Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

A CAPABILITY APPROACH FOR DESIGNING BUSINESS INTELLIGENCE AND ANALYTICS ARCHITECTURES

Julian Ereth
University of Stuttgart
julian.ereth@bwi.uni-stuttgart.de

Henning Baars
University of Stuttgart
henning.baars@bwi.uni-stuttgart.de

Abstract

Business Intelligence and Analytics (BIA) is subject to an ongoing transformation, both on the technology and the business side. Given the lack of ready-to-use blueprints for the plethora of novel solutions and the ever-increasing variety of available concepts and tools, there is a need for conceptual support for architecture design decisions. After conducting a series of interviews to explore the relevance and direction of an architectural decision support concept, we propose a capability schema that involves actions, expected outcomes, and environmental limitations to identify fitting architecture designs. The applicability of the approach was evaluated with two cases. The results show that the derived framework can support the systematic development of fundamental architecture requirements. The work contributes to research by illustrating how to capture the elusive capability concept and showing its relation to BIA architectures. For further generalization, we created an open online repository to collect BIA capabilities and architectural designs.

1. Introduction

The integrated approaches to IT-based management and decision support, subsumed under the term *Business Intelligence and Analytics* (BIA), have proven to be relevant drivers for sustained business development [1, 2]. Advances into new application areas and ongoing technology innovations both on the side of data capturing and storage and on the side of data analysis and presentation are constantly transforming the role of BIA. Established business intelligence centers with their roots in management support and reporting are increasingly being complemented by Big Data and analytics initiatives that can often be found in the decentral Line-of-Business units. The awareness rises that there is a need to come up with an enterprise-wide governance and infrastructural umbrella for all data-utilization activities in order to truly reap strategic benefits [3, 4]. Such endeavors, however, are hampered by the plethora of novel technologies, frameworks, and products as well as the constantly changing solution space, particularly in the arcent of Big Data and advanced predictive analytics. Outside the "traditional" data warehousing and reporting environments, architectural blueprints and guidelines only exist for a few specific and narrow BIA domains like the Industrial Internet of Things. We suspect that this leads to a high level of uncertainty when it comes to strategic decisions in BIA architectures.

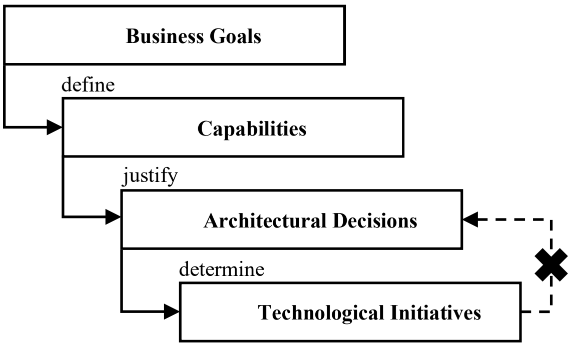

Figure 1. Architectural Decisions Hierarchy

Ideally, organizations follow the hierarchy outlined in Figure 1 and base their IT architectures decisions upon a well-conceived rationale and map them to business goals – prior to selecting particular technologies and tools [5]. Enterprise Architecture Management (EAM) frameworks, like Zachman [6] or TOGAF [7], provide guidelines for building business-oriented IT architectures. TOGAF uses *capabilities* as a linking pin between business strategy and IT architecture. However, from a practical perspective, the capability construct remains rather lofty and all frameworks leave a methodological gap when it comes to the derivation of concrete requirements [8].

The goal of this research is to provide a conceptual guidance for deriving and formulating BIA capabilities in order to bridge the chasm between the business goals and architectural requirements. Accordingly, the research question we pursue to answer is: *How can we structure and operationalize BIA capabilities in a way that can be translated into architectural requirements?*

We approached this topic by first exploring the relevance of such an approach with a series of expert interviews in six organizations that all were confronted with necessary changes in their BIA architectures. At this point we expected to find a limited portfolio of domain specific architecture blueprints that can be chosen from in a straightforward fashion (like the Industrial Internet Reference Architecture Analytics Framework [9]). As we explain later, however, our results instead support the conclusion that a much more granular approach is needed. This puts more weight on the formulation of detailed and well-structured capa-

URI: https://hdl.handle.net/10125/64400
978-0-9981331-3-3

bilities which can be linked to BIA architecture components and patterns that in turn lead to the selection of adequate products.

Informed by our studies and the literature on the subject, we developed a scheme that is designed to support this process. For purposes of evaluation, we tested our approach in two selected cases by formulating concrete capabilities and discussing the resulting architecture building blocks with the decision makers in charge. By doing so, we confirmed the general applicability of the approach and finished a full design research cycle. For further exploration and generalization, we have used our results to create a more wide-ranging online repository of BIA capabilities that is publicly accessible and extendable by the entire BIA community.

We deem our research as a direct *contribution to BIA practice* as it supports the process of either a) deriving a set of strategy-conforming BIA capabilities and selecting concrete architecture designs or b) scrutinizing an existing architecture for its strategy-alignment. We also *contribute to research* as our results help understanding the role of capabilities for the BIA domain and support the translation of the abstract capability construct into more concrete and measurable terms. The course of the paper is as follows: Section 2 discusses related work on BIA architectures and capability management. Afterwards, Section 3 outlines the methodology. Section 4 presents results of the exploration that are then used to derive the capability framework in Section 5. In Section 6, we present the evaluation based on two exemplary case studies. Section 7 then discusses results, limitations, and contributions. The paper closes with a conclusion that also gives an outlook on future work.

2. Related Work

2.1. BIA Architectures

The term BIA is the result of a continuous evolution of decision support systems in the enterprise context and thereby lines up with concepts like management support systems or executive information systems [10, 11]. Today, BIA refers to integrated approaches to IT-based management and decision support [11], which entails the collection, integration, refinement, and presentation of information throughout the business [12].

These tasks require concepts to collect heterogeneous data from different business departments or external sources as well as systems for storing, transforming, and visualizing data according to business requirements. BIA research and textbooks often use a layered approach, similar to Figure 2, to structure the various types of components in a BIA landscape [10, 12-14]. Here, the data management layer comprises various kinds of database or file-based tools that integrate and store data [15, 16]. Traditional BIA architectures are often designed to provide a consistent *single point of truth* with a data warehouse (DW) as an integrated, non-volatile, time-variant, and subject-oriented data repository [17] that is usually implemented with a relational database. Recently, alternative data management concepts have gained traction in the IT community, like Big Data stores or data lakes that utilize NoSQL databases or distributed file systems to store large volumes of unstructured information. Similarly, with the increasing importance of real time data, message queues and event-logs that allow to rapidly store and distribute data are becoming more popular [18]. At the next layer, the data is processed, i.e., extracted and transformed according to the requirements of the decision makers [15, 16]. This encompasses approaches like batch ETL jobs or real time stream processing, but also various data transformation methods, e.g., "classical" data mining or the application of more complex machine learning models [12]. The third layer then comprises the data presentation [12, 16] which can refer to the visualization of data with static reports, interactive dashboards, as well as the use of other ways to distribute BIA results like XML files, analytics notebooks, web services (e.g. via virtual machines, containers or function-as-a-service-models),

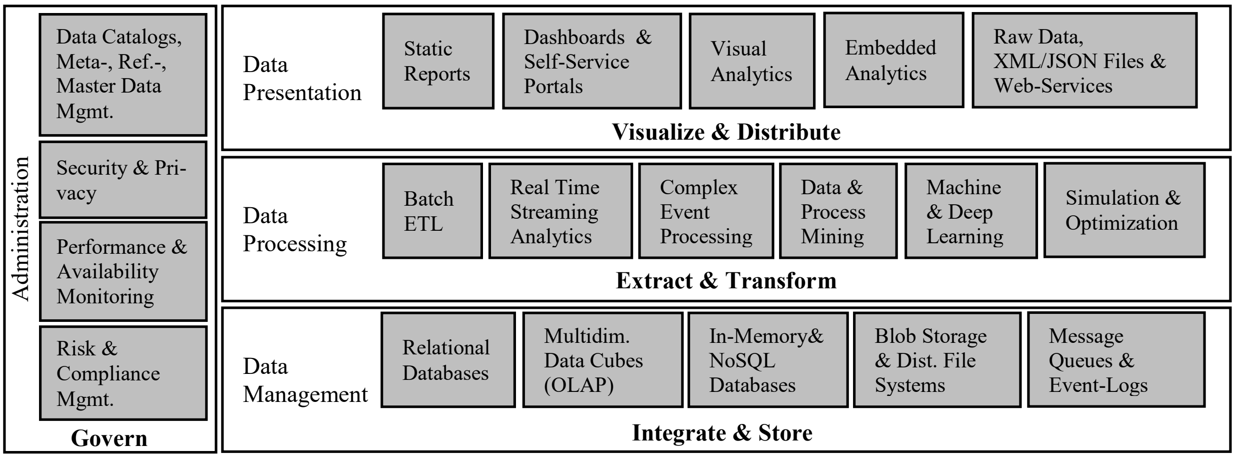
Administration
Data Catalogs, Meta-, Ref.-, Master Data Mgmt.
Security & Privacy
Performance & Availability Monitoring
Risk & Compliance Mgmt.
Govern
Data Presentation
Static Reports
Dashboards & Self-Service Portals
Visual Analytics
Embedded Analytics
Raw Data, XML/JSON Files & Web-Services
Visualize & Distribute
Data Processing
Batch ETL
Real Time Streaming Analytics
Complex Event Processing
Data & Process Mining
Machine & Deep Learning
Simulation & Optimization
Extract & Transform
Data Management
Relational Databases
Multidim. Data Cubes (OLAP)
In-Memory& NoSQL Databases
Blob Storage & Dist. File Systems
Message Queues & Event-Logs
Integrate & Store

Figure 2. Selected BIA components

source code management platforms like GitHub™, or even simple interfaces to raw data [14]. Lastly, there is an administration layer spreading over all other layers that contains components for orchestration and governance with systems like data catalogs for meta, master and reference data, for managing governance/risk/compliance requirements (including data lineage/data heritage), for security and privacy, access control, as well as for basic tasks like monitoring, backup or archiving [10, 14, 15].

Considering that Figure 2 only contains a few selected components on a highly abstract level where each component can again be implemented with numerous actual (and often only partly compatible) products, the complexity and intricacy of BIA architectures becomes palpable. Moreover, scenarios that go beyond classical management reporting often introduce manifold of new types of data sources and bring along unprecedented requirements on all presented layers [13, 18]. This stresses the relevance of a systematic design of BIA architectures.

2.2. Business Capabilities

Business Capabilities are abilities of an organization or system to perform certain actions in order to achieve certain outcomes [7, 19]. The idea of capability management goes back to the resource-based view of the firm, which states that the sustained competitive advantage of a firm is by the application of its set of resources [20]. An increasingly discussed variant of capability-based thinking is the concept of dynamic capabilities. In contrast to *operational (or zero-level)* capabilities that enable an organization to effectively execute its day-to-day activities, *dynamic capabilities* are the planned ability to effectively reconfigure existing operational capabilities in order to match changes in the business environment [21-24]. Apart from that, the term capability has gained popularity in the context of Enterprise Architecture Management (EAM). Here, capabilities constitute core building blocks of EAM and are used for strategic planning and decision making [7, 25].

Several researchers have shown how capability-thinking helps to strategically align IT domains [26] and explain how IT and BIA provides business value [1, 27]. Capabilities have also been used to discuss the adaption and maturity of BIA in companies [28, 29]. Moreover, Davenport et. al [30] developed a holistic framework that explains how a company can facilitate strategy, skills, organization, and technology to generate analytics capabilities for transforming data into knowledge. All these contributions see capabilities as core building blocks for bridging the gap between business and analytics. We propose to build up on those ideas and connect them with concrete architectural decisions (i.e. selection of BIA components and patterns as well as adequate implantations).

3. Methodology

The core artefact of our research is a schema for formulating capabilities. As Figure 3 illustrates, its development followed a four-stage processes, which encapsulates the three-cycle design science approach [31]. In the first stage we used literature and a series of expert interviews to explore the relevance of such a capability approach as well as its general direction (relevance cycle). In the second stage, we derived the actual capability schema, which we evaluated in the third stage with two concrete case studies (design cycle). Eventually, we scrutinized our results for their research impact (rigor cycle).

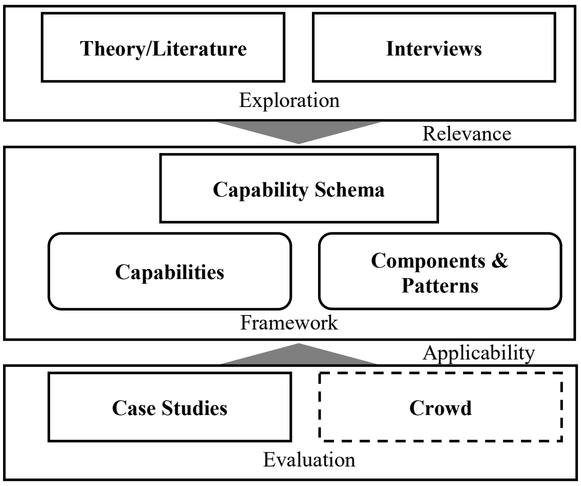

Figure 3. Course of the research

For the exploration in stage 1, we conducted 10 interviews (on average 40 minutes) in 6 organizations with decision makers from business and IT. The interviews were designed and qualitatively analyzed along a conceptual framework that was comprised of the *strategic priorities*, *challenges for the BIA architectures*, the *need for a systematic architecture decision process*, and *possible benefits of a capability-based approach*. All 10 interviews were fully transcribed [32]; the results were also informed by various documents from the organizations on architectures, technologies, and decision processes. As the findings supported the relevance of our approach, we continued with stage 2 in which we developed our generic schema according to the general requirements elicited the interviews. Apart from the study results, we also considered literature on capability management, EAM, and BIA as well as prior work and experiences in BIA requirements and architectures. In stage 3, we tested the applicability of our framework with two case studies where organizations were facing the need to transform their BIA architectures. Here, we conducted 19 additional semi-structured interviews with stakeholders from IT and business (60 minutes on average) where we reviewed the *status quo, the needs and possible solutions*. These interviews were also fully transcribed and analyzed qualitatively in order to derive a set of capabilities for each case. The capabilities were then used to identify adequate architectural patterns and components that were again mapped to exemplary products and then discussed and evaluated in 4 interactive workshops with BIA decision makers in the corresponding organizations (2 half-day group workshops per case). This evaluation step was primarily intended to test the general applicability of our capability-based approach for deriving an architecture. Given the broad range of possible capabilities and architectural options, we have not yet reached a "complete" solution that covers all possible components/patterns but rather came up with an exemplified outline of our approach. From here on, we propose to follow a crowd-based approach [33] to further explore and generalize this concept. Consequently, we implemented an online repository for BIA capabilities [34] that is publicly accessible and open to input from the entire BIA community.

4. Exploration

Table 1 lists the six cases and the number of conducted interviews. The table also summarizes the major architectural challenges that the respective organizations are currently facing.

Table 1. Explorative Interviews

Case (# Interviews)	Architectural Challenges
1: Life Science (1)	*Managing heterogeneous data in a data lake*
2: Airport (3)	*Connecting operational and managerial analytics systems*
3: City Administration (2)	*Building an open IoT data platform*
4: Wind Park Management (1)	*Processing large data sets in near real-time*
5: Energy provider (2)	*Operational data reporting*
6: Car manufacturer (1)	*Exploring unstructured data*

The first insight from the exploration is the variety of solutions and the broad range of individual architecture priorities. For instance, the vision of one organization was a storage for all company-generated data, whereas another organization saw real-time processing as its top priority. Accordingly, the organizations follow orthogonal approaches, with a distributed data lake as a focal point for the first company and multiple event-driven streaming pipelines for the other. Both scenarios differ strongly from the traditional management reporting world which in most cases quickly gravitate towards a centralized, multi-layered data warehouse.

Another result of the interviews was the lack of experience and generally accepted methods for architectural decision making. When asked about the origin of architectural approaches, only two interviewees could refer to a systematic approach or a reference architecture. The other architecture designs seemed primarily to be driven by a stepwise evolution or even trial-and-error. Even in the two cases, in which some kind of systematic architectural decision making was pursued, they were hardly aligned with business goals, but rather based on copying technically-focused architecture designs that were deemed to have some kind of "reference" character (e.g. an event hub based on Kafka™) or by simply following vendor ecosystems.

Moreover, the highly different approaches to similar challenges in the examined organizations indicate that it is not sufficient to simply select some kind of available blueprint. Operational reporting, for instance, was approached by one organization with an individual data warehouse that is loaded with regular ETL jobs (the "traditional" Business Intelligence approach), whereas another organization plans to solve the same requirement with a novel data virtualization layer. Only a closer look at the underlying environments can reveal if these choices were justified, which in this example seemed to be the case, as the solution in the second case had to integrate highly heterogeneous systems within which data structures often change. One blueprint that covers both approaches would probably end up in an over-engineered and inefficient approach. Two different blueprints (e.g. centralized vs. federated), as an alternative, might lead into a situation where there are as many "blueprints" as cases and thereby no blueprints at all. The conclusion is that instead of providing a one-fits-it-all blueprints, a guidance for the selection of architecture components and patterns is a preferable way.

The heterogeneity of the approaches also highlights the importance of environmental factors, as the underlying basic business problems are often the same and only differ in the environmental limitations like the required amount of expected or unexpected changes (agility), the number of involved technical systems or organizational units, or other specific characteristics like privacy requirements.

Lastly, the exploration showed that there is indeed a high uncertainty when it comes to architectural decisions, especially in new scenarios that involve requirements for Big Data, IoT, or real time analytics. The results support our assumption that one culprit here is the sheer number of tools and technologies. In some of the cases the interviewed architects tried to cope with these uncertainties by developing multiple proof-of-concepts in parallel in order to identify the best fitting solution. However, this approach had been described as highly inefficient as the teams had to learn different technologies and build multiple systems of which only one made its way to production.

All these issues lead to serious consequences with (i) inefficiencies due to constant changes and redesigns, as in case 1, where large efforts were required to cope with the quickly-evolving source systems, (ii) architectures that are unable to adapt to changing strategic objectives, as in case 2, where the traditional business intelligence stack is too rigid to support operational reporting, or (iii) even complete project failures, a point that was almost reached in case 6 where the integration of all required data sources has turned out to be too complex.

The results confirm the gap between business goals and actual architectural decisions. Moreover, all interviewed decision makers stated that a more business-oriented design of BIA architecture would be beneficial and a systematic and generally-accepted concept to derive architectural decisions would help them in their daily job.

5. Operationalizing BIA capabilities

According to the capability understanding introduced in section 2.2, capabilities are composed of *actions* and *expected outcomes*. Existing BIA research provides the conceptual means to instantiate these concepts: The relevant actions can be derived from the BIA layers in Figure 2. The expected outcome of a BIA application is *information* that needs to be provided in a certain *quality* and that is subject to an intended use. To break down the outcome, we build upon the existing body of knowledge on information quality and its dimensions [35-37]. Most authors agree on at least the following six core dimensions: *accuracy,* as the extent to which data correctly describes its subject, *accessibility,* as the level to which information is easily retrievable, *completeness,* as the extent to which relevant information is available in sufficient breadth and depth, *consistency,* as the absence of contradictions among information, *timeliness,* as the degree to which data is sufficiently up-to-date for the task at hand, *security,* as the level to which access to data is restricted appropriately, and *privacy,* as the degree of which data discloses personal or confidential information. There are more information quality dimensions discussed in the respective literature (e.g. believability or objectivity) that we intentionally exclude as we do not see immediate impacts on BIA architectures. Although a high degree of fulfillment in all dimensions is desirable, different use cases come with different priorities and acceptable trade-offs (e.g. high timeliness in real-time monitoring, low privacy when dealing with publicly available data, or low accuracy in data discovery scenarios).

These components allow us to articulate BIA capabilities in the form of "*we need to [action] in a way to get insights with [expected outcome] in order to support [business task]*". Here, actions can be expressed in a bipolar way (yes/no). The expected outcome can additionally be refined with a unipolar, ordinal scale (e.g. low, medium, high). Lastly, a link to a business task clarifies the intention of a capability and its right to exist. An example for a capability stated according to this scheme could be "*We need to extract and store stock data in a way to get insights with high accuracy and medium timeliness in order to support replenishment decisions in logistics*".

The identification of capabilities with this framework can happen top-down starting with business requirements (e.g. real-time machine monitoring presupposes certain capabilities) or bottom-up by abstracting from the current usage of technologies, tools, or solutions (e.g. a streaming platform enables certain capabilities). Hence, this approach can support decisions in greenfield scenarios as well as legacy (brownfield) architectures. So far, we have focused on the top-down direction as our exploration indicated that this is currently the primary pain point.

The study has also shown, however, that a capability in this form does not yet provide enough guidance for making concrete architectural choices as it still misses the specific conditions of the environment – it specifies the *what* and the *what for*, but not the *how*. This is why we extend the schema by adding particular *environmental limitations* that can stem from technical, organizational, personal, or legal factors. Our starting point for deriving those limitations is by conceptualizing a BIA architecture as a variation of an IT infrastructure: BIA architectures combine various IT components in order to support a larger portfolio of continuously changing (decision and management support) applications – which is basically the definition of an "IT architecture" applied to the domain of decision and management support. As a specific IT infrastructure, a BIA architecture can be approached with the conceptual toolset for IT architectures developed in the IS community, which includes attributes like *reach* and *range* [38] or *agility* [39]. In this context, *reach* captures the number and variety of busi-

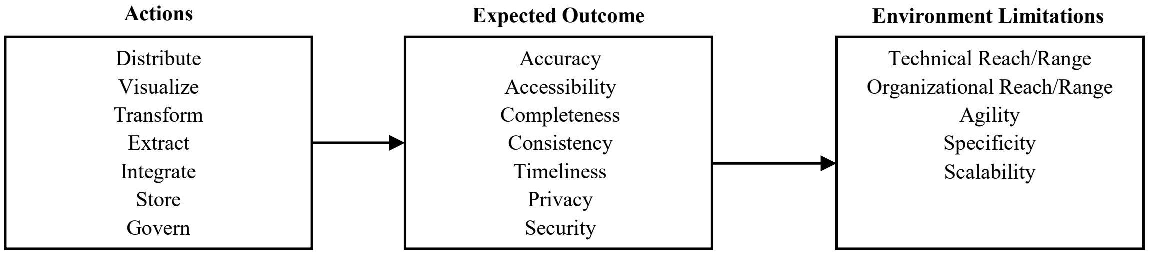

Figure 4. Schema for BIA capabilities

ness units and systems that are connected by the infrastructure and *range* refers to the depth of the integration (e.g. from a simple exchange of text messages to interwoven joint transactions). When applied to BIA, the construct of *reach and range* can be further broken down into a technical part, which adheres to the supported software sub-systems (e.g. a centralized DW, dislocated cloud services, or operational edge processing), and an organizational part that addresses the structure variety of supported units (e.g. a single business unit, a company, or a loosely-coupled supply chain). *Agility* addresses the ability to efficiently react on foreseen and unforeseen changes in the infrastructure environment and is a key trait when trying to use BIA as a dynamic capability that is supposed to ensure sustained success. This concept has extensively been scrutinized in the realm of BIA with respect to its business relevance, architectural consequences, and organization [40-42]. Moreover, literature [43, 44] and the conducted exploration indicate that the *specificity* and *scalability* are essential factors when choosing an adequate architectural component. S*pecificity* helps to cover factors like portability of the solution to other industries (e.g. if a transformation involves a highly specific risk calculation or a rather generic outlier detection). S*calability* has a large impact on architectural decisions when it comes to the degree of integration (e.g. are results only used for onetime data exploration or are they regularly fed into other systems for further use) as well the quantity structure of a solution (i.e. number of data sources and consumers).

Figure 4 visualizes the refined framework with the environment limitations introduced above. This extended schema allows us to formulate capabilities with the following layout: *"we need to [action] to get insights with [expected outcome] in order to support [business task] and be able to achieve it in an environment with [environment limitations]"*. Accordingly, the above example might be extended to: "*We are able to extract and store stock data to get insights with high accuracy and medium timeliness for the support replenishment decisions in logistics in an environment with many distributed systems and various organizations that rapidly change.*".

With this refinement the framework represents a generic way to formulate BIA capabilities. Note that (i) a formulation requires a thorough understanding of the intended applications and their benefits (e.g. supported by concepts that capture and structure BIA value, success, maturity, or contingencies) and (ii) although the capabilities already read like requirements, they do not lead to a full systems requirements specification. In fact, they are just framing and guiding valid architecture specifications with respect to architectural priorities and goals. Hence, this concept gives a structure for discovery of more concrete BIA requirements.

6. Evaluation

6.1. Case 1: Car Sharing Fleet Management

The organization in the first case is a leading car sharing company with over 2 million customers that operates over 14,000 vehicles in eight countries. The strategic business priority for its fleet management is to maximize the availability of rentable vehicles under efficiency constraints. This entails the relocation and maintenance of vehicles, issues with non-rentable entities, claim management, as well as the car rollout and the lifecycle management. The organization faces challenges in their BIA architecture as it is planning to integrate an increasing number of highly dynamic operational data sources, like sensors in and around vehicles, with dispositive data from traditional management support systems in order to implement new scenarios like real-time customer support or predictive maintenance.

A major prerequisite for the business model is a *real-time monitoring of the fleet* in order to be able to control actions and quickly react on upcoming issues. For this purpose, data from active vehicles, customer accounts, and business transactions has to be integrated. This is particularly challenging as many systems are geographically distributed entities, i.e. active rentals, which can quickly appear or vanish. The current monitoring environment is a rather isolated application and not embedded in an overarching data pipeline, which makes it hard to correlate with data from other systems. Translated to a capability this issue can be formulated as in C1.1.

C1.1: *We need to extract, integrate and visualize data to get insights with high timeliness, high completeness, and without violating privacy in order to support real-time monitoring of the fleet under consideration of the idiosyncratic requirements of our consumer-oriented fleet management environment (high business specificity).*

When vehicles show unusual behavior, the fleet management department needs to be able to conduct a *root cause analysis*. In this case, domain experts explore data from a few specific systems, e.g. data coming from a vehicle's CAN bus. This is a very specific requirement that has no relation to other parts of the data pipeline or organizational units. This leads to the following capability.

C1.2: *We need to extract, integrate and transform data to get insights with high accuracy, high completeness, and high consistency in order to find root causes of issues with vehicles in an environment with highly specific transactional system.*

In the course of claim management, the fleet management department is responsible for the *controlling of partner garages* (which spans organizations and systems). This mainly involves checking, approving, and reporting costs for maintenance and repairs. These tasks require an entire data pipeline to provide reliable and clearly understandable and accessible information, as all missing, wrong or misunderstood information costs real money. Formulated in our capability schema this can be stated similar to C1.3.

C1.3: *We need to extract, store, integrate, transform, and visualize data to get insights with high accuracy, high accessibility, high completeness, and high consistency in order to control costs at partner garages in an environment with organizational distributed systems and high governance limitations.*

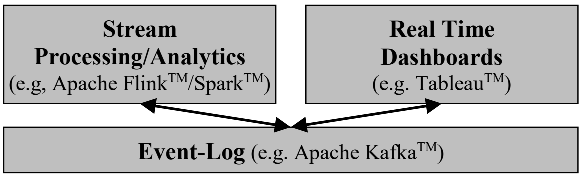
Stream Processing/Analytics
(e.g, Apache Flink™/Spark™)
Real Time Dashboards
(e.g. Tableau™)
Event-Log (e.g. Apache Kafka™)

Figure 5. Simplified Architecture in Case 1

Considering these three capabilities, it becomes apparent that the requirements for a BIA architecture in this domain are highly diverse depending on the application, and it does not seem like there is a unified solution. As a consequence, it seems reasonable to follow a decentralized architectural approach, e.g. with micro-services and an event-logs (architecture components), e.g. based on the Apache Kafka™ framework as a concrete messaging middleware (product for realizing the components) [45].

A1.1: *Using decentralized architecture with an event-log as a middleware.*

Moreover, the need to quickly process data from various sources in order to minimize the time to respond to an incident. This can be achieved with stream processing which can be realized with products like Apache Flink™ and/or Apache Spark™ which have functionality to feed real time dashboards (e.g. implemented with Tableau™) [46].

A1.2: *Using stream coordinator and a stream analytics tool for stream processing.*

Figure 5 illustrates a simplified sketch of the derived architecture based on A1.1 and A1.2 that was discussed with the stakeholders of the organizations. The general validity of the approach can be confirmed as the company's architects agreed with the capabilities C1.1 – C1.3. It was noted that not all parts of the formulations were immediately understandable and could be more concrete, e.g. instead of saying a "highly specific environment" it would be more useful to say a "highly specific environment that deals with specific data sets generated in a car". The derived architecture designs were also confirmed as valid from the architects. The company even decided to realize a similar architecture that also contains Apache Kafka™ as a focal point of a future BIA landscape.

From Case 1 we primarily took that it is recommendable to derive and/or gather concrete instantiations of the framework for specific settings.

6.2. Case 2: Port Traffic Management

The subject in this case was a port authority that manages all administrative concerns of one of the largest sea ports in Europe. Among other things, traffic management is a key task to ensure a smooth daily operation of the port. This encompasses planning, monitoring, and reporting of traffic on thousands of seaways, railroads, and streets. The BIA landscape has to process structured data from logistic systems but also IoT data from roads, traffic lights, bridges, and other traffic participants. The business goal in this case was the creation of an integrated port traffic management that allows to monitor and control the traffic flow of all participants in the port (i.e. ships, cars, trucks, trains).

One major requirement is a real-time road traffic management that allows to quickly react to unforeseen events like accidents or construction sites. Currently, road traffic is monitored with induction loops under the streets, visual tracking with cameras, as well as manual counting that is used to estimate volume of traffic. In the future, these data sources could be complemented with information coming from smart cars and trucks, or from smartphones of their drivers. This data has to be integrated and scanned for trends and abnormal patterns in real time. The corresponding capability can be formulated similar to C2.1.

C2.1: *We need to extract, integrate, and visualize data to get insights with high timeliness, medium accuracy and consistence, and high privacy in order to identify major trends and issues in the port traffic in an environment with various heterogeneous and distributed IoT systems.*

Another important task of a port authority is the provision of public traffic reports that present the number of vehicles in the port, the utilization of roads and sea lanes, as well as the flow of goods. For this, heterogeneous data from various departments have to be consolidated and transformed into an analytics-ready format. Here, most source systems already provide some kind of cleansed data sets, however, the integration has to be flexible, since the structures in the source systems frequently change. Moreover, the results are directly used for subsequent actions and need to meet several governance restrictions. In some cases, it is essential that data will not leave its source system. Capability C2.2 reflects these requirements.

C2.2: *We need to extract, store, integrate, transform data to get insights with high accuracy and high completeness, and high consistence to create overarching traffic reports in an environment with a medium technical and organizational reach, high agility in source systems and restrictive governance rules.*

The formulated capabilities touch different parts of the BIA landscape in this organization. Accordingly, they can be translated to two isolated architectural designs A2.1 and A2.2. C2.1. comes with the need for real time analytics that is needed to quickly react on current trends in the data. Thereby, the results do not have to be perfectly accurate and storage is less important. This all indicates the use of a complex event processing system. For instance, a stream processing tool like Apache Flink™ can be used to build pipelines and scan data for certain patterns and abnormalities [47].

A2.1: *Using stream processing tool for complex event processing.*

The crux of second capability C2.2 is the integration part, as there are manifold operational systems affected and governance rules prevent traditional ETL jobs that extract data and store it in a separate database. A flexible solution to these issues can be a virtual DW (e.g. implemented with Denodo™) that integrates various data sources in a virtual layer [48]. The benefit of this approach is the possibility to directly forward queries to source databases without saving operational data permanently. This can help to fulfill the governance restrictions. Moreover, the virtual layer allows to quickly react on structural changes, e.g. by including custom on-the-fly transformations or the provision of multiple views.

A2.2: *Establish a virtual data warehouse that integrates data from various departments.*

The evaluation of the capabilities and the corresponding architectural designs with the simplified architecture sketch in Figure 6 was met with general consent. The capabilities were mostly confirmed. The architectural designs, however, provoked mixed reactions. Especially A2.2 was heavily discussed and the experts in the organization pointed out that a virtual data warehouse could not meet their performance requirements and does not fit in the existing technology stack. The general fit of this architecture option to the formulated capability was not criticized, though.

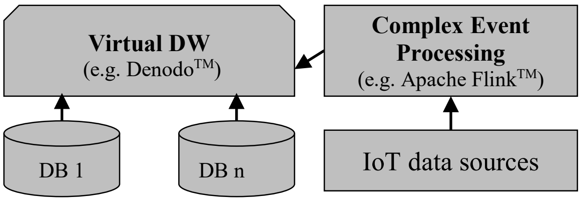

Figure 6. Simplified Architecture in Case 2

The case showed that it is hardly feasible to come up with direct, formal, or even automatic derivation of architecture options, but that the value of the framework lies in guiding discussions as well as collecting and structuring best practices.

7. Discussion

The exploration in this research confirmed the lack of systematic and business-oriented methods for architectural decision making in BIA landscapes. The interviewed BIA architects acknowledged that many architectural decisions are driven by implicit knowledge, personal experience and preferences, or vendor and legacy constraints. This makes architectural decisions hard to comprehend and often leads to inferior results in the long term. In one case, for instance, a lead developer chose a certain NoSQL database because of his personal interest in the technology. The organization then built a product around this technology, which hit a wall when it came to an international rollout which required a new level of scalability. If this architectural decision would have been discussed from a business perspective, a global use would have been a key requirement in the first place and an adequate scalability might have been considered when choosing a solution. Next to this possible improvement of decision quality, it also turned out that a systematic approach would help to justify decisions when it comes to funding initiatives or auditing existing solutions.

The practical evaluation in Section 6 then confirmed a general applicability of the capability framework derived in Section 5. However, it also illustrated challenges and limitations of the approach. Especially, that the manifold different environment factors and the variety of possible architectural consequences are hard to reconcile in one concept. Therefore, it is important to keep in mind that the goal of the proposed framework cannot be production-ready solutions, but rather the preparation of fundamental architectural requirements that lay the foundation for further discussions. How this can look like in practice, became clear in the discussion of C2.2 and the corresponding architectural imitative A2.2 which led to a lively debate about concrete implementation details, whereas the underlying requirement was accepted by all parties.

Regarding the proposed framework, the natural formulation of the capabilities made them easily understandable. Moreover, the dimensions used for expressing actions, expected outcomes, and environment limitations were sufficient to cover the requirements in the two cases above. The evaluation of the formulated capabilities in Section 6, however, showed that it can be hard to choose the right level of abstraction. A too vaguely formulated capability is hard to translate into concrete actions, and a too precise formulation leads to unrewarding discussions about implementation details. Another challenge of the application of the framework is the translation of derived capabilities into concrete architecture designs with specified products. For the initial evaluation of the framework, we used literature and personal experience to identify adequate architectural components and products. Here, it would be helpful to have a catalog of reference capabilities and corresponding architectural concepts to draw on. This would minimize the subjective influence in the processes and enable a look beyond individual experiences as well as the consideration different implementation possibilities. As a first step towards such a catalog, we created a public online repository to collect analytics capabilities, corresponding architectural designs and adequate tool choices. Next to the further exploration and generalization of our framework, this public repository can also help to gain insights in current BIA challenges and identify architectural best practices.

8. Conclusion

The presented research illustrates the need for systematic and business-oriented methods for architectural decision making in BIA landscapes. The goal of the capability-based framework derived in Section 5 is to realize this with a schema for formulating BIA capabilities. The case-based evaluation in Section 6 has then illustrated the applicability of the framework and helped to identify benefits and limitations of our approach. It turned out that the framework particularly supports the extraction of necessary capabilities for strategic goals that have either already been realized or at least been partially designed and that these capabilities can then be translated to fundamental architectural designs which serve as a starting point for further discussions. The evaluation also showed that it might be necessary to derive finer grained instances of the capability schema for certain application domains and environments, e.g. “IoT-based manufacturing”. Secondly, due to the variety of available tools, the translation of capabilities into architectural designs can become cumbersome and there is a need for a reference catalog to support this step. This is why we created a public online repository where we collect capabilities and corresponding architecture design options to provide a guidance for a practical use.

The research contributed to the body of knowledge by translating the capability-concept into a more precise and defined schema and by exploring its adaptation in the BIA sector. Regarding the practical use, the framework can be used to a) translate business goals to architectural designs by systematically deriving necessary capabilities or by b) facilitating the design of new business models by pinpointing what capabilities are available and what actions they support. This is also a starting point for further enhancements and adaptations to other domains. The two companies from Section 6 followed path a) and used the approach to choose BIA architecture building blocks and suitable supporting technologies for a previously formulated strategy. We have so far not yet followed route b), i.e. derived for new strategic goals based on existing capabilities. While this is clearly a creative process, the framework can guide questions like: “*What other applications can be supported with the derived BIA actions, outcomes, and environment limitations?”* or *“What applications might be supported if the capability is slightly altered?”*.

Further research should focus the following three aspects: 1) possible domain/environment specific instantiations/specifications, 2) the scales used to measure the individual construct, and 3) a more formalized process of translating capabilities to concrete architectural components, patterns and products. By setting up an open GitHub^{TM} repository for a community-driven collection of relating data, we took the first steps to close these gaps. We aim at collecting as many capabilities, architectural designs and solutions, as well as experiences regarding their suitability as possible in order to identify patterns and best practices and come up with a robust method that reaches from the strategy to the set of individual tools.

9. References

[1] Seddon, P.B., D. Constantinidis, T. Tamm, and H. Dod, "How does business analytics contribute to business value?", Information Systems Journal, 27(3), 2017, pp. 237-269

[2] Shanks, G., R, Sharma, P., Seddon, and P. Reynolds, "The Impact of Strategy and Maturity on Business Analytics and Firm Performance: A Review and Research Agenda", Australasian Conference on Information Systems (ACIS), 2010, p51.

[3] Serrato, M. and J. Ramirez, “The Strategic Business Value of Big Data” in Big Data Management, García M.F., Lev B. (eds.), Springer, Cham, pp. 47-70.

[4] Wamba, S.F., S. Akter, A.J. Edwards, G. Chopin, D. Gnanzou. “How ‘big data’ can make big impact: Findings from a systematic review and a longitudinal case study”, International Journal of Production Economics, 165, 2015, pp. 234-246.

[5] IEEE. “ISO/IEC/IEEE 42010: Systems and software engineering - Architecture description”, 2011.

[6] Zachman, J.A. “A framework for information systems architecture”, IBM systems journal, 26(3), 1987, pp. 276-292.

[7] The Open Group. “TOGAF Version 9.2”, 2018.

[8] Rouhani, B.D., M.N. Mahrin, F. Nikpay., and P. Nikfard. “A systematic literature review on Enterprise Architecture Implementation Methodologies”, Information and Software Technology, 62(1), 2015, pp. 1-20.

[9] Industrial Internet Consortium. “The Industrial Internet of Things - Volume T3: Analytics Framework”, 2017.

[10] Shariat, M. and R. Hightower. “Conceptualizing business intelligence architecture”, Marketing Management Journal, 17(2), 2007.

[11] Chen, H., R.H. Chiang, and V.C. Storey. “Business Intelligence and Analytics: From Big Data to Big Impact”, MIS quarterly, 36(4), 2012, pp. 1165-1188.

[12] Chaudhuri, S., U. Dayal, and V. Narasayya. “An overview of business intelligence technology”, Communications of the ACM 54(8), 2011, pp. 88-98.

[13] Golfarelli, M., S. Rizzi, and I. Cella. “Beyond data warehousing: what's next in business intelligence?”, 7th ACM international workshop on Data warehousing and OLAP, 2004.

[14] Baars, H. and H.G. Kemper. “Management Support with Structured and Unstructured Data - An Integrated Business Intelligence Framework”, Information Systems Management, 25(2), 2008, pp. 132-148.

[15] Ong, I.L., P.H. Siew, and S.F. Wong. “A five-layered business intelligence architecture”, Communications of the IBIMA, 2011.

[16] Watson, H.J. and B.H. Wixom. “The current state of business intelligence”, Computer, 40(9), 2007, pp. 96-99.

[17] Inmon, W.H., Building the data warehouse, John Wiley & Sons, New York, USA, 2005.

[18] Golfarelli, M. and S. Rizzi. “From Star Schemas to Big Data: 20+ Years of Data Warehouse Research”, A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, S. Flesca, et al. (eds,), Springer, Cham, 2017, pp. 93-107.

[19] Leonard-Barton, D., Wellsprings of knowledge: Building and sustaining the sources of innovation, Harvard Business School Press, Boston, MA, USA, 1995.

[20] Penrose, E.T., The Theory of the Growth of the Firm, Oxford University Press, Oxford, USA, 1995.

[21] Winter, S.G.. “Understanding dynamic capabilities”, Strategic management journal, 24(10), 2003, pp. 991-995.

[22] El Sawy, O.A. and P.A. Pavlou. “IT-enabled business capabilities for turbulent environments”, MIS Quarterly Executive, 7(3), 2008, pp. 139-150.

[23] Teece, D.J., G. Pisano, and A. Shuen. “Dynamic capabilities and strategic management”, Strategic management journal, 18(7), 1997, pp. 509-533.

[24] Eisenhardt, K.M. and J.A. Martin. “Dynamic capabilities: what are they?”, Strategic management journal, 21(10/11), 2000, pp. 1105-1121.

[25] Keller, W.. “Using capabilities in enterprise architecture management”, White Paper, 2009.

[26] Venkatraman, N., J.C. Henderson, and S. Oldach. „Continuous strategic alignment: Exploiting information technology capabilities for competitive success”, European Management Journal, 11(2), 1993, pp. 139-149.

[27] Wamba, S.F., A. Gunasekaran, S. Akter, S.J. Ren, R. Dubeye, and S.J. Childe. “Big data analytics and firm performance: Effects of dynamic capabilities”, Journal of Business Research, 70, 2017, pp. 356 - 365.

[28] Cosic, R., G. Shanks, and S. Maynard. “Towards a business analytics capability maturity model”, Australasian Conference on Information Systems (ACIS), 2012.

[29] LaValle, S., E. Lesser, R. Shockley, M.S. Hopkins and N. Kruschwitz. “Big data, analytics and the path from insights to value”, MIT Sloan Management Review, 52(2), 2011, pp. 21-32.

[30] Davenport, T.H., J.G. Harris, D.W. De Long, and A.L. Jacobson. “Data to Knowledge to Results: Building an Analytic Capability”, California Management Review, 43(2), 2001, p. 117-138.

[31] Hevner, A.R. “A three cycle view of design science research”, Scandinavian journal of information systems, 19(2), 2007, p. 87-92.

[32] Myers, M.D. and M. Newman. “The qualitative interview in IS research: Examining the craft”, Information and organization, 17(1), 2007, pp. 2-26.

[33] Love, J. and R. Hirschheim. “Crowdsourcing of information systems research”, European Journal of Information Systems, 26(3), 2017, pp. 315-332.

[34] Analytics Capabilities Collection, GitHub, https://github.com/uni-stuttgart-wi1/analytics-capabilities-collection.

[35] Wand, Y. and R.Y. Wang. “Anchoring data quality dimensions in ontological foundations”, Communications of the ACM, 39(11), 1996, pp. 86-95.

[36] Pipino, L.L., Y.W. Lee, and R.Y. Wang. “Data quality assessment”, Communications of the ACM, 45(4), 2002, pp. 211-218.

[37] DAMA UK. “The six primary dimensions for data quality assessment”, Technical report, 2013.

[38] Keen, P.G.W.. “Information technology and the management difference: a fusion map”, IBM Systems Journal, 32(1), 1993, pp. 17-39.

[39] Conboy, K., “Agility from first principles: reconstructing the concept of agility in information systems development”, Information Systems Research, 20(3), 2009, pp. 329-354.

[40] Krawatzeck, R. and B. Dinter. “Agile Business Intelligence: Collection and Classification of Agile Business Intelligence Actions by Means of a Catalog and a Selection Guide”, Information Systems Management. 32(3), 2015, pp. 177-191.

[41] Zimmer, M., H. Baars, and H.G. Kemper. “The impact of agility requirements on business intelligence architectures”, Hawaii International Conference on System Science (HICSS), 2012.

[42] Knabke, T. and S. Olbrich. “Understanding Information System Agility - The Example of Business Intelligence”, Hawaii International Conference on System Sciences (HICSS), 2013.

[43] Bakshi, K.. “Considerations for big data: Architecture and approach”, IEEE Aerospace Conference, 2012.

[44] Khine, P.P. and Z. Wang. “A Review of Polyglot Persistence in the Big Data World”, Information. 10(4), 2019, pp. 141-165.

[45] Kleppmann, M., A.R. Beresford, and B. Svingen. “Online Event Processing”, ACM Queue, 17(1), 2019, pp. 40-62.

[46] Salloum, S., R. Dautov, X. Chen, P.X. Peng, and J.Z. Huang. “Big data analytics on Apache Spark”, International Journal of Data Science and Analytics, 1(3), 2016, pp. 145-164.

[47] Carbone, P., A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas. “Apache flink: Stream and batch processing in a single engine”, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 36(4), 2015, pp. 28-38.