-mails received every day and the resulting information overflow, this paper proposes a conceptual architecture for enterprises and organizations to support demand-oriented information supply. Well-established information demand patterns are used. Information demand patterns themselves are the results from the information demand analysis, which leads to the information demands needed by a worker in an enterprise to accomplish his/her tasks. Information demand patterns are used to feed the proposed e-mail client plug-in, which enables to provide the needed e-mails to the user just in time by filtering and presenting them in an appropriate manner. Due to the fact that this e-mail client extension is used just locally at the work place of one user, we extended the solution by a recommendation system. The recommendation system monitors users' behavior like deleting, reordering or ignoring messages and proposes the results to other users with a similar information demand. This might reduce negative consequences of information overload like reduced efficiency, wrong decisions and excessive demands of the employees.

The proposed solution has mainly two advantages for an organizational application. Firstly, our extension of an email client is able to filter both new and existing e-mails analog of given information demand and provides them appropriate to the user. For this purpose, we use information demand pattern, which offers a well-established approach to encapsulate the information demand and make it reusable. Secondly, the proposed overall concept is able through the use of the recommender system to observe the interactions by users with their emails. Therefore the system is able to benefit from a user's attitude towards an e-mail, which leads to better recommendation for other users with similar information demand.

The biggest shortcoming of our approach so far is that it has not been fully implemented and validated in practice. The core elements of the proposed architecture, the recommendation system and the plug-in for the e-mail client including transformation of textual IDP, were both implemented and evaluated in practical application, but separately from each other (see, e.g., [10] and [28]). The integration of both into a joint system so far mainly happened on a conceptual level. From this perspective, we presented the work in progress, which has to be continued technically and conceptually.

From a technical perspective, the implementation of the proposed architecture in a collaborative recommendation system with IDP use and e-mail frontend has to be finished beyond the presented interface descriptions, data structures and sequence diagrams. Since the interfaces of both components are well known and suitable for integration, we expect this to cause substantial efforts but no principal problems. The configuration of the system for the actual use in an organization using different IDPs will probably create additional insights regarding the need for further automation. Even for a human actor, to identify overlapping information demands between different IDPs sometimes is not straightforward since the vocabulary used in different IDPs is not necessarily fully adjusted. Automation of this mapping might require techniques from text matching and ontology matching [29].

From a conceptual perspective, we plan to investigate the utility and value of the collaborative recommendation system. For this purpose we need a model how to measure or at least estimate the value and a set-up for performing measurements in everyday practice. Regarding the model for measuring the value, we intend to use our experience in balanced scorecards [30] and develop a specific scorecard for this purpose. The set-up for practical evaluation will in the first step probably be a team at a university and the demand of the team members regarding information about education and research activities of the team. Later on, we intend to extend this to an industrial setting.

Acknowledgements

The work presented in this paper was supported within the project KOSMOS (Konstruktion und Organisation eines Studiums in offenen Systemen) funded by the BMBF (Federal Ministry of Education and Research, Germany) and the European Social Funds of the European Union.

The research was also supported partly by projects funded by grants 13-07-00271, 13-07-00039, 13-07-12095, 13-07-13159 of the Russian Foundation for Basic Research, project 213 (program 15) of the Presidium of the Russian Academy of Sciences, and project #2.2 of the basic research program "Intelligent information technologies, system analysis and automation" of the Nanotechnology and Information Technology Department of the Russian Academy of Sciences. This work was also partially financially supported by the Government of Russian Federation, Grant 074-U01.

References

[1] D. Stamer, A. Ponomarev, K. Sandkuhl, N. Shilov, and A. Smirnov, "Collaborative Recommendation System for Improved Information Logistics: Adaption of Information Demand Pattern in E-Mail Communication," presented at the Proc. 7th International Workshop on Information Logistics and Knowledge Supply co-located with th International Conference on Perspectives in Business Informatics Research BIR, Lund, Sweden, 2014, pp. 35–48.

[2] M. Castells, *The Rise of the Network Society*. John Wiley & Sons, 2011.

[3] P. Melinat, T. Kreuzkam, and D. Stamer, "Information Overload: A Systematic Literature Review," presented at the 13th International Conference on Perspectives in Business Informatics Research, Lund, Sweden, 2014, vol. 194, pp. 72–86. http://dx.doi.org/10.1007/978-3-319-11370-8_6

[4] A. Edmunds and A. Morris, "The problem of information overload in business organisations: a review of the literature," *International Journal of Information Management: The Journal for Information Professionals*, vol. 20, no. 1, pp. 17–28, Feb. 2000. http://dx.doi.org/10.1016/S0268-4012(99)00051-1

[5] K. Sandkuhl, "Information Logistics in Networked Organizations: Selected Concepts and Applications," presented at the 9th International Conference on Enterprise Information Systems, Madeira, 2009, pp. 43–54. http://dx.doi.org/10.1007/978-3-540-88710-2_4

[6] A. Öhgren and K. Sandkuhl, "Information Overload in Industrial Enterprises - Results of an Empirical Investigation," presented at the Proc. ECIME 2008, London, 2008.

[7] C. Speier, J. S. Valacich, and I. Vessey, "The influence of task interruption on individual decision making: An information overload perspective," *Decision Sciences*, vol. 30, no. 2, pp. 337–360, 1999. http://dx.doi.org/10.1111/j.1540-5915.1999.tb01613.x

[8] M. Volnhals and B. Hirsch, "Information overload und controlling," *Controlling & Management*, vol. 52, pp. 50–57, 2008. http://dx.doi.org/10.1365/s12176-012-0189-9

[9] R. Soucek and K. Moser, "Coping with information overload in email communication: Evaluation of a training intervention," *Computers in Human Behavior*, vol. 26, no. 6, pp. 9–9, Oct. 2010. http://dx.doi.org/10.1016/j.chb.2010.04.024

[10] D. Stamer, "Reduktion der Informationsüberflutung durch Informationsbedarfsmuster in der E-Mail-Kommunikation," presented at the Multikonferenz Wirtschaftsinformatik 2014, Paderborn, 2014, pp. 1226–1233.

[11] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, Sep. 2012. http://dx.doi.org/10.1016/j.eswa.2012.02.038

[12] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, Jul. 2013. http://dx.doi.org/10.1016/j.knosys.2013.03.012

[13] R. Burke and M. Ramezani, "Matching Recommendation Technologies and Domains," Springer US, 2011, pp. 367–386. http://dx.doi.org/10.1007/978-0-387-85820-3_11

[14] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, no. 3, Boston, MA: Springer US, 2011, pp. 73–105. http://dx.doi.org/10.1007/978-0-387-85820-3_3

[15] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," vol. 2009, p. 4, Jan. 2009. http://dx.doi.org/10.1155/2009/421425

[16] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information systems*, 2000.

[17] F. Lorenzi and R. Ricci, "Case-based recommender systems: A unifying view," *Current Trends in Database*

Technology - Edbt 2004 Workshops, Proceedings, vol. 3169, no. 5, pp. 89–113, 2005. http://dx.doi.org/10.1007/11577935_5
[18] A. Felfernig and R. Burke, "Constraint-based recommender systems: technologies and research issues," presented at the ICEC '08: Proc.10th international conference on Electronic commerce, New York, New York, USA, 2008, p. 1. http://dx.doi.org/10.1145/1409540.1409544
[19] M. Lundqvist, E. Holmquist, K. Sandkuhl, U. Seigerroth, and J. Strandesjo, "Information Demand Context Modelling for Improved Information Flow: Experiences and Practices," presented at the The Practice of Enterprise Modeling, Stockholm, 2009. LNBIP Volume 39, 2009, pp 8-22. http://dx.doi.org/10.1007/978-3-642-05352-8_3
[20] K. Sandkuhl, "Information Demand Patterns," presented at the PATTERNS 2011, The Third International Conferences on Pervasive Patterns and Applications, Rome, 2011, pp. 1–6.
[21] K. Sandkuhl and D. Stamer, "Towards Improving Structure and Content of Information Demand Patterns," presented at the Perspectives in Business Informatics Research - 11th International Conference, BIR 2012, Nizhny Novgorod, Russia, September 24-26, 2012. Satellite Workshops and Doctoral Consortium, Nizhny Novgorod, 2012.
[22] L. Heinermann, D. Stamer, and K. Sandkuhl, "Usability Evaluation of Method Handbook," presented at the 6th International Workshop on Information Logistics, Knowledge Supply and Ontologies in Information Systems (ILOG), Warsaw, 2013, 1st ed., pp. 11–22.
[23] K. Sandkuhl, "Improving Engineering Change Management with Information Demand Patterns," presented at the 8th International Conference on Poduct Lifecycle Management, 2011.
[24] K. Sandkuhl, "Validation and Use of Information Demand Patterns in Higher Education," presented at the Business Information Processing, Berlin, Heidelberg, 2010, vol. 57, no. 26, pp. 204–213. http://dx.doi.org/10.1007/978-3-642-15402-7_26
[25] F. R. H. Zijlstra, R. A. Roe, A. B. Leonora, and I. Krediet, "Temporal factors in mental work: Effects of interrupted activities," *Journal of Occupational and Organizational Psychology*, vol. 72, no. 2, pp. 163–185, 1999. http://dx.doi.org/10.1348/096317999166581
[26] M. Tomita, *Efficient Parsing for Natural Language*. Norwell: Kluwer Academic Publishers, 1986. http://dx.doi.org/10.1007/978-1-4757-1885-0
[27] (2014, November 28). "Yandex free tool for fact elicitation." [Online]. Available: https://tech.yandex.ru/tomita/
[28] A. Smirnov, A. Kashevnik, A. Ponomarev, N. Shilov, M. Schekotov, and N. Teslya, "Recommendation System for Tourist Attraction Information Service," presented at the Open Innovations Association (FRUCT), 2013 14th Conference of, 2013, pp. 148–155. http://dx.doi.org/10.1109/FRUCT.2013.6737957
[29] F. Lin and K. Sandkuhl, "A Survey of Exploiting WordNet in Ontology Matching," in Artificial Intelligence in Theory and Practice II, vol. 276, no. 33, Boston, MA: Springer US, 2008, pp. 341–350. http://dx.doi.org/10.1007/978-0-387-09695-7_33
[30] R. S. Kaplan and D. P. Norton, The Balanced Scorecard: Translating Strategy into Action. Harvard: Harvard Business Press, 1996.