International Journal of Information Management 35 (2015) 662–671

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

On the relevance of reports—Integrating an automated archiving component into a business intelligence system

Michael Schulz*, Patrick Winter, Sang-Kyu Thomas Choi

School of Business Administration & Economics, Philipps University at Marburg, 35037 Marburg, Germany

ARTICLE INFO

Article history:
Received 10 February 2015
Received in revised form 26 June 2015
Accepted 21 July 2015
Available online 11 August 2015

Keywords:
Archiving
Business intelligence (BI)
Operational BI
Information storage

ABSTRACT

In the last years, the scope of business intelligence (BI) systems has been extended from strategic to operational decision support (operational BI). This has led to an increase in the number of information needs and, at the same time, to a decrease in the "efficiency" of reports in terms of how many information needs they address. As a consequence, the number of reports has exploded. This slows down knowledge workers' manual or automated search for information, resulting in high search costs to companies. However, it can be observed that in many cases only a small subset of all reports is (still) relevant to knowledge workers. The remainder is an unnecessary burden that could be sorted out without obstructing the access to information that still is needed. In this paper, we develop a framework to identify such reports and archive them automatically. The relevance of reports is concluded from users' information retrieval behavior as recorded in the log files of the BI system, particularly of its search component. We evaluate the proposed framework through a simulation study. The results indicate that the integration of an automated archiving component into a BI system can significantly reduce search effort and, hence, search costs.

1. Introduction

It is long understood that a company's competitiveness largely depends on how *effectively* it can make use of information (e.g., Menon & Varadarajan, 1992). Due to an increasing volume and variety of data available for analysis on one hand and changes in the audience of information systems (IS) on the other hand, however, the number of information resources stored in many IS has significantly increased in the last years. As a consequence, it has become difficult for knowledge workers to locate relevant information in reasonable time (that is, *efficiently*) because distinguishing between relevant and irrelevant information resources takes (too) long (e.g., Davenport & Beck, 2000). Enabling them to do so, therefore, is a major challenge to modern information management (IM).

Not meeting this challenge can have severe negative consequences for companies. In the extreme case, commonly referred to as information overload (see (Edmunds & Morris, 2000) for a review), it may prevent the effective use of information and, hence, significantly weaken competitiveness. Putting the danger of this happening aside, increasing search effort may cause knowledge workers to base their working process on only a subset of all available information resources that they can look through in a given amount of time (e.g., to the end of a deadline), leading to results of lower quality (Chewning & Harrell, 1990; Hwang & Lin, 1999; O'Reilly, 1982). In order to avoid this, they may also try to still find all relevant information, thereby spending a lot of time that they could have spent for other tasks, causing opportunity costs to the company (Cleverley & Burnett, 2015; Haas & Hansen, 2007). The same applies if they recreate existing information resources that they do not find. The impact this has on business, commonly referred to as search costs, has been quantified by the International Data Corporation (IDC), a market research firm specialized in IT (Feldman & Sherman, 2003): "*Using the scenarios outlined above, IDC estimates that an enterprise employing* 1000 *knowledge workers wastes at least* $*2.5 to* $*3.5 million per year searching for nonexistent information, failing to find existing information, or recreating information that can't be found. The opportunity cost to the enterprise is even greater, with potential additional revenue exceeding* $*15 million*

* Corresponding author. Fax: +49 64212826554.
E-mail addresses: michael.schulz@wiwi.uni-marburg.de (M. Schulz), patrick.winter@wiwi.uni-marburg.de (P. Winter), thomas.choi@wiwi.uni-marburg.de (S.-K.T. Choi).

http://dx.doi.org/10.1016/j.ijinfomgt.2015.07.005
0268-4012/

annually" (p. 9). This was over ten years ago; more recent studies (e.g., Schubmehl & Vesset, 2014) give even higher cost estimates.

When considering the general case, approaching the challenge of reducing search effort is difficult. This is because across different IS, usually both, information resources *and* user groups, are heterogeneous, even within one company. Therefore, it is hard to find general patterns of the formers' relevance to the latter. For this reason, we restrict ourselves to a special case in this paper, the case of business intelligence (BI) systems. With these, the company's data are analyzed to derive information that can be used for decision support later on. The results of these analyzes are saved in reports, which are the only and, thus, comparatively homogeneous information resources that exist in BI systems (e.g., Golfarelli, Rizzi, & Cella, 2004). Reports after their creation remain in the system, so that they can be accessed later on by all knowledge workers (who have the necessary rights). The major drawback associated with this process is that while new reports are steadily created, such that are not relevant anymore usually do not get deleted. Therefore, the total number of reports increases over time.

In the past, this has not been a major problem because BI systems back then were almost exclusively used for strategic purposes (Herring, 1988), for which the total number of reports required is relatively low. This has changed in the last years since numerous companies have started to employ BI systems also for operational purposes (operational BI) (White, 2005). As we will elaborate on in more detail later, this has increased the total number of reports stored dramatically – e.g., in a case reported in (Eckerson, 2008) from 1,400 to 4,000 within only one year. This can lead to the aforementioned consequences if no techniques to reduce search effort are employed. Therefore, there is an acute need in BI to develop and introduce such techniques, which is why we focus particularly on this field in this paper.

When aiming to reduce the total number of (active) reports, care must be taken to not obstruct the access to information that still is needed. A natural approach to do so is to sort out reports that have become irrelevant. The difficulty in this, however, lies in discerning these from the remainder. In this paper, we develop a technique to do so automatically. More concretely, we investigate the integration of an archiving component into a BI system that identifies and archives reports based on the information retrieval (IR) behavior of its users (as recorded in its log files). By this, we transfer the concept of archiving from the level of data (Inmon, 2010) to the level of information resources, constituting an information storage (IST)-based approach to IM. We propose a framework for archiving that consists of four parts: which elements an archiving component should have, which types of relevance patterns reports can exhibit, which indicators can be used to infer their relevance patterns, and how the archiving component needs to interact with the BI system's other components. We evaluate our framework through a simulation study.

We structure this paper as follows: in Section 2, we elaborate in more detail on the historical development of BI and how it has affected the report portfolio. We further briefly review and discuss some alternative approaches to reduce search effort. In Sections 3 and 4, we present our archiving framework and the simulation study to evaluate it, respectively. Section 5 concludes this paper with an outlook for further research.

2. Background

2.1. Historical development of BI and consequences

Companies have employed IS to support their business processes for many years now. While the data stored in these systems are recorded for operational use, it soon has been recognized that they also provide a valuable basis for decision support (Sprague, 1980). For this purpose, they are extracted from the operational IS, transformed, and loaded into analytical IS (Moore & Chang, 1980). The latter often are tailored to certain user groups or certain purposes, which is why they exhibit various functionalities and appear under various labels (e.g., "management IS", "expert systems", etc.). In the 1980s, the more general term "business intelligence" became popular (e.g., Gilad & Gilad, 1988). We use this term in this paper to emphasize the goal of deriving information from data, regardless of what happens with this information later on. Nevertheless, the common understanding of BI was still such that its primary application was strategic decision making and its primary audience, therefore, the top and middle management (Hannula & Pirttimäki, 2003; Herring, 1988).

Enabled and, as some may argue, driven by technological progress, the scope of BI systems has been extended in the last decade. The possibility to store and analyze large amounts of data in reasonable time (e.g., through in-memory databases) has motivated companies to base no longer only strategic but also operational decisions on data. While this in the beginning has promised competitive advantages (Marjanovic, 2007), it today has become a necessity to avoid competitive disadvantages (Nadj, Morana, & Maedche, 2015). Doing so within operational IS is difficult, however, because these cannot simply be put on hold for analysis and, further, usually lack the necessary analytical functionalities (such as, e.g., historization of data). As BI systems are separate from operational business and provide these functionalities, it is not surprising that they soon were employed for this purpose (e.g., Marjanovic, 2007), constituting operational BI. As a consequence, the number of reports stored in these systems has increased dramatically, as mentioned earlier. This is essentially due to the following two reasons:

First, an increase in the number of information needs (INs) to be fulfilled with the aid of BI systems (Böhringer, Gluchowski, Kurze, & Schieder, 2009), which on one hand simply results from a lot more knowledge workers being concerned with operational decisions than with strategic decisions. On the other hand, a lot more and more heterogeneous data have to be stored for operational decision support (in particular, disaggregated data). This is amplified by the availability of new data sources (such as, e.g., sensor networks). Once these data are stored in the system, it is likely that they will be analyzed sometime out of curiosity, bringing new INs into being. Because a report can address only a few INs (often just one), many new reports have to be created to fulfill all of them.

Second, a decrease in the "efficiency" of reports, that is, in the ratio between their number and the number of INs they are suited to fulfill. This is caused by a new practice of granting all knowledge workers access to BI systems, so that they can create reports by themselves (self-service BI, SSBI) (Imhoff & White, 2011) instead of having to wait for experts to create them. By introducing this practice, companies have reacted to the observation that the traditional way of supplying knowledge workers with information through the IT is too time-consuming to be efficient and too slow to be effective in supporting heterogeneous decisions as they occur in operational BI (Böhringer et al., 2009). This is particularly true because for supporting operational decisions, INs often have to be fulfilled in (near) real-time (Işik, Jones, & Sidorova, 2013). The problem associated with SSBI is that the new audience of BI systems contains a lot of users with a low expertise in BI. These may not be aware of or fully comprehend the existing reports and, therefore, create new reports to fulfill INs that also could have been fulfilled using the existing ones. Furthermore, they foremostly create so-

called ad hoc reports (Mills, Davis, & Bluhm, 2012).[1] These address a single acute IN but can usually not be generalized (e.g., because fixed values instead of variable parameters are used when setting filters). Therefore, they seldom can be reused, so that new ad hoc reports are created steadily. An extreme example case of this can be found in (Eckerson, 2008): in a BI system with 450 users, 26,000 reports were available after some years of SSBI while 300 (1.2%) would have sufficed to fulfill nearly the same INs.

2.2. Previous approaches to reduce search effort

IR and IST are often subsumed under the term "information storage and retrieval" (see (Hjørland, 2015) for the historical development of this view). However, there is a clear distinction (e.g., Cooper, 1971): while IR is related to an IN (as explicated, e.g., through a textual search query), IST is, in principal, independent of these. Therefore, approaches to IM can be distinguished by whether they relate to IR or IST (or both).

Against this background, it is surprising that previous research has mainly focused on IR-based approaches to reduce search effort by developing more and more elaborate search engines and algorithms (e.g., Engler, Schulz, & Winter, 2014; Hawking, 2004; Kulkarni & Callan, 2015; Ronen et al., 2009). Putting aside the usual disadvantages of these (such as, e.g., users often being unable to formulate accurate search queries (Ruthven & Lalmas, 2003)), this is the approach of choice if all information resources in an IS can be assumed to, generally, be still of relevance and the problem is "just" to decide which of them should be presented to a user with a specific IN (and in which order). However, for the case of BI systems (and some other IS), the problem is different. As illustrated by the cases mentioned above, a large share of the stored reports lacks any future relevance and, therefore, is just an unnecessary burden. Sorting these reports out is an IST-based approach; after having done so, IR-based approaches can be applied to the remainder. In this sense, our approach is complementary to IR.

For IST-based approaches, on the other hand, it is necessary to estimate the future relevance of information resources. This can be tried manually for some time but as soon as their total number exceeds a certain critical value, the effort to do so regularly becomes disproportionate.[2] Furthermore, one for this purpose would be required to be familiar with all INs of all system users, what obviously is unrealistic.

Therefore, automated methods need to be found that are based on objective relevance indicators. The quality of these indicators, of course, depends on the data on which they are based. When taking into account only static (meta) data on information resources, only very limited information on their future relevance can be derived. An example of such an approach is sorting out reports with similar content from a BI system (Hsu & Li, 2011). These can be considered irrelevant as long as the original reports exist. However, irrelevant reports are not affected by this approach if they are unique (what is typical for ad hoc reports).

Since the relevance of information resources depends on users' INs, which are reflected by the users' IR behavior, it seems more suitable to base relevance indicators on data on the latter. Thereby, a connection between IST and IR is established.

[1] This is partly because many SSBI systems are designed in such a way that they aid users especially in creating ad hoc reports. Capterra (www.capterra.com), a website to support companies in selecting software appropriate for their specificities, lists 264 different BI software products, of which 110 (41.7%) provide functionalities for ad hoc reporting.

[2] An example case reported in (Wixom & Watson, 2010, p. 202) suggests that one person can process only ca. 100 reports per day.

2.3. Concept of relevance

Throughout this paper, we often make use of the term "relevance". As this concept is not used consistently in literature (and cannot be, as will become clear in the following), it is helpful to briefly clarify our understanding of it. For this purpose, it again has to be distinguished between IR and IST, as the meaning of "relevance" slightly differs between these contexts.

IR happens when a user searches for information resources in order to fulfill an IN. We call an information resource relevant if it, in the perception of the user, is suited to fulfill this IN, and irrelevant otherwise. Therefore, relevance in our view is (1) relative to a user, (2) relative to an IN, (3) dependent on the user's perception, and (4) qualitative (either the IN is fulfilled or not). Furthermore, it is (5) static, because we will consider it only at a fixed point in time, so that the user's perception is also fixed. There is a dependency between these aspects; for a thorough discussion of this dependency and alternative concepts of relevance, we refer to (Borlund, 2003) as IR-based or individual relevance is only indirectly related to our approach.

For IST-based approaches such as archiving, the relevance of an information resource needs to be assessed differently. Essentially, it is obtained by aggregating the individual relevances described above (1) across all users and (2) across all INs. It still is (3) dependent on user perceptions because the individual relevances are as well. But in contrast to these, it is (4) quantitative, with its level being determined by the number of users and INs for which the information resource is individually relevant. However, we will map this quantity again to a qualitative assessment of relevance or irrelevance. Finally, relevance here is (5) dynamic because it is considered for multiple points in time (and thus, over a period of time), during which it may change because of changes in the number of related INs. This is the key idea that makes archiving possible, as only information resources that have become irrelevant (over time) can safely be archived.

3. Archiving framework

3.1. Elements of an archiving component

An archiving component of a BI system should contain at least four elements:

1. *decision rules* to determine which reports should be archived or reactivated,
2. a *screening mechanism* that applies these decision rules,
3. an *execution rule* that specifies when the screening mechanism is to be executed, and
4. a *physical mechanism* that carries out archiving and reactivation physically.

The decision rules are crucial. By them, reports that are likely to be relevant in the future are distinguished from such that are likely to be irrelevant. Therefore, the difficulty lies in predicting a report's future relevance. Since the latter depends on a company's specificities (e.g., the number of employees), however, concrete decision rules cannot be part of a universal framework. Instead, we in the following elaborate on how they can be derived.

The screening mechanism links the other elements. For most companies and BI systems, a simple mechanism like the following suffices:

```
Program screenReports()
  For every report r that is active
    Decide if r should be archived using the decision rules
    If so,
      Archive r according to the physical mechanism
    End
  End
  For every report r that is archived
    Decide if r should be reactivated using the decision rules
    If so,
      Reactivate r according to the physical mechanism
    End
  End
End.
```

An important characteristic of this mechanism is that it screens all a active and all b archived reports during its execution, so that its running time is in $O(a+b)$. We remark that other mechanisms that screen only a subset of all reports could be more efficient in terms of running time; however, when employing them, it has to be decided which subset should be screened, so that the problem is often just shifted.

Whether considerations on running time carry weight at all also depends on the execution rule. It may be sufficient to use a time-based rule, that is, to screen the reports periodically in certain time intervals (e.g., months). Alternatively, an event-based rule can be employed that ties screening to the occurrence of certain events. These can be foreseeable (e.g., holidays) or unforeseeable (e.g., the total number of reports exceeding a certain value).

The physical mechanism is specific to the infrastructure of the examined BI system. E.g., when a directory structure is used to navigate through reports, archiving and reactivation can happen through moving them to or from a specific folder (comparable to the "recycle bin" of many operating systems). Alternatively, an attribute can be introduced for each report that describes its current archiving status (e.g., active vs. archived). Archived reports can then, say, be displayed "grayed out". When choosing the physical mechanism, one has to pay attention that the possibility to access archived reports remains for users as well as for the BI system, so that a basis for potential reactivation is given.

It should be noted that, despite being automated in principle, an archiving component with the aforementioned elements also allows for manual interventions: First, users (who have the necessary rights) can set specific decision rules to inhibit the archiving or reactivation of certain reports from the start. Second, reports that have been automatically archived (reactivated) can manually be reactivated (archived again) by carrying out the physical mechanism.

3.2. Relevance pattern types of reports

In order to estimate the future relevance of a report for choosing the decision rules, it is helpful to first investigate how the relevance of a report changes over time in dependence on its type. Prior research (e.g., Maciariello, 1984, p. 30, Chapter 2; Seidel, Knackstedt, & Janiesch, 2006; Switzer, 1994) has identified three common types of reports: routine, ad hoc, and exception reports. From these report types, corresponding relevance pattern types can be derived. This is done in Table 1, where both are discussed jointly. The relevance pattern types are illustrated in Fig. 1.

It is noteworthy that a report's relevance pattern often is largely determined by its time reference as indicated in Table 1. If this information can be assessed automatically, it can be used in addition to the relevance indicators presented hereafter.

3.3. Relevance indicators

To be able to capture the relevance pattern of a report, indicators have to be defined that reflect its relevance. A measure that intuitively seems to be suitable for this purpose is the *number of accesses* $A(t)$, that is, how often the report has been accessed in total up to a point in time t. Since it is cumulative, one can consider normalizing it in order to enable a fair comparison between older and younger reports. For this purpose, it can, e.g., be divided by the length of the time period between the report's creation to t. Alternatively, one can use the *increment of accesses* compared to the previous point in time, $a(t) = A(t) - A(t-1)$. In most current BI systems, however, $A(t)$ is stored only as a simple attribute that is incremented and, thus, overwritten each time a report is accessed. Even if this is changed during the integration of an archiving component in such a way that the history of $A(t)$ is from that point on recorded, relevance patterns of reports that have existed beforehand can still only be partially reconstructed using, e.g., the *date of the last change*. For this reason, other sources for deriving relevance indicators have to be found.

Such a source can be the search component of the BI system (or, depending on the company's information infrastructure, a company-wide search engine). The *number of searches* $S(t)$ that have resulted in a certain report up to t and the derived *increase of searches* $s(t) = S(t) - S(t-1)$ can serve as indicators for its historical

Table 1
Types of reports and corresponding relevance pattern types.

Report type	Relevance pattern type	Description	Typical examples	Suitability for archiving
Routine	Continuous	Reports with a continuous pattern type usually become relevant at the time of their creation, but it can take a while until all users realize this. This may be, e.g., because the search component of the BI system has to index reports first before it can find them. Afterwards, their relevance settles around a constant value, to which it always returns in the long run, albeit it can depart upwards or downwards due to, e.g., seasonal effects.	Typically, reports with a continuous relevance pattern are used without major interruptions and have a relative and fine-grained time reference (e.g., up-to-date product sales) or no time reference at all.	Reports that have a continuous relevance pattern can only be archived safely if even their maximum relevance level is very low or close to zero.
	Periodic	The relevance of reports with a periodic relevance pattern type is linked to periodically recurring events. Dependent upon the type of this event, it oscillates shortly before, during, and/or shortly after its occurrence. Between two events, it exhibits a period of irrelevance that is usually longer than the period of relevance.	Typical examples are reports with a relative and coarse-grained time reference. Often these are reports that provide regular information on the bygone period (e.g., aggregated monthly product sales, whereby the triggering event is the end of the month).	While reports with a periodic relevance pattern are not suited for archiving, they are most prone to being falsely archived. This is because their transient irrelevance is hard to distinguish from permanent irrelevance.
Ad hoc	Ad hoc	By definition, ad hoc reports become relevant immediately after their creation. However, they usually are relevant only to a few users (often just one) and only for a short time – until a concrete IN is fulfilled. After this has happened, they in most cases immediately lose their raison d'être and, from then on, exhibit a very low relevance close to zero.	Ad hoc reports usually are created to fulfill a certain acute IN (e.g., "why were sales slowing down in Germany in 2014?"), which sometimes may be exotic, so that it cannot be fulfilled by routine reports. They often exhibit an absolute time reference.	Reports with an ad hoc relevance pattern that have already lost their raison d'être are particularly suited for archiving because it can be expected that they never become relevant again.
Exception	Unpredictable	Exception reports are created when a potential threat is detected. However, they become relevant not until then a corresponding incident occurs. Since this cannot be foreseen, they exhibit an unpredictable relevance pattern. In fact, it is also possible that they never become relevant – if no incident occurs. Once an incident occurs, however, they immediately are of high relevance (although it may take a while until all users realize this) as long as it has not been resolved. The level of relevance depends on the threat and the number of users affected.	Typical examples of threats to be covered by exception reports include unexpected behavior from the outside of the company (e.g., unusually low sales figures) as well as unexpected results from the inside (e.g., a product batch not meeting quality standards). They usually exhibit a relative and fine-grained time reference.	Exception reports can only be archived safely if the threat that they relate to does not exist any longer, so that no incidents can happen anymore (e.g., if the product they relate to has been taken out of production).

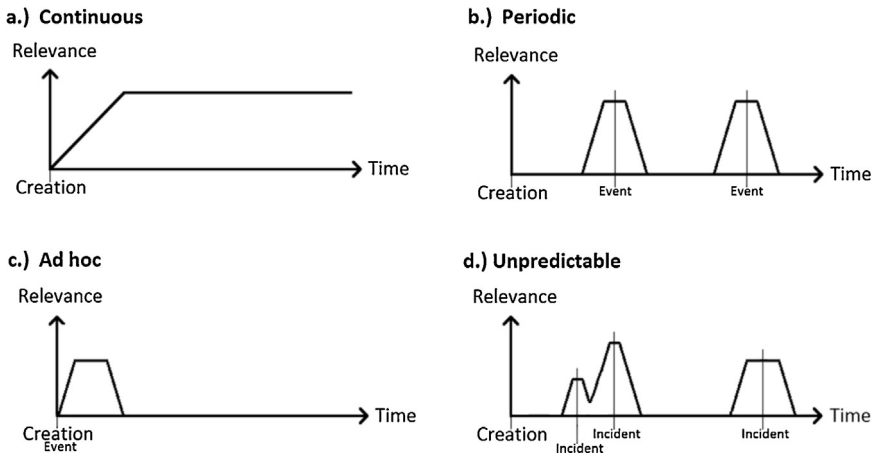

Fig. 1. Relevance pattern types of reports.

relevance. They can be extracted automatically from the log files of the search component (e.g., Park, Jee, Lee, Jung, & Lim, 2012). These are usually stored long enough to be able to fully characterize a report's relevance when the screening mechanism is executed.

While the complete relevance pattern of a report may be derived solely from $S(t)$, other available data can be utilized to counteract possible contortions of this indicator. E.g., it can be assumed that reports that are used frequently are over-proportionally accessed directly, while less frequently accessed reports first have to be searched for. This cannot be accounted for when relying solely on $S(t)$ but by comparing $S(t)$ and $A(t)$. Besides, it is not unusual that some reports upon the occurrence of certain events (e.g., the end of a period) are send out of the BI system (e.g., per email) in form of a static document (such as, e.g., a pdf-file) after they have been filled with data (executed) (push principle). This is reflected neither in $S(t)$ nor in $A(t)$ since in this case no search or direct access takes places. For such cases, the report's *range* or *penetration* (that is, the number of its recipients) can be used as an substitutive indicator for its relevance.

3.4. Architecture

Summarizing, Fig. 2 shows the architecture of a BI system with an integrated archiving component that estimates the relevance of a report based on the indicators discussed above.

The origin of the presented architecture is a user who has an IN and intends to retrieve information in order to fulfill it. If she/he knows a report that is suited for this purpose, she/he can access this report directly (using the BI system's navigation), resulting in an increase in $A(t)$. Otherwise, she/he can use the search component to find suitable reports. In this case, she/he enters a search query (e.g., "sales"), whereupon the search component scans all reports and responds with a list of links to the reports it considers matching this search query; the estimated degree of match defines the order of these links. The user then can open one or more of the suggested reports, whereby their values of $A(t)$ increase, change the search request, or cancel the search. Each time a report is opened, it is executed by the execution component. The report opened as the last within a session (one or several related search requests) can be considered the result of the search (e.g., Albakour et al., 2011). Only for this report, a specific entry is made in the log files of the search component, illustrating another difference between $S(t)$ and $A(t)$. The archiving component can access the log files and extract $S(t)$ for all points in time t. Whenever its screening mechanism is executed (depending on its execution rule), it on this basis reconstructs each reports' historic relevance pattern. The latter may subsequently be modified using further information on the report such as $A(t)$ or its range. Based on the result and possibly additional information on the environment (e.g., the current date), the archiving component decides about archiving or reactivating the report by its decision rules and implements this decision using its physical mechanism. Note that this phase of IST happens at a different point in time than each phase of IR described above.

As mentioned earlier, the interaction between the user and the BI system can also be initiated by the latter's execution component. Triggered by an event or incident, it can execute a report and send the resulting document to specified recipients (one or more users). Within the proposed framework, this is not affected by whether the report has been archived or not.

4. Evaluation

4.1. Approach

We evaluate the effect of integrating an archiving component into a BI system within the proposed framework through simulation. For this purpose, we create simulated BI systems. In one half of them, an archiving component is implemented as described. The other half, which represents conventional BI systems, lacks such a component. Then we compare the search effort simulated users have to make to find a relevant report between both groups and observe these values over time.

In real BI systems, search effort depends on the algorithm the search component employs. As elaborated on earlier, this factor that relates to IR is complementary to our IST-based approach of archiving reports. This is why we want to exclude its influence as far as possible. Therefore, we implement the simplest "search algorithm" imaginable in all our simulated BI systems: the list of results presented to a simulated user simply contains links to *all* available reports. These links further are just sorted randomly – not by any degree of match between the search request and the reports' attributes, as it would be the case in most real BI systems. For reports that are not (only) accessed via the search engine, the list of links can be interpreted as the order in which the user looks through the available reports manually.

The only difference between the BI systems with and without an archiving component is that in the former, there effectively exist two lists of results: one for the active reports and one for the archived. Both lists are as well just randomly sorted. Since the goal of archiving is to distinguish relevant from irrelevant reports, how-

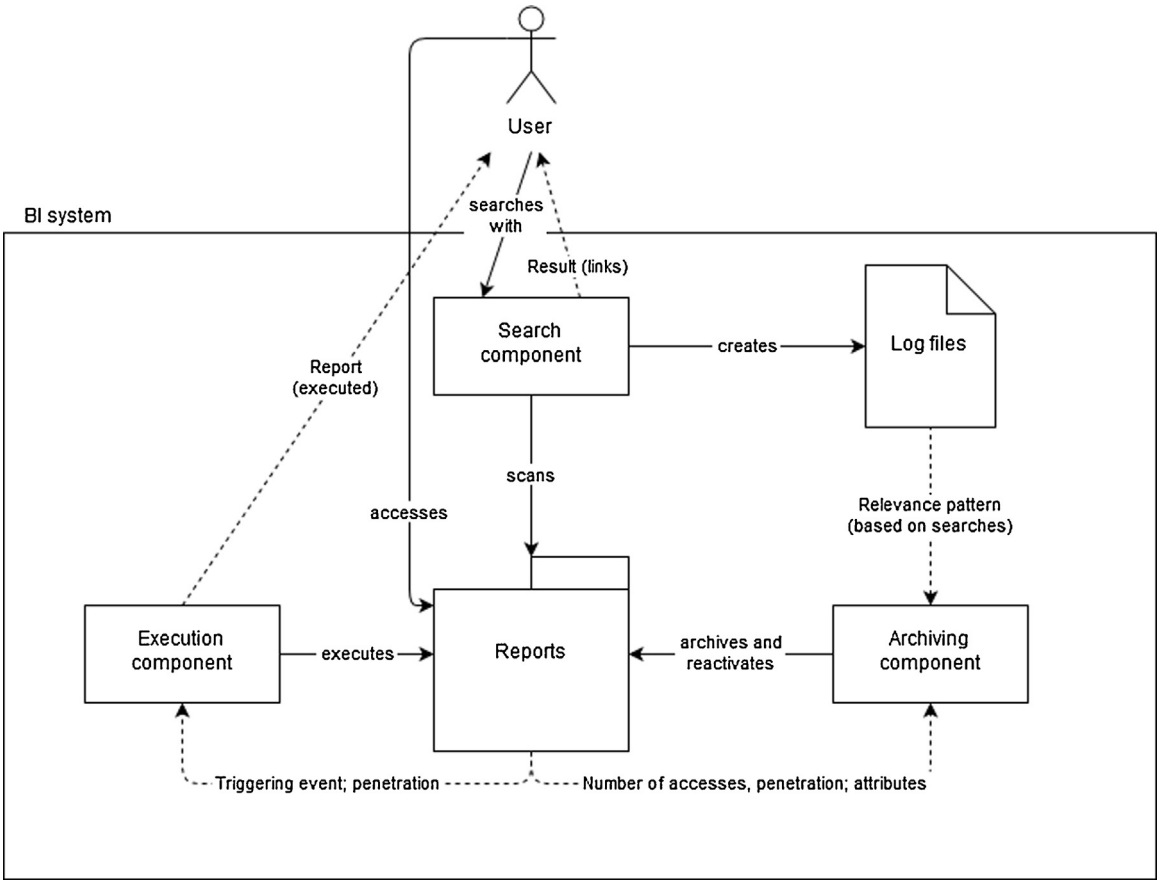

Fig. 2. Architecture of a BI system with an archiving component.

ever, search effort should be reduced if the list of the active reports is looked through *before* the list of archived reports. In this case, archiving has been successful. In contrast, given an unfavorable archiving, search effort can even increase, since the search in both separate lists would then be worse than a search in a shared list. The difference in search efforts, therefore, is a valid and non-trivial indicator for the performance of the archiving component.

Formally, we measure the *mean relative search effort (MRSE)*, which we define as

$$\mathrm{MRSE}_t^j = \frac{1}{n_t^j} \times \sum_{i=1}^{n_t^j} \frac{p_{t,i}^j}{r_t^j}. \quad (1)$$

Intuitively, MRSE_t^j states which proportion of the r_t^j reports that are available in a BI system j at a point in time t have to be looked through on average until a report is found that satisfies the IN i. This is averaged over the n_t^j INs of all users. Correspondingly, archiving in j is the better, the smaller MRSE_t^j is. $p_{t,i}^j$ denotes the position of the first report fulfilling i in a list of all reports. Without archiving, MRSE_t^j is expected to take a value around 0.5, reflecting that due to the random ordering of reports, a suitable report is found on average at position $r_t^j/2$. If archiving is successful, $p_{t,i}^j$ and, hence, MRSE_t^j should be reduced. Note that the MRSE relates the number of reports that have to be looked through to the total number of available reports. This enables a fair comparison between different BI systems and between different points in time within one BI system, since possibly differing total numbers of reports are accounted for.

4.2. Design of the archiving component

To make all BI systems with an archiving component comparable, we implement the same archiving component into each of them. In the following, we describe its design.

First, we employ the screening mechanism screenReports given in Section 3.1. As we have explained there, this is a simple screening mechanism, but there is no reason to use another in this context. Second, we tie its execution to a time-based execution rule. More concretely, it is executed every $d = 10$ time units; we denominate the corresponding points in time as t^*. We skip the first two executions (that is, we allow for a lead time of $2 \times d$ time units) in order to enable the archiving component to collect sufficient data for archiving. Third, the physical archiving and reactivation process is realized through setting a flag for each report that indicates whether it currently is active or archived. Furthermore, we create an additional attribute V_t that records how often it has been archived in total up to each point in time t. For specifying, fourth, the decision rules to decide on the archiving and reactivation of reports, we take into account that there may be different data available for each report depending on its access type, as explained in Section 3.3. For reports that are at least partly accessed via the search component, the number of searches $S(t)$ is known for each t. Instead of relying on $S(t)$ directly, however, we derive the indicator $S_t{}^* = S(t^* - d) - S(t^* - 2 \times d)$ from it that only utilizes data from the period between the last two executions of the screening mechanism. This is in order to prevent newly created reports from being directly archived. To, similarly, prevent reports with a periodic relevance pattern from being archived and reactivated periodically, one can specify through $V_t{}^*$ that they should not be archived more

than a certain number of times. For reports that can only be directly accessed, one can solely utilize their number of accesses $A(t)$ up to t^* (and derived values) in addition to metadata such as the date of the last change in t^*, D_t*. As long as these are not historized, it cannot be decided on the reactivation of such reports, which is why we do not reactivate them.

Specifically, we use the following decision rules:

Rule 1:	Archive reports that have only been searched for using the search component if $S_{t^*} \leq 3$ and $V_{t^*} \leq 1$.
Rule 2:	Archive reports that have only have been directly accessed if $A(t^*) \leq 3$ or $D_{t^*} \leq t^* - 60$.
Rule 3:	Archive reports that have been directly accessed or searched for using the search component if $S_{t^*} \leq 3$ and, in the case of $A(t^*) \geq 3$, additionally $D_{t^*} \leq t^* - 60$.
Rule 4:	Reactivate reports that have not only been directly accessed if $St > 3$.

As reasoned in Section 3.1, decision rules in practice should be chosen in dependence on the company's specificities. Therefore, the given ones should be considered exemplary.

4.3. Simulation design

We simulate a period of $T = 200$ time units and a number of $2 \times J = 100$ BI systems. As mentioned earlier, the latter are partitioned in a group of size J with an archiving component and a group of the same size without such a component. To enable a fair comparison between both groups, they are based on the same data regarding reports and relevance patterns; more precisely, for each BI system in the one group exists a BI system in the other group that is based on the same data, and vice versa. In the following, we describe how these data are simulated.

4.3.1. Reports

We start by simulating the reports available in each BI system. This involves three steps:

First, it has to be decided on the report portfolio, that is, on the share of reports with a certain relevance pattern type. For this purpose, we draw for each BI system the total number of reports with relevance pattern type k from a uniform distribution over $\{1;m_k\}$. Accounting for reports with a continuous relevance pattern being underrepresented in real BI systems (Mills, Davis, & Blum, 2012), we set $m_k = 1000$ for this pattern type and $m_k = 2500$ for the periodic and the ad hoc pattern type. We do not consider reports with an unpredictable relevance pattern in our simulation because they regularly would not be archived (see Table 1), and, therefore, just affect the BI systems with and without an archiving component in the same way. Thus, omitting them does not affect the comparison between both groups.

Second, one should take into account that not all reports are created at the same time or, as we have put it earlier, that new reports are steadily created. To simulate this, we draw for each report a creation time t_{from} from a uniform distribution over $\{1;T\}$. At each point in time that comes before t_{from}, we regard the corresponding report as non-existent. As an exception to this, we set $t_{\text{from}} = 1$ for all reports with a continuous relevance pattern because these often are created at the implementation of the BI system (e.g., due to the request of the client) (Loshin, 2013, p. 57, Chapter 4). Note that we do not delete reports because this would contradict the intention pursued with the integration of an archiving component.

Third, to simulate that a report can be accessed in different ways, we randomly assign each report one of three possible access types, each with the same probability: only direct access, access only via the search component, or a combination of both.

4.3.2. Relevance patterns

Once the reports have been created, the next step is to simulate their relevance patterns. These depend on the reports' relevance patterns types on one hand and on the users' concrete IR behavior on the other hand. To account for this, we simulate them in two steps:

First, we assign each report periods of potential relevance that correspond to its relevance pattern type (reduced to its essentials). For reports with a continuous relevance pattern type, this period is $\{1;T\}$, since they, by definition, can always be relevant. For reports with an ad hoc relevance pattern, the period is set to $\{t_{\text{from}};t_{\text{to}}\}$, where t_{to} is drawn from a uniform distribution over $\{t_{\text{from}};t_{\text{from}}+20\}$. Note that such reports, thus, are relevant at most for 10% of the simulation period, by which we model their ad hoc character. Reports with a periodic relevance pattern can be relevant in several periods $\{t^l; t^l + v\}$ with $t^1 = t_{\text{from}}$ and $t^{l+1} = t^l + v + w$. Their duration of (potential) relevance v and their duration of irrelevance w are drawn from uniform distributions over $\{1;10\}$ and $\{10;50\}$, respectively, implying $v \leq w$.

Second, the actual relevance of a report results from user interest in (the content of) this report. To simulate this interest, we draw for each BI system j the total number n_t^j of INs in a point in time t from a uniform distribution over $\{1;1000\}$. Then, we distribute these INs randomly across all reports of j that are potentially relevant in t (according to their periods of potential relevance), corresponding to the users' IR behavior. Each IN is assigned to each report with the same probability and independently of previous assignments. As a consequence, each report can fulfill none, exactly one, or multiple INs in t, and a different number of INs later on. Thereby, its relevance is simulated to be dynamic.

4.4. Results

Fig. 3 shows the development of the mean relative search effort MRSE_t^j as defined in Section 4.1 in BI systems with and without the archiving component over time. Besides the respective averages $\frac{1}{J} \times \sum_{j=1}^{J} \text{MRSE}_t^j$, the minimum values $\min_j \text{MRSE}_t^j$ and maximum values $\max_j \text{MRSE}_t^j$ across all BI systems are depicted for both groups.

As we had expected, the average MRSE without archiving fluctuates around the value 0.5. It should be noted that this value means that search effort increases with the total number of reports. In contrast, the average MRSE with archiving decreases over time. At the end of the simulation period ($t = T = 200$), at which 51.6% of all reports are archived, it takes the value 0.3143. Interestingly, however, it can be observed that it increases again immediately after archiving. This is due to falsely archived or newly created (and not archived) reports. Thus, the choice of the archiving period d is associated with a trade-off between an increase in search effort due to a too frequent archiving and due to not archived irrelevant reports.

While the corridor between the minimum and the maximum MRSE across all BI systems without archiving remains almost constant, it clearly increases over time with archiving. At the end, the corresponding values for the latter group are 0.2117 and 0.4573, respectively. This can be explained by the differing composition of the report portfolio. E.g., the MRSE tends to be the lower the higher (lower) the number of reports with an ad hoc (a periodic) relevance pattern is, as these types of reports are the least (most) common to be falsely archived. However, starting at about $t = 180$, the MRSE for all BI systems with an archiving component is lower than for all BI systems without such a component, even when the report portfolio is least favorable for archiving. Finally, one can observe that the average MRSE lies below the middle of the corridor. This indicates

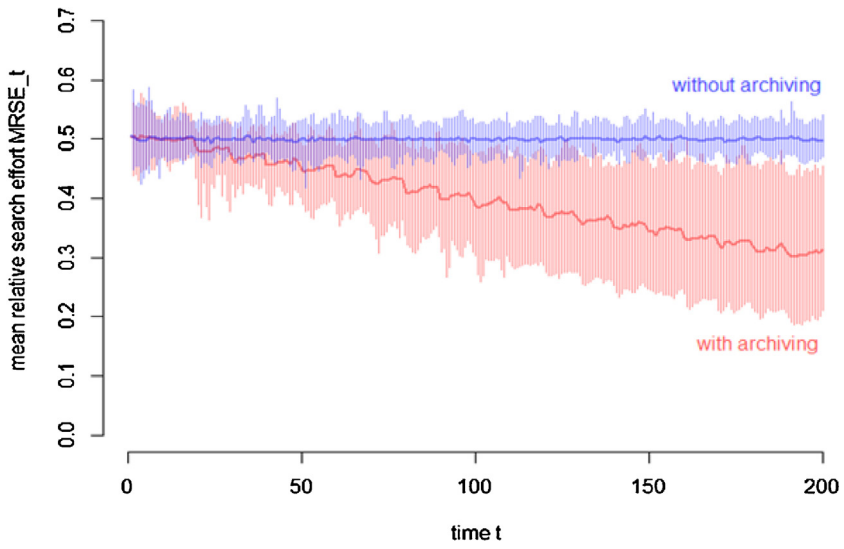

Fig. 3. MRSE with and without archiving over time.

that there were more BI systems with report portfolios well-suited for archiving than with poorly-suited report portfolios.

5. Conclusion and future research

In this paper, we have demonstrated how an automated archiving component can be designed and integrated into a BI system. Doing so significantly reduces users' search effort, as we have shown by comparing BI systems with and without an archiving component in a simulation study. Therefore, companies through our approach can meet the danger of knowledge workers being unable to find relevant information in reasonable time and, hence, reduce their search costs. The next step can thus be to implement our approach into real BI systems and investigate by how much the search costs decrease under which conditions.

The aforementioned danger is acute in the field of BI, as explained earlier, which is why we have focused on this field. However, it also occurs in other fields in which a large number of information resources has to be managed. Some parts of this work, particularly the idea of archiving information resources that have become irrelevant, can be transferred to these. This is another direction for future research. However, we are convinced that for archiving to be successful, the evaluation of information resources' future relevance needs to be field-specific. That is, one should be aware of that the relevance indicators we have used and especially the relevance patterns that they measure may not be transferrable to other fields.

Archiving is, of course, not the only way to deal with the aforementioned danger. While most other approaches are based on IR, what has the drawbacks mentioned earlier, future research can also explore further IST-based approaches. Since our method is in some sense complementary to IR, it may be suited as a basis to combine both types of approaches.

References

Albakour, M., Kruschwitz, U., Nanas, N., Song, D., Fasli, M., & de Roeck, A. (2011). Exploring ant colony optimisation for adaptive interactive search. *Lecture Notes in Computer Science*, *6931*, 224–231.

Borlund, P. (2003). The concept of relevance in IR. *Journal of the American Society for information Science and Technology*, *54*(10), 913–925.

Böhringer, M., Gluchowski, P., Kurze, C., & Schieder, C. (2009). On the role of social software techniques for the design of self-organising enterprise reporting portals. In *Proceedings of the 31st International Conference on Information Technology Interfaces (ITI)* Cavtat, Croatia, (pp. 153–158).

Chewning, E. G., Jr., & Harrell, A. M. (1990). The effect of information load on decision makers' cue utilization levels and decision quality in a financial distress decision task. *Accounting, Organizations and Society*, *15*(6), 527–542.

Cleverley, P. H., & Burnett, S. (2015). Retrieving haystacks: A data driven information needs model for faceted search. *Journal of Information Science*, *41*(1), 97–113.

Cooper, W. S. (1971). A definition of relevance for information retrieval. *Information Storage and Retrieval*, *7*(1), 19–37.

Davenport, T. H., & Beck, J. C. (2000). Getting the attention you need. *Harvard Business Review*, *78*(5), 118–125.

Eckerson, W. W. (2008). *Pervasive business intelligence—Techniques and technologies to deploy BI on an enterprise scale*. Best Practice Report. TDWI.

Edmunds, A., & Morris, A. (2000). The problem of information overload in business organisations: A review of the literature. *International Journal of Information Management*, *20*(1), 17–28.

Engler, T. H., Schulz, M., & Winter, P. (2014). Towards a social data enriched algorithm for business intelligence portals. In *Proceedings of the 1st European Conference on Social Media (ECSM)* Brighton, UK, (pp. 740–743).

Feldman, S., & Sherman, C. (2003). The high cost of not finding information. White paper. IDC.

Gilad, B., & Gilad, T. (1988). A systems approach to business intelligence. *Business Horizons*, *28*(5), 65–70.

Golfarelli, M., Rizzi, S., & Cella, I. (2004). Beyond data warehousing: What's next in business intelligence? In *Proceedings of the 7th ACM international workshop on data warehousing and OLAP (DOLAP)* Washington DC, USA, (pp. 1–6).

Haas, M. R., & Hansen, M. T. (2007). Different knowledge, different benefits: Toward a productivity perspective on knowledge sharing in organizations. *Strategic Management Journal*, *28*(11), 1133–1153.

Hannula, M., & Pirttimäki, V. (2003). Business intelligence empirical study on the top 50 Finnish companies. *Journal of American Academy of Business*, *2*(2), 593–599.

Hawking, D. (2004). Challenges in enterprise search. In *Proceedings of the 15th Australasian Database Conference (ADC)* Dunedin, New Zealand, (pp. 15–24).

Herring, J. P. (1988). Building a business intelligence system. *Journal of Business Strategy*, *9*(3), 4–9.

Hjørland, B. (2015). The phrase information storage and retrieval (IS&R): An historical note. *Journal of the Association for Information Science and Technology (ASIS&T)*, *66*(6), 1299–1302.

Hsu, K. C., & Li, M. Z. (2011). Techniques for finding similarity knowledge in OLAP reports. *Expert Systems with Applications*, *38*(4), 3743–3756.

Hwang, M. I., & Lin, J. W. (1999). Information dimension, information overload and decision quality. *Journal of Information Science*, *25*(3), 213–218.

Imhoff, C., & White, C., (2011). *Self-service business intelligence—Empowering users to generate insights*. Best Practices Report. TDWI.

Inmon, W. H. (2010). Manage data growth and optimize the data warehouse infrastructure with data warehouse archiving. White paper. Informatica.

Işik, Ö., Jones, M. C., & Sidorova, A. (2013). Business intelligence success: The role of BI capabilities and decision environments. *Information & Management*, *50*(1), 13–23.

Kulkarni, A., & Callan, J. (2015). Selective search: Efficient and effective search of large textual collections. *ACM Transactions on Information Systems (TOIS)*, *33*(4), 17.

Loshin, D. (2013). *Business intelligence: The savvy manager's guide* (2nd ed.). Waltham: Morgan Kaufman.

Maciariello, J. A. (1984). *Management control systems*. Engelwood Cliffs: Prentice-Hall.

Marjanovic, O. (2007). The next stage of operational business intelligence: Creating new challenges for business process management. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS)* Waikoloa, Hawaii, (p. 215c).
Menon, A., & Varadarajan, P. R. (1992). A model of marketing knowledge use within firms. *Journal of Marketing*, *56*(4), 53–71.
Mills, G. Z., Davis, A. Q., & Bluhm, J. L. (2012). *2010 Census cost & progress assessment report*. US Census Bureau.
Moore, J. H., & Chang, M. G. (1980). Design of decision support systems. *ACM SIGMIS Database*, *12*(1–2), 8–14.
Nadj, M., Morana, S., & Maedche, A. (2015). Towards a situation-awareness-driven design of operational business intelligence and analytics systems. In *Proceedings of the 10th International Conference on Design Science Research in Information Systems and Technology (DESRIST)* Dublin, Ireland, (pp. 33–40).
O'Reilly, C. A. (1982). Variations in decision makers' use of information sources: The impact of quality and accessibility of information. *Academy of Management Journal*, *25*(4), 756–771.
Park, K., Jee, H., Lee, T., Jung, S., & Lim, H. (2012). Automatic Extraction of User's Search Intention from Web Search Logs. *Multimedia Tools and Applications*, *61*(1), 145–162.
Ronen, I., Shahar, E., Ur, S., Uziel, E., Yogev, S., Zwerdling, N., et al. (2009). Social networks and discovery in the enterprise (SaND). In *Proceedings of the 32nd ACM SIGIR International Conference on Research and Development in Information Retrieval* Boston, USA, (p. 836).
Ruthven, I., & Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, *18*(2), 95–145.
Schubmehl, D., & Vesset, D. (2014). The knowledge quotient: Unlocking the hidden value of information using search and content analytics. In *White paper*. IDC.
Seidel, S., Knackstedt, R., & Janiesch, C. (2006). Procedure model for the analysis and design of reporting systems: A case study in conceptual modelling. In *Proceedings of the 17th Australasian Conference on Information Systems (ACIS)* Adelaide, Australia, 59
Sprague, R. H. (1980). A framework for the development of decision support systems. *MIS Quarterly*, *4*(4), 1–26.
Switzer, G. J. (1994). A modern approach to retail accounting. *Management Accounting*, *75*(8), 55–58.
White, C. (2005). The next generation of business intelligence: Operational BI. *DM Review*, *15*(5), 34–37.
Wixom, B., & Watson, H. (2010). The BI-based organization. *International Journal of Business Intelligence Research*, *1*(1), 13–28.