Towards a Taxonomy of Real-Time Business Intelligence Systems

Mario Nadj
Karlsruhe Institute of Technology (KIT), Institute of Information Systems and Marketing (IISM), Karlsruhe, Germany, mario.nadj@sap.com

Christian Schieder
BHS Corrugated, Weiherhammer, Germany, cschieder@bhs-corrugated.de

Follow this and additional works at: http://aisel.aisnet.org/ecis2017_rp

Recommended Citation
http://aisel.aisnet.org/ecis2017_rp/33
TOWARDS A TAXONOMY OF REAL-TIME BUSINESS INTELLIGENCE SYSTEMS

Research paper

Nadj, Mario, Karlsruhe Institute of Technology (KIT), Institute of Information Systems and Marketing (IISM), Karlsruhe, Germany, mario.nadj@kit.edu and SAP SE, Walldorf, Germany, mario.nadj@sap.com

Schieder, Christian, BHS Corrugated, Weiherhammer, Germany, cschi"eder@bhs-corrugated.de

Abstract

As the provisioning of timely business insights becomes increasingly relevant for organizations, real-time business intelligence (RTBI) is considered a promising vehicle to minimize the time span between elicitation, analysis, and subsequent action. However, so far, there seems no structured and systematic taxonomy in which RTBI systems can be classified and uncertainty regarding the dimensions and characteristics that constitute these systems. By analyzing extant business intelligence literature, this paper develops a taxonomy for RTBI systems to address these current impediments. Reviewing 89 studies in leading journals and conferences during the years 2000-2016, we found 29 relevant characteristics along seven dimensions for RTBI systems. Our taxonomy may serve as a foundational step to incorporate a broader theoretical perspective to integrate concepts and findings across all seven dimensions. The main contribution of the paper is in the organization and structuring of the body of knowledge in RTBI along the identified dimensions and characteristics for the advancement of the field, which is specifically relevant due to its relatively young nature. For practice, our taxonomy helps organizations to evaluate their RTBI systems or conceive the challenges of building such a system either from scratch or as an update of their existing infrastructure.

Keywords: Business intelligence, real-time business intelligence, taxonomy development, literature review, decision support.

1 Introduction

Business intelligence (BI) covers all “techniques, technologies, systems, practices, methodologies, and applications that analyze critical business data to help an enterprise better understand its business and market and make timely business decisions” (Chen et al. 2012, p. 1166). The challenges involved with the provisioning of timely insights based on the swamping of high-velocity data, exceed the scope of traditional BI tools (Geerdink 2013), impeding companies to use such systems in operational contexts (Watson & Wixom 2007). A prominent case stems from the logistics industry by real-time monitoring the movements of packages from pickup to delivery, as well as identifying failures of packages being at risk of delay. In the case of imminent breakdowns, such information can optimize subsequent actions, like re-routing, only if the required data freshness is assured (Hackathorn 2004). The common denominator for these challenges is a breed of BI technologies summarized under the umbrella term real-time BI (RTBI) (Chaudhuri et al. 2011). RTBI addresses the time span between elicitation, analysis and action capable of reacting in a timely fashion in changing business situations (Hang & Fong 2010). But RTBI systems are multiplex, diverging in scope and functionality (Nadj & Schieder 2016). Some solutions incentivize employees’ behavior. Others enhance the customer/supplier relationship or focus on the main performance indicators (KPIs) for monitoring purposes (Watson 2009).
However, this diversity includes a complex of dimensions and characteristics across different RTBI systems, resulting in confusion which components RTBI systems incorporate. Despite the young nature of RTBI research (Kees et al. 2015), numerous studies have already been published. These investigations range from software agents that deal with supply chains (almost) autonomously (Nissen & Segupta 2006), advanced data warehouses (DWH) that are able to perform data mining while executing online analytical processing (OLAP) queries (Rácz et al. 2007), or sophisticated dashboards that help inexperienced users to trigger ad-hoc queries (Steinkamp & Mühlbauer 2013). Still, these studies are lacking what RTBI embraces. Both layman and experts agree that RTBI includes the support of operational decision-making by performing real-time analyses of real-time data resulting in a better understanding of the actual situation (Azvine et al. 2006). However, there seems to be no structured and systematic form to classify RTBI systems and an uncertainty remains concerning the dimensions and characteristics that constitute these systems (Hang & Fong 2010). Although information systems (IS) scholars are addressing the real-time environment as their context of study (Baboo & Prabhu 2013), the IS community has not yet been satisfactory engaged in contemplating upon methodological considerations of examining RTBI and applications of theory building (Nadj & Schieder 2016).

In this paper, we create a taxonomy for RTBI systems with a combined three phase approach based on Webster and Watson (2002) and Nickerson et al. (2013) to tackle down the aforementioned impediments in RTBI research. Taxonomies are a tool for developing intricate, sophisticated systems depicting a phenomenon in its defining characteristics (Rich 1992). The contributions of our work address both academics and practitioners. First, creating a RTBI taxonomy is a foundational step towards facilitating the development of theories and future publications in a real-time environment (Iivari 2007). Williams et al. (2008) argue that a taxonomy creation is the first step to differentiate between various characteristics of objects of interest and rigorous theory development. Bapna et al. (2004) suggest taxonomies as a prerequisite to create theories ex-post, which is important as the RTBI field lacks in methodological reflections (Nadj & Schieder 2016). Second, our taxonomy structures the body of RTBI knowledge for the advancement of the field, which is specifically relevant due to its young nature (e.g., Kees et al. 2015). Third, it offers a novel understanding of possible reasons why adequate RTBI efforts have remained challenging. Fourth, our taxonomy describes systematically the RTBI area paving the way for software development endeavors. Practitioners could evaluate their systems and conceive the challenges of building a RTBI system either from scratch or validate their existing RTBI infrastructures. Following chapter 1, we present our research method in chapter 2. Our results are shown in chapter 3, before we discuss them in chapter 4. Chapter 5 concludes the paper.

2 Research Method

Taxonomies are relevant in research and practice since “the classification of objects helps researchers and practitioners understand and analyze complex domains” (Nickerson et al. 2013, p. 336) and offer “a fundamental mechanism for organizing knowledge” (Wand et al. 1995, p. 291). We used a combined three-phase approach (cf. Figure 1) applying the well-established approaches of Webster and Watson (2002) for literature reviews and Nickerson et al. (2013) for taxonomy development.
2.1 Literature Search Process

For preparing the keyword search (step a), we employed – besides “real-time business intelligence” – an assemblage of related keywords, such as “operational business intelligence”, “real-time data warehouse”, “real-time enterprise”, and “real-time analytics”. “Active data warehousing” was also used in the search process as it refers to the timely update of the DWH in near real-time. Within the first phase we gathered studies along two different sources: database (DB) search (step b) and articles proposed by a senior scholar (step c). Due to the young nature of RTBI this research is being published in BI-specific outlets outside of the typical IS domain (→ database search). Further, a keyword search usually does not use already available literature (→ senior scholar’s advice). We concentrated our search on the years 2000 – 2016. The DB search employed ACM Digital Library, Ebscohost, IEEE Xplore Digital Library, and ProQuest. As proposed by Webster and Watson (2002) a complete keyword search was applied leading to an initial set of 987 papers within the DB search. With this, we removed duplicates from every search result and identified 334 relevant papers.

In the second phase we conducted a backward (step d) and forward search (step e). As suggested by Webster and Watson (2002), we went backward by reviewing the citations for the studies identified in the first phase to discover prior articles relevant to our topic. We identified 22 articles for the backward search. Further, we conducted a forward search using Web of Science to identify studies citing the articles identified in the first phase. Through this procedure we identified 18 additional papers. In total, the final basket of literature contained 374 papers. We examined the resulting hits regarding their suitability by reading the full-text version of the respective articles out of which 89 were analyzed in detail. To ensure high relevance, we defined the following inclusion criteria by concentrating only on (1) architectural concepts, (2) frameworks, (3) visions, and/or (4) descriptions of RTBI systems.

2.2 Taxonomy Development

In the third phase, we used the iterative, seven-step process approach by Nickerson et al. (2013) to classify the identified papers within a RTBI taxonomy. Numerous IS studies implement the method, illustrating its robust and adaptable nature along multiple contexts (e.g., Emamjome et al. 2014; Kees...
et al. 2015; Müller et al. 2015). Further, the method suggests determining a meta-characteristic in the first instance that mimic the primary purpose of the taxonomy to facilitate the identification of relevant dimensions and characteristics. Finally, the method also incorporates quality criteria along a priori defined subjective and objective ending conditions to ensure the termination of each iteration.

Following this approach, scholars first have to determine the meta-characteristic – the “most comprehensive characteristic that will serve as the basis of the choice of characteristics in the taxonomy” (Nickerson et al. 2013, p. 343). The meta-characteristic should be grounded on the purpose of the taxonomy; this means the audience of the taxonomy and their expectations need to be considered. We target to support both practitioners and researchers in understanding the “state-of-the-art” of RTBI dimensions and characteristics required to implement such a system or compare existing instantiations along the identified dimensions. Hence, we selected the identification of central properties of RTBI systems for the realization of real-time processing as our meta-characteristic. Second, researchers need to define the conditions that finish the iterative taxonomy development process. We used the objective and subjective ending conditions formulated by Nickerson et al. (2013) for our process. To ensure an unbiased judgment of the ending conditions, three researchers assessed these conditions separately and discussed divergences together (cf. fourth step). Third, for each iteration, researchers must decide whether to follow either an empirical-to-conceptual (inductive) or conceptual-to-empirical (deductive) approach to identify the dimensions and characteristics. The deductive approach creates a taxonomy from theory or conceptualization by synthesizing the existing concepts into one integrative theoretical framework (Baumeister & Leary 1997; Nickerson et al. 2013). So far, such integrative knowledge is missing in RTBI literature, but valuable as it can serve as a ‘route map’ in structuring and organizing the various RTBI components. As there exists a vast amount of (partially) overlapping and/or incomplete RTBI conceptualizations or architectures, we applied the (deductive) conceptual-to-empirical approach with the aim to synthesize this available top-down knowledge (Baumeister & Leary 1997).

Fourth, based on the holistic meta-characteristics, more fine-grained dimensions need to be developed. For each dimension, characteristics need to be identified. In each iteration, the dimensions used in the present study were reviewed and those related to the meta-characteristic were integrated. For instance, Azvine et al. (2006) define three dimensions of RTBI systems: (1) analytics, (2) data integration, and (3) data sources. Based on these dimensions, we identified the associated characteristics based on our understanding of the latter. By classifying these characteristics, we recognized further significant differences between them, which we adjusted in the associated dimensions as well. For example, Hang and Fong (2010) supplement the architecture of Azvine et al. (2006) with a presentation dimension. When the entire set of studies had been clustered, we recognized discrepancies between the authors’ categorizations and reviewed their root causes. In some cases, a mismatch resulted from a false understanding. For instance, we clearly differentiated between the three dimensions: presentation (e.g., via dashboard), access to the application (e.g., via desktop), and data access (e.g., by performing an ad-hoc query). In other cases, a problem occurred in the dimension or characteristic itself, which resulted in the discovery of new or the refinement of existing dimensions or characteristics. For example, instead of following Azvine et al.’s (2006) solution to combine the data storage and data integration, we separated both dimensions as promoted by Chaudhuri et al. (2011). We repeated the development process until we could not identify any additional or redundant dimensions and characteristics.

Fifth, scholars need to map objects, i.e., examples, to the dimensions and characteristics. Objects can have different characteristics in a dimension, but do not have to entail one of the characteristics in a dimension (cf. Bailey 1994). This is typically the case for BI systems as they can rely on different means of data structures (structured, unstructured, or semi-structured), combinations of data access (batch and stream processing), or various forms of information presentation. We used the technologies from our reviewed studies to illustrate the distribution of RTBI research along the identified dimensions and characteristics. The assignment of studies to the dimensions and characteristics was conducted independently by two researchers. Any discrepancy between the two coders’ results was discussed and resolved by revisiting the original papers. Particularly with dashboards and advanced data visualization the assignment was sometimes not fully clear. We resolved this issue by analyzing the
degree of interactivity offered by the respective tools. Classical graphical elements, such as pie charts, bar charts, line charts, spark lines, bullet graphs, box plots, or scatterplots were assigned to dashboards, whereas more interactive forms, such as parallel coordinates diagrams, tree maps, or visual data discovery tools, were assigned to advanced data visualization. In turn, the codings for data-centric characteristics were rather straight forward as the respective articles typically investigated one specific characteristic, such as batch-oriented extract, transform and load (ETL) processes. Lastly, the assignment to the human access layer (e.g., mobile, web portal) was difficult as this information was simply missing in some papers and thus, in these cases, we could not assign the respective article. Sixth, the outcome of the steps three to five represents an initial or a revised taxonomy. Seventh, in case the objective and subjective ending conditions (cf. step two) are met, the taxonomy development process ends. Otherwise, the researchers need to repeat the steps three to seven. We examined all 89 studies before the objective ending conditions were achieved. After these iterations, all authors described the taxonomy as concise, robust, comprehensive, extensible, and explanatory; thus, the subjective ending conditions were met. For all iterations, we applied the (deductive) conceptual-to-empirical approach in step 3, as we aimed for consolidating the vast amount of (partially) overlapping and/or incomplete existing conceptualizations into an integrative theoretical framework (Baumeister & Leary 1997).

3 A Taxonomy of Real-Time BI Systems

In this paper, we developed a taxonomy with 29 characteristics along seven dimensions (cf. Table 1). We moved the coding schema of the taxonomy with a detailed view of all studies into the appendix.

<table>
<thead>
<tr>
<th>Human Access</th>
<th>Presentation</th>
<th>Analytics</th>
<th>Data Access</th>
<th>Data Storage</th>
<th>Data Integration</th>
<th>Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>18,0</td>
<td>59,6</td>
<td>77,5</td>
<td>67,4</td>
<td>88,8</td>
<td>78,7</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>1</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>C</td>
<td>2.2</td>
<td>9.0</td>
<td>13.8</td>
<td>11</td>
<td>36.3</td>
<td>37.4</td>
</tr>
</tbody>
</table>

Table 1. Taxonomy for RTBI Systems: Classification along Dimensions and Characteristics

3.1 Descriptive Statistics

Most publications study data storage (88.8%), followed by data integration (78.7%) and analytics (77.5%). Data storage is dominated by analytical DB characteristics due to the broad application of the DWH concept (79.8%). Data integration characteristics mostly focus on the optimization of (micro) batch-oriented ETL/ELT processes (66.3%). As for analytics, real-time data mining is the most distinctive characteristic to derive data insights (59.6%), whereas only 31.5% of the studies address business activity monitoring (BAM). The second relevant dimensions’ block refers to data access (67.4%), followed by data sources (62.9%), and results presentation (59.6%). Query (39.3%) and stream processing (40.4%) are the mostly addressed characteristics within data access. As for data sources, studies predominantly examine operational systems (49.4%) and external sources (32.6%), such as social media content. For presentation, the most studied characteristics refer to notifications and alerts (37.1%), followed by reports (36%) and dashboards (36%). The least important dimension is human access (18%). Here, numerous studies investigate web portals (13.5%) and mobile BI (9%).

Twenty-Fifth European Conference on Information Systems (ECIS), Guimarães, Portugal, 2017
3.2 Relevance of the RTBI Dimensions

With regard to **data access** (1), RTBI processes are executed over data coming from a wide range of different source systems (e.g., operational systems, sensors, or external sources). Challenges refer to the various data structures (structured, unstructured, or semi-structured) and frequencies at which the data are created (streaming, transactional, or master data) (Azvine et al. 2006). **Data integration** (2) comprises all characteristics responsible for moving the data from the source systems into the DWH in real-time. However, the plethora of source systems entail data of “varying quality, use inconsistent representations, codes, and formats, which have to be reconciled” and considered for RTBI scenarios (Chaudhuri et al. 2011, p. 89). Due to the specifics of the data sources, the combination of different **data storage** (3) technologies seems necessary for RTBI scenarios to develop a comprehensive, analytical IS capable of handling structured and poly-structured data at high levels of frequency (Lechtenbörger & Vossen 2016). Within the **data access** (4), the primary challenge for RTBI refers to providing the exponentially growing amount of data in such a way that it can be processed in real-time (Agrawal 2009). Hereby, most businesses rely on queries or stream processing solutions.

The **analytics** (5) dimension is responsible for turning raw data into actionable knowledge (Baars & Kemper 2008). The layer is of specific importance for RTBI solutions due to its role to reduce the “analysts in the middle” issue and enforce the degree of automation. In contrast, the **presentation** (6) dimension involves the user “with direct, interactive, or batch access to data, while hiding the technical complexity of data retrieval” (Khan & Quadri 2012, p. 67). RTBI systems require interactive visualization forms to enable the decision maker to induce informed actions (Dix 2013). Fresh data are only of help, if they result in timely decisions and actions. To enable RTBI, tools also need to be accessible for diverse business users of all management hierarchies. Such **human access** (7) is specifically relevant due to its potential to make RTBI solutions more pervasive, for instance, by offering mobile and cloud-based solutions (Chen et al. 2012). The next sections address in detail the relevance of the RTBI characteristics along the identified dimensions and highlight prominent contributions.

3.2.1 Data Sources

**Operational systems** still define the backbone of businesses. Transactional data are captured from business systems, such as enterprise resource planning (ERP) or customer relationship management (CRM) (Hang & Fong 2010). Further, as some organizations still perpetuate legacy systems, we use the term **legacy DBs** in our taxonomy to account for this aspect. In turn, technical advances offer new possibilities of real-time monitoring and analyses by capturing sensor data (e.g., machines’ pressure or temperature) or clickstream data (e.g., web server logs in online retail stores). Contrary to transactional data, which are stored in a central DWH for analysis purposes, sensor or clickstream data are too large to be centrally stored. However, only small portions of such data are relevant for most analyses (Chen et al. 2014). Thus, it is not even necessary or recommendable to store all data in one place.

The use of **sensors** and **server logs** enable the collection and analyses of very fine-grained data for specific machines with a strong location and/or time dimension. With the introduction of RFID scanners and tags, goods can be marked, tracked, and analyzed (Chen et al. 2012). Thus, literature emphasizes the growing challenge regarding platform technologies (e.g., J2EE), syntax standards (e.g., XML), semantics, and data quality metrics (Agrawal 2009). Typical solutions of integrating diverse meta-data semantics refer to shared ontologies to improve compatibility, performance, and reduce semantic ambiguity (Cui et al. 2007). As an example, product entries might entail different price currencies, which are automatically converted via context-based mismatch reconciliation at runtime to the respective user location (Azvine et al. 2005). With the increased availability of **external data**, enterprises also try to integrate this content (Negash & Gray 2008). Shroff et al. (2011) propose an enterprise information fusion framework to integrate tweets or Facebook posts into the DWH.
3.2.2 Data Integration

During the ETL process data are harmonized to fit a unified schema and cleansed of syntactic and semantic deficiencies (Kemper et al. 2010). While standard ETL tools offer reliable technologies, anomalies occur within the real-time domain (Jörg & Dessloch 2010). To process large amounts of data in a short period, ETL processes demand exclusive access to the source systems, which inevitably leads to downtimes (Pareek 2010). One key bottleneck describes the transformation stage as it represents a resource-intensive activity. Since the harmonization of heterogeneous data schema of the source systems is not practically feasible, Freudenreich et al. (2013) transform the ETL into an ELT process: the data are extracted and loaded into the DWH before the time-consuming transformation. Another variation is the so-called microbatch ETL process with shortened execution intervals to perform changes (often within one to four hours) (Adzic et al. 2003). However, the batches’ interval cannot be reduced arbitrarily as shortened intervals drive the risk of overstraining the operational systems (Schröder 2007).

Contrary, continuous approaches avoid the massive transport of data in a limited time period as it occurs within (micro-) batch processing. Instead data “trickle” continuously from the source system to the DWH (Kimball 2004). This approach generates less workload, reduces the probability that the networks are “clogged”, as well as spreads the workload to the systems involved over a longer period. A prominent continuous approach refers to enterprise application integration (EAI) – a congregation of technologies using standardized interfaces to prevent the need for applications that communicate with each other to implement the interface of one another (Schiefer & Seufert 2005). Such functionality makes EAI particular interesting to integrate data in real-time (Bange 2016). Usually, EAI is implemented by an EAI Message Bus (also called Enterprise Service Bus or ESB) – a message-oriented middleware that receives and forwards messages of particular applications (Panahi et al. 2009).

Changes are propagated by the source systems and are continuously reported as messages to the ESB, which forwards them directly to the DWH or an ETL application, where the data are adapted to the schema of the DWH. Thus, EAI can meet the requirements for a real-time ETL process; however, its introduction into an information technology landscape is time-consuming (Schröder 2007).

Service-oriented architectures integration (SOA) is another promising approach to implement a continuous ETL approach (Linthicum 2003). Applications are set off, and their individual abilities are offered as a service. Notably, the involved source systems and DWH are connected via web services to the company’s network. The transformation stage can be implemented within the DWH itself or a dedicated application (Schleisinger et al. 2005). The approach presented is similar to the EAI infrastructure. Web services take the role of the adapters between the proprietary middleware and the applications. The effort to introduce a SOA is high as ideally, each interconnected application in a company needs to be split into services. Latency and speed are comparable to an EAI solution (Schröder 2007).

Data integration can also be executed on demand through the arrival of concrete queries leveraging data federation and virtualization techniques in order to reduce the overhead of continuous data capture. This approach enables a virtual, unified view of small amounts of time-sensitive, heterogeneous data that do not require much processing prior to its presentation within reports (Russo et al. 2014). Enterprise information integration (EII) represents a prominent example for on-demand data integration approaches. Swaminathan (2006) define EII as means for integrating data in real time enabling standardized data access via a single, federated data layer. Multiple online transaction processing (OLTP) sources are encapsulated via EII. The uniform access to unstructured or semi-structured sources (e.g., text, XML files, or web services) is a beneficial capability. Obstacles refer to the limited historical data storage and long-lasting response times in case of low-quality data (Schröder 2007).

3.2.3 Data Storage

Traditionally, structured, decision-relevant data are stored in a centralized location – a DWH. The idea is to create “a subject-oriented, integrated, non-volatile, and time-variant collection of data” to perform retro perspective analyses supporting strategic management decisions (Inmon 2002, p. 31). DWHs are considered analytical DBs optimized for the application of analytical IS and not for universal applica-
bility (e.g., for transactional systems). Queries are complex and proposed on demand, as well as include aggregation on a large number of datasets. This degree of pre-determinedness enables the use of specific DB architectures and designs, such as the star or snowflake schema. This results in benefits regarding query performance, scalability, and maintainability (Bange 2016). Analytical DBs have become the de facto standard of existing DWHs and BI systems (Nationaler IT Gipfel 2014). However, while improvement methods, such as buffering or indexing, have further reduced the query response time, the continuous growing amount of data stored puts limits on analytical DBs.

In contrast, transactional DBs manage a large number of small online transactions, with basic functions, such as insert, update, or delete. The focus relies on simple queries, which can be quickly processed while assuring data integrity in multi-access environments. Transactional DBs work with detailed and current data, and the schema used refers to the entity model (usually 3NF), which is not well suited for analytical purposes. However, with the growing importance of analytical IS in the course of digitization of business processes, large quantities of poly-structured data at high frequencies from various sources (e.g., social media, sensors) need to be considered for decision making (Bange 2016).

Since these oceans of data streams cannot be stored economically within a centralized location, increasingly the Hadoop distributed file system (HDFS) with its complementary components (e.g., MapReduce), are used to extend traditional DWHs approaches (Lechtenbörger & Vossen 2016). HDFS represents a distributed file system, where significant amounts of data can be stored, read, and displayed in a fault-tolerant manner (Nationaler IT Gipfel 2014). Data are subdivided into blocks that are replicated through local storage by cluster nodes. HDFS is based on a master-slave architecture, in which a naming node manages the file name space as master (including the file-block-assignment and the localization of blocks), and the data codes manage the actual blocks as slaves. The data decomposition of data into several independent blocks enables parallel processing of these blocks.

Another relevant storage technology refers to in-memory DBs, which has been established by advances in multicore architectures, 64-bit technology, and reliable, large main memory (Wessel et al. 2013). Such DBs leverage large main memory leading to performance gains that have been prevented by comparatively slow disk access in the past. Benefits manifest in the immediate receipt of relevant results when entering a search query promoting RTBI scenarios (Plattner & Zeier 2011). While traditional DBs rely on row-oriented data organization, where the attribute values of data records are arranged side by side (Krueger et al. 2010), in-memory DBs mainly use column-oriented data storage, which has already been successfully applied in the OLAP environment (Plattner 2009). Further, combined with in-memory DBs, column-oriented data storages might also be relevant for OLTP systems (Krueger et al. 2010). The implementation of efficient compression algorithms might lead to a significant reduction of the data volume up to a factor of ten enabling the storage of large data amounts (Sinzig et al. 2011). Similarly, with the in-memory DB technology hybrid approaches of both OLTP and OLAP are conceivable making their separation obsolete (Knabke & Olbrich 2016).

NoSQL DBs – generally understood as “not only” SQL – attempt to overcome the limitations of relational DBs by using non-relational, distributed DBs, such as document-oriented DBs (e.g., CouchDB), column stores (e.g., Google Big Table), key-value stores (e.g., Amazon SimpleDB), or graph-oriented DBs (e.g., Neo4J). NoSQL DBs are characterized by open source, horizontal scalability, as well as high availability abandoning rigid DB schemas and the ACID (atomicity, consistency, isolation, durability) properties of DB transactions. Instead, the concept of “eventual consistency” is often applied to increase performance. When partitioning is fixed, and there are no new updates for a sufficient amount of time, this concept assures that any updates are propagated through the system, so that “eventually” all nodes are consistent. Prominent critics state that “No ACID Equals No Interest” or “NoSQL Means No Standards” as OLTP represents the core business for many companies requiring consistency guarantees and uniform interfaces between the used DBs (Lechtenbörger & Vossen 2016).

As a counter draft, NewSQL DBs (e.g., Google’s F1) emerged to assure SQL standards and ACID properties, while addressing the scalability benefits of NoSQL to handle vast data amounts (Shute et
al. 2013). Research showed that even web applications (e.g., AdWords) cannot operate without ACID transactions, and that at this point, the eventual consistency does not suffice (Bailis et al. 2015).

3.2.4 Data Access

In context of data access, the batch-oriented processing paradigm has been successfully employed. Batch jobs are performed without manual intervention and trained against a dataset at scale creating results in the shape of data files and computational analysis. This approach focuses on data throughput of high volume data, rather than on latency resulting in significant time lags (Agrawal 2009). Large datasets are gathered all at once, processed and then the batch outcomes are created. A prominent example is MapReduce addressing mainly two purposes: (1) separating work to diverse nodes within the cluster or map, as well as (2) organizing and reducing the outcomes per node into a cohesive query response. Benefits manifest in automatically handling issues, such as “data partitioning, node failures, managing the flow of data across nodes, and heterogeneity of nodes” (Chaudhuri et al. 2011, p. 95).

In contrast, stream processing typically uses “continuous” queries to process and analyze a window of recent data. Such solutions complete real-time computations (seconds at most) enabling the analysis of data in motion including a scalable, fault-tolerant, and highly available architecture (Hang & Fong 2010). A class of systems that can enable stream processing refers to complex event processing (CEP), which captures and processes data expressed as events in a data stream without the need to access a DWH (Pareek 2010). While monitoring data from multiple streams at once (as well as from traditional sources), CEP engines can correlate across multiple streams, detect pre-defined complex patterns, and take appropriate actions thereupon. CEP engines have been employed in the financial sector (e.g., for algorithmic stock trading), clickstream analysis, or process monitoring. The growing relevance of real-time analytics makes it necessary to revisit many traditional data mining techniques in the context of streaming data. Best practices refer to the use of the lambda data architecture, to combine streaming data with other enterprise data, or to store stream data for offline analytics (Russom et al. 2014).

Lastly, queries are the standard technique for directly accessing data for analytical purposes and knowledge discovery. While this approach induces high flexibility and reusability, only users with a profound understanding of the data manipulation language (such as SQL) can execute these queries. However, the strengths of a direct data access should not be underestimated as queries can be applied to both the DWH and the source system paving the way for real-time reporting (Kemper et al. 2010).

3.2.5 Analytics

Reporting servers are the simplest analytical method and de facto standard to define, conduct, and render reports, where the desired data are selected and aggregated (Chaudhuri et al. 2011). Datasets could be exported using knowledge distribution functionalities for data exploration (Bange 2016). For achieving real-time reporting, most research suggests technical solutions, such as a better EAI integration, the use of cloud computing, or mobile devices to further close the latency gap (Trigo et al. 2014).

Common BI operations, such as filtering, aggregation, drill-down and pivoting are supported by OLAP servers. OLAP aims at quickly answering analytical questions to generate decision-relevant information leveraging so-called data cubes – multidimensional and hierarchical data models (Thomas & Datta 2001). OLAP requirements refer to the speed of information delivery, analysis complexity, and data to be processed (Bange 2016). For applying OLAP on real-time data, Conn (2005) establish a connection between the OLAP system and DWH by duplicating the data in the OLAP repository for subsequent analysis. Further, as recalculation affect the analysis latency, Mazur et al. (2008) suggest an auxiliary database, which stores previous calculations, to solve this issue.

The set of algorithms provided by data mining engines supports in-depth data analyses exceeding what is offered by standard OLAP or reporting servers, and promotes the ability to develop predictive models (Chaudhuri et al. 2011). Traditionally data mining approaches have been packaged distinctively (e.g., SAS or SPSS) in order to select a dataset from the DWH, run a sophisticated analysis, identify
predictive models (e.g., a decision tree), and finally integrate these models in the operational DBs to be actionable (Russom et al. 2014). Such course of action is time-consuming and often requires manual intervention from highly educated human experts and thus is antithetical for RTBI. Azvine et al. (2006) urge the need for a high automation degree within RTBI solutions to reduce such latency. For instance, Netz et al. (2001) suggest to use “in-database analytics” by integrating data mining technology in the DWH backend. Azvine et al. (2005) and Deng et al. (2009) propose the use of a real-time data mining engine to minimize the analysis time lag. With it, the data mining process is executed automatically and with different parameters before the best solution is illustrated in a suitable means to (non-expert) users without the need to comprehend, chose, or configure analysis algorithms. By using incremental learning methods, the suitability of each model can be improved (Chapinski et al. 2008).

Lastly, the observation of web activity, business processes, or sensor data (Hang & Fong 2010) is supported by BAM (Golfarelli et al. 2004). BAM monitors business processes and real-time performance measures on the basis of KPIs (Azvine et al. 2005). Present performance deviations are identified and displayed to the user, so that corrective actions can be induced instantaneously. Wang and Wang (2005) suggest the use of multiple software agents based on pre-defined business rules. To promote real-time data processing, inefficient business processes require an automatic realignment.

3.2.6 Presentation

Reports continue to be a major presentation form (even for RTBI scenarios). They rely on combinations of text, tables, and/or charts (Russom et al. 2014). Traditionally, reports were created performing several SQL queries by BI analysts. An improved automation within RTBI systems decreases the necessity of human interaction in the process of creation (Azvine et al. 2005). Reporting tools promote increasingly the automatic creation of periodic reports on a regular basis or ad-hoc queries on demand (Chen et al. 2012). Löser et al. (2009) suggest a system that facilitates the creation of complex queries on the basis of a basic input terminology to perform ad-hoc queries across unstructured web data.

Other research points to dashboards as a prevalent front-end application of RTBI enabling users an overview of the business situation at a glance (Negash & Gray 2008). Compared to reports, dashboards allow more detailed analyses and interactivity (Kohlhammer et al. 2013). Users can observe, drill down, filter, and customize visualizations of data (Eckerson 2006). For fulfilling the real-time paradigm, dashboards require time-critical visualizations for real-time monitoring with continuous and automatic data updates (Hang & Fong 2010). Advanced operational dashboards enable users to activate processes in business applications or to draw real-time data snapshots from them (Azvine et al. 2005). Whereas a consolidated access to multiple applications is desired, many real-time dashboards are streamlined to a particular application increasing decision latency (Russom et al. 2014). Further, graphical representations offer rich meaning into trends, interdependencies, and relationships (Larkin & Simon 1987).

However, within RTBI scenarios, more complex and multidimensional data show up representing a challenge for standard visualizations (often labelled as “read only”). Hence, advanced visualization techniques facilitate the promotion of more interactive techniques (Basole et al. 2012). Most of these tools can operate with in-memory computing increasing the available data to users who wish to build (complex) data visualizations or run sophisticated queries without having to experience I/O bottlenecks due to data access on disk storage. Contrary, some queries or reports still show higher performance when accessed on disk-based analytical platforms (Russom et al. 2014). Studies expect that users will require data refreshes more often than this has been the case with traditional analytic platforms (e.g. OLAP) and that this will increase the demand of interactive visualizations (e.g., parallel coordinates diagrams, tree maps, or visual data discovery tools).

As RTBI are more persuasive, alerts and notifications become more relevant to proactively make users aware about imminent business situations. Scholars differ between time-critical alerts demanding immediate attention, and postponable, non-time critical alerts (Ho & Intille 2005). Influencing variables include type (e.g., whether it represents a warning or announcement), modality (e.g., visual, au-
dio, haptic), frequency rate (i.e., how often to inform the user), and timing (i.e., when to trigger the alert). Regarding timing, alerts might induce automatic corrective actions or notify the user within a dashboard as soon as critical measures deviate from predefined objectives (Negash & Gray 2008). Research proposes an event ticker based on relevance and priority (e.g., via colourful traffic lights).

3.2.7 Human Access

BI tools are often enabled by desktop clients to access information. For improving the user access, RTBI tools plug into standard office tools or provide web portals (e.g., on a J2EE infrastructure), which grant access to various applications relying on the benefits of a consolidated appearance and easy administration (Baars & Kemper 2008). Automatic adaptive user profiling enables the modification of the software’s appearance on required user needs (Kemper et al. 2010). Technical performance advances of mobile devices, as well as cloud computing, promote new opportunities for mobile RTBI. Mobile devices enable a new way of computer interaction allowing users to access and analyze data anytime from anywhere (Chen et al. 2012). Visualizing data on mobile screens is bound to limitations (e.g., limited screen size, differing aspect ratio), and thus requires special attention (Chittaro 2006).

4 Discussion

In this paper, we developed a taxonomy for RTBI systems following a three-step approach. In the following, we want to emphasize six central implications.

First, BI is facing a paradigm shift towards provisioning real-time support during process execution. While traditional BI approaches integrate data in batch processes on a daily, weekly, or monthly basis supporting strategic decisions from (mostly) retrospective and manual analysis connected to a restricted audience of BI experts and managers, RTBI, in contrary, is persuasive and strive to gather, integrate and automatically analyze data as they occur fostering operational decisions (Bucher et al. 2009). This implies that RTBI needs to automate both the information flow from operational to strategic levels, as well as the actions required to convert strategic objectives back to operational drivers. Common obstacles proposed by the investigated studies refer to costs, design issues within real-time systems, the state of data management infrastructure, and insufficient staffing (Russom et al. 2014). Second, for transferring a traditional BI system into an RTBI solution, bottleneck components of the DWH environment that induce data latency have to be also recognized, so that technological and organizational countermeasures can be carried out (McKenna 2011). Hereby, the load cycles of the DWH describes a main bottleneck for minimizing data latency. To overcome this bottleneck, we identified several integration techniques that go well beyond traditional ETL processes, such as data federation and virtualization, message buses, micro batches, or SOAI. Third, another problem refers to the diverse metadata semantics from the various data structures. Hereby, ontology-based techniques are proposed to support the building of a real-time data fusion platform. At this time, most studies still examine the handling of structured data (or more specifically relational) in real time, followed by logs and semi-structured data. In turn, studies anticipate increasingly more real-time processing of weblogs and clickstreams, social media data, and unstructured data (Russom et al. 2014).

Fourth, within the analytics dimension, we identified the “analyst-in-the-middle” problem in a traditional BI system as a common bottleneck for real-time analysis. BI scenarios involve analysts to drive the operations through tools providing operational decision makers with necessary data via reporting. However, the time span needed by the analyst for handling and configuring the BI software should be minimized in a real-time environment. Prominent solutions refer to a high degree of automation achieved by the application of real-time data mining engines or sophisticated OLAP engines (i.e., with a direct DWH connection). Fifth, whereas the reduction of data and analysis latency are mostly embraced by technical solutions (Olsson & Janiesch 2015), the minimization of decision latency seems more complex as it refers to both technology and the user (Watson 2009). Regarding decision latency, most studies suggest the use of advanced dashboards with real-time data to decrease decision latency. Contrary to standard dashboards, RTBI dashboards should be more interactive allowing users to acti-
vate processes or gather snapshots of real-time data (Azvine et al. 2005). Users are increasingly leveraging advanced visualization techniques due to their ease of use, powerful self-service functions, and the ability of tools to process large datasets with real-time performance. **Lastly**, the term “real-time” has become an umbrella concept involving various speeds, time frames and execution frequencies (Nadj & Schieder 2016). As a time delay between the data occurrence and the execution of an action seems imminent, “real-time” in the verbatim, immediate sense is often not realized (Eckerson 2004). Often data do not need to be real-time. The freshness of data merely has to suit the respective business process (Watson et al. 2006), which defines what can be seen as acceptable latency (McKenna 2011): “the key concept behind »real time« is that our artificial representation must be in sync with the real world so that we can respond to events in an effective manner” (Hackathorn 2004). Promising practices refer to columnar databases or in-database analytics. From a technical view terms like “near-real-time”, “right-time” or “just-in-time” might be more precise. However, the term “real-time” has become a catchphrase for academics and practice (Azvine et al. 2006). But with the latest generation of event processing and in-memory functions, the term “true real time” has appeared in the sense of milli- or nanoseconds.

5 Conclusion

The contributions of our work target both academics and practitioners: **First**, we created a taxonomy of RTBI systems as a first step towards facilitating the development of theories in a real-time domain. This move is relevant as the IS community has not yet been satisfactory engaged in neither contemplating upon methodological considerations, nor subsequent theory building for RTBI systems. **Second**, our taxonomy classifies the RTBI knowledge in a structured form, for illustrating the state-of-the-art and future research, which is specifically relevant due to the community’s young nature. **Third**, the classification offers a new understanding of reasons why adequate RTBI efforts have remained so intricate considering the prevalent complexity that defines these systems. **Fourth**, practitioners may apply our taxonomy as criteria to evaluate to what extent their solution portfolios address the real-time paradigm. This is important as organizations increasingly realize the potential value of RTBI.

To ensure sufficient breadth, we conducted a structured literature review. Our taxonomy is designed upon a broad foundation integrating various architectural concepts, frameworks, visions, and/or descriptions. Embedding RTBI systems in existing concepts offers a theoretically based understanding of their constituting parts. This reveals that RTBI systems are too complex to be considered as a homogeneous group, which justifies the necessity of a classification order. To ensure sufficient depth, we used the taxonomy development process proposed by Nickerson et al. (2013) allowing us to account for all relevant dimensions and characteristics. Thus, we believe that our taxonomy represents the current state-of-the-art of RTBI systems. However, our research comes also with some limitations: Any bias in the selected key words, would also provoke a bias in our results and thus threaten the completeness of our taxonomy. Further, new technological paradigms might extend or change the current taxonomy with regard to both: dimensions and characteristics. On the one hand, advanced machine learning algorithms might decrease the role of decision makers and reduce the relevance of the presentation dimension. On the other hand, new forms of human-computer interactions (voice and gesture recognition) might revolutionize the way RTBI dashboards and reports are designed. Both trends might have downstream effects on the requirements for data management and modelling. As future work, we plan to conduct an extensive survey among community members and practitioners to validate our results.

Acknowledgements

We thank Jonas Braun and Dirk Hoffmann for their great support in the qualitative content analysis.
Table 2. Taxonomy for RTBI Systems with reviewed Studies

| Human Access | Presentation | Analysis | Management | Data Access | Data Storage | Data Integration | Data Sources | Human Access | Presentation | Analysis | Management | Data Access | Data Storage | Data Integration | Data Sources |
|--------------|--------------|----------|------------|-------------|--------------|-----------------|--------------|--------------|--------------|------------|-------------|-------------|--------------|--------------|----------------|--------------|
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |
|              |              |          |            |             |              |                 |              |              |              |            |             |             |              |              |                |              |

Twenty-Fifth European Conference on Information Systems (ECIS), Guimarães, Portugal, 2017
References


Nadj & Schieder / Towards a Taxonomy of Real-Time Business Intelligence Systems


(SUGI'04). SUGI, pp.1-12.


ment Review 17 (4), 758–781.
IEEE, pp. 606–610.


