

Marco POSPIECH, Carsten FELDEN
Institut of Information Systems TU Bergakademie, Freiberg

CHANGING BI PARADIGMS REFLECTING SERVICE ORIENTED BUSINESS INTELLIGENCE

Summary. Business Intelligence promises improved decision making. To meet the changing business requirements, BI has to reduce latency between data acquisition and decision. As a result of growing needs, concepts like Real Time, Active, Operational, Business Performance Management or embedded BI are introduced in the literature. Nevertheless, the technical implementation prevents the realization of these concepts. Thus, overloaded infrastructures and heterogeneous system landscapes avoid further evolution. To prevent this lack of system integration and performance, we address the changing BI paradigms through a service-oriented approach. Hence, the novel BI concepts and Service-oriented Business Intelligence (SoBI) theories are illustrated and combined in a uniform artifact. The construct is developed by using reference modeling. As a result, a SoBI reference architecture accrues, which becomes validated successfully and provides a basis for future implementations. This paper offers a flexible approach that reduces the latency in order of dynamic market environments.

Keywords: Business Intelligence, Service-oriented Architecture, Operational, Embedded.

ZMIANA PARADYGMATÓW W ZAKRESIE INTELIGENTNYCH SYSTEMÓW KOMPUTEROWYCH UKIERUNKOWANYCH NA KLIENTA

Streszczenie. Inteligentne systemy komputerowe pomocne są w procesie wspomagania decyzji do zmieniających się wymagań biznesowych. System BI ma zmniejszyć opóźnienie między procesami akwizycji danych i podejmowania decyzji. Niemniej w praktyce trudna jest strona techniczna dotycząca realizacji tych koncepcji. Aby uniknąć problemów związanych z integracją i wydajnością systemu, możemy dostosować się do zmieniających się paradygmatów BI przez podejście zorientowane na usługi. Stąd nowa koncepcja BI zorientowana na usługi Business Intelligence. W publikacji przedstawiono elastyczne podejście, które zmniejsza opóźnienia w dynamicznym środowisku rynkowym.

Słowa kluczowe: systemy inteligentne, architektura zorientowana na usługi, operacyjny.

1. Introduction

Dynamic markets force companies to reduce production cycles, identify early trends, and react fast [1]. Usually, the value of an action is higher the earlier the action is taken. To reduce such delitescence, Business Intelligence (BI) attempts to provide the right information at the right time to the right place for the right purpose to support decision making [1] the time between the originating business event (for example a sales activity) and the decision is called latency and should be as small as possible. It can be divided into data acquisition, analysis, and decision latency. [2] Various concepts have been introduced to minimize latency. E.g. Real Time BI decreases the time between data origin and loading into the data warehouse (DWH) by immediate loading procedures. Active BI addresses the decision latency directly and the analytical focus on data has changed in the recent years, too. The need of analyzing operational [3] and process data has been evolved. Thus, decisions can be done in the moment a business event occurs [4]. Consequently, analytical capabilities must be expended towards operational users and processes to ensure an analytical approach throughout the company and an accelerated decision finding [5]. This requires a flexible and more sophisticated (compared to traditional ETL) enterprise wide data integration [6]. To solve this issue, a recommendation can be illustrated through a reference model. Reference modeling as method enjoys a wide distribution to support those activities [7]. Maturity model research done by Chamoni and Gluchowski [8] or Schulze et al. [9] shows that neither the architectural concepts mentioned above nor reference architectures are implemented or used in most of the companies, yet. The papers goal is to introduce the initial steps of a BI reference model recommendation to address the reduction of latencies between business cases and decision. This should support changing analytical views and enables a flexible enterprise wide integration to support competitiveness in dynamic markets.

We believe in a valuable combination of BI and Service-oriented architectures (SOA) through reference modeling to address the stated issues. SOA can be understood as „... a way of designing and implementing enterprise applications that deals with the intercommunication of loosely coupled, coarse grained (business level), reusable artifacts (services). Determining how to invoke these services should be through a platform independent service interface ...“ [10]. SOA enhances BI with a vendor independent access to all systems by services that pass real time data through the DWH and analytical functions into processes and operational applications. In this spirit, the paper contributes to the discussion about SoBI to introduce capabilities for changing analytical views and reduced latencies in decision making processes.

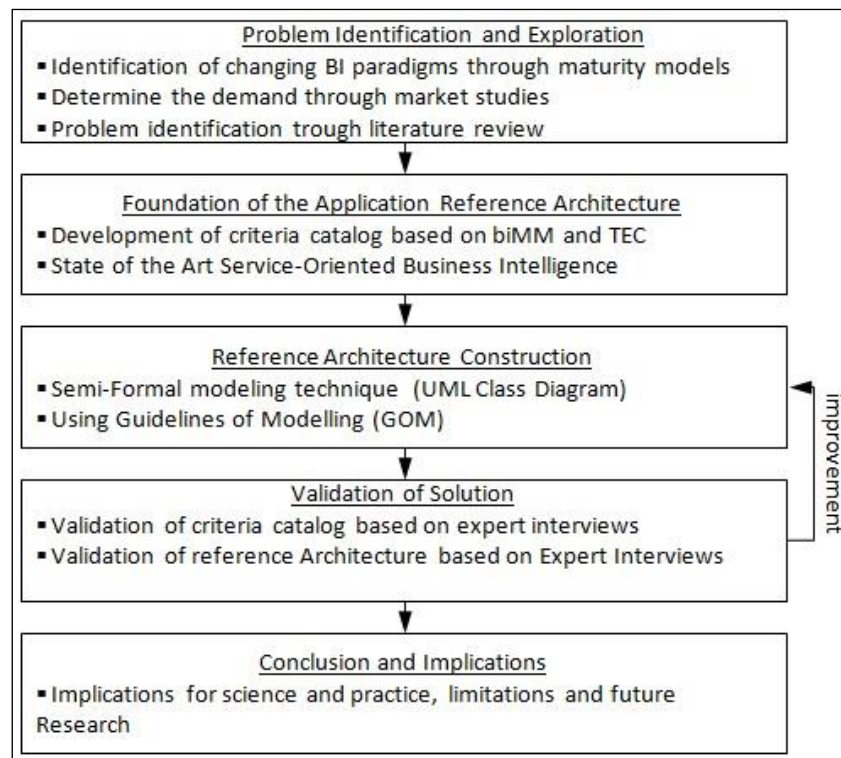


Fig. 1. Research Design

Source: Modified [11].

Rys. 1. Schemat realizacji badań

Źródło: poprawione [11]

The reference architecture's development follows the research framework of [11] (see Figure 1). Their procedural method is designed for conceptual reference architectures and contains important aspects like literature review, requirements analyses, design methods and validation techniques to allow a rigorous research. The architecture traverses several development cycles and contains deductive and inductive research elements. Relevant BI concepts have to be identified and illustrated based on several maturity models. The problem identification of discovered BI concepts occurs through a literature analysis. The second phase contains the foundation of the reference architecture. A requirement analysis is done in order to identify fulfilling criteria. Furthermore, a SoBI-State-of-the-Art is presented as fundament of the proposed reference architecture. The resulting architecture shows relationships, attributes and operations of elements, and provides expansion capabilities [7]. To gain model quality, well accepted guidelines of modeling (GOM) [12] are applied. The validation of the proposed architecture occurs through expert interviews, implications, limitations and further research are presented.

2. Problem identification and exploration

The enhanced BI concepts are selected by existing maturity models of [8, 13]. This paper considers actual market surveys of the TDWI to support the importance of the selected concepts. Due to the reason that all studies are done by a single research institute, which used similar survey methods, the results are comparable. In this context, Real Time BI, Active BI, Operational BI, Embedded BI, Business Performance Management (BPM) and Enterprise wide DWH were selected [14, 15, 16, 17].

All concepts are functionally driven and require a technical implementation. But especially the technical realization prevents the implementation of the identified concepts. Real Time BI contains a timely data collection, but more than 91 percent of all companies use time consuming batch processes in BI (ibid.). ETL itself is unsuitable for time-critical dynamic environments. [4] There are two alternatives [18]. Change data capture (log-based import trigger procedure; push oriented data supply) and materialized views (pull oriented data supply). Triggers require changes in databases and initiate a performance penalty to the source system each time a trigger fires. The drawback of materialized views belongs to the matching of different views, whose updates and their availability during actualizations. Active BI requires systems that detect and evaluate events. These systems respond in a synchronous or asynchronous way and trigger actions in processes or external systems. Here, an enterprise wide system integration and real time ahead is required. [19] BPM controls processes on base of analyzed and monitored data or utilizes the gathered information in other calculations. Especially the consistence between process and system level is still a problem. Thus, a system's cross linking is solved insufficient. [20] Such a lack of connectivity also prevents embedded BI [21]. Larger companies own approximately 50 different IT systems, where most of them are linked by manually coded point-to-point connection. [22] Point-to-point connections will lead to more complexity and reduced data quality, if data are transferred between systems. (ibid.) Additionally, BI requires a flexible platform that allows an easy integration and modification of components, provides a high data quality, enables operational and process analysis, works in real time, enables active decision finding, and offers analytical functions in external systems and processes.

3. Formation of a reference architecture

The architecture tackles the reduction of a decision latency. The goal requirements have to be detailed through a requirement analysis. One accepted method belongs to the reuse and consolidation of established BI criteria catalogs. [23] We use the accepted [23, 8] catalogs

Technology Evaluation Center (TEC) [24] and Business Intelligence Maturity Model (biMM) [9]. This leads to 909 considered requirements. All requirements are triple blind peer-reviewed and reduced to verifiable elements based on the selected BI concepts. Only conceptual aspects can be proofed against suitability. The chosen criteria are proofed against importance through expert interviews iteratively and reduced to 27 requirements. Since the enhanced BI concepts contain overlapping content, a decision by majority was made to which concept a criterion belongs to. The selected requirements are presented in Table 2 (p. 8)

According to Cooper, a literature review represents a research method and is defined as a number of primary examinations with the similar questions and do not provide own new findings concerning to the research field [25]. In order to identify relevant papers, the following academic databases, e.g. ACM Digital Library, Science Direct, IEEE, Xplore, SpringerLink, Business Source Complete, Elsevier Engineering Village, Academic Search Complete, were used. To ensure comparability, search items like Service oriented Business Intelligence, Service oriented BI and SoBI were requested in quotation marks and limited to abstract, title, or keywords. As a result, 1,552 hits regarding the search term have been obtained. The literature evaluation serves in favor of proofing and securing quality and relevance. In an initial step, duplications and unserious papers were removed according to a four-eyes-principle. This leads to ten remaining articles. We also conducted a backward-search to avoid missing relevant articles [26]. In sum, 12 articles were identified (table 1). Authors, mentioned concepts and level of detail are compared.

Table 1

Results State-of-the-Art

Author	Enterprise Wide DWH	OpBI	Real Time BI	BPM	Embedded BI	Active BI	ETL Service	Analytical Service	level of detail
[27]			x				x	x	low
[28]							x		low
[29]		x	x	x	x	x	x	x	low
[30]	x					x	x	x	low
[20]	x	x	x		x	x	x	x	low
[31]	x		x	x	x	x	x	x	low
[32]	x	x	x				x	x	low
[22]	x	x	x	x	x	x	x	x	low
[33]	x	x	x	x		x		x	low
[21]	x	x	x	x	x		x	x	low
[34]		x	x		x	x			low
[35]	x	x	x		x	x	x	x	low

Source: Author's own.

First SoBI ideas can be traced back until 2005. Since the origins, the essential approach of establishing a SoBI is a service bus. This bus connects all components, guarantees a universal access and therefore any system access to BI. The main tasks are messaging, data transformation, and intelligent routing of messages. According to [35], the service bus includes both, the event-driven asynchronous communication by using publish-and-subscribe as well as synchronous service communication, which follows a request-response model. The bus acts as an intermediary between service users and service providers and enables the communication between service and event. In this context, a service reacts to or produces new events. Thus, the combination of both communication models enables a timely reaction of occurring events and can trigger multiple services in a simultaneous and asynchronous way. Hence, all business events are located in the service bus and can be processed in real time. (ibid.) According to [31], all BI functionalities have to be encapsulated into services, so that BI can act either as service user or service provider. Analytical services are encapsulated modules. They provide analytical business logic and correspond to classical BI tools [22]. Operational, tactical, and strategic data is provided by Software services. These services are the fundament of BI analyses [1]. In this context, services can be combined in a flexible way reflecting the business needs. Such an assembling is called orchestration. Software services can be divided into Data Access, Transformation and Infrastructure services [20]. The Data Access service includes four basic operations, namely create, read, update and delete. Consequently, the service combines abilities of reading and filling systems. [22] Transformation services represent the encapsulated transformation phase of ETL. Thus tasks like aggregation, encoding, filtering, conversion, join and mapping have be fulfilled by services. [20, 22] There are existing cross-cutting tasks in addition to the presented services that are realized through infrastructure services. [20] These services include data security and data protection features as well as aspects of data quality, master-data, and meta-data management. [32] In this context, SoBI promises the integration of BI services into processes and applications, avoids redundant implementations, enables an unproblematic integration, improves scalability, and allows an open communication between all components.

4. Service-oriented business intelligence reference model

Reference modeling is a method that allows a formal description of knowledge and is used among others in system development. Here, the model should be on a general level to serve a huge audience, but must be extendable, to meet specific requirements. Figure 2 shows the proposed SoBI architecture in a semi-structural representation, which is successively refined by UML class diagrams. The suggested reference model moves from a traditional BI architecture to a generic SOA architecture [30].

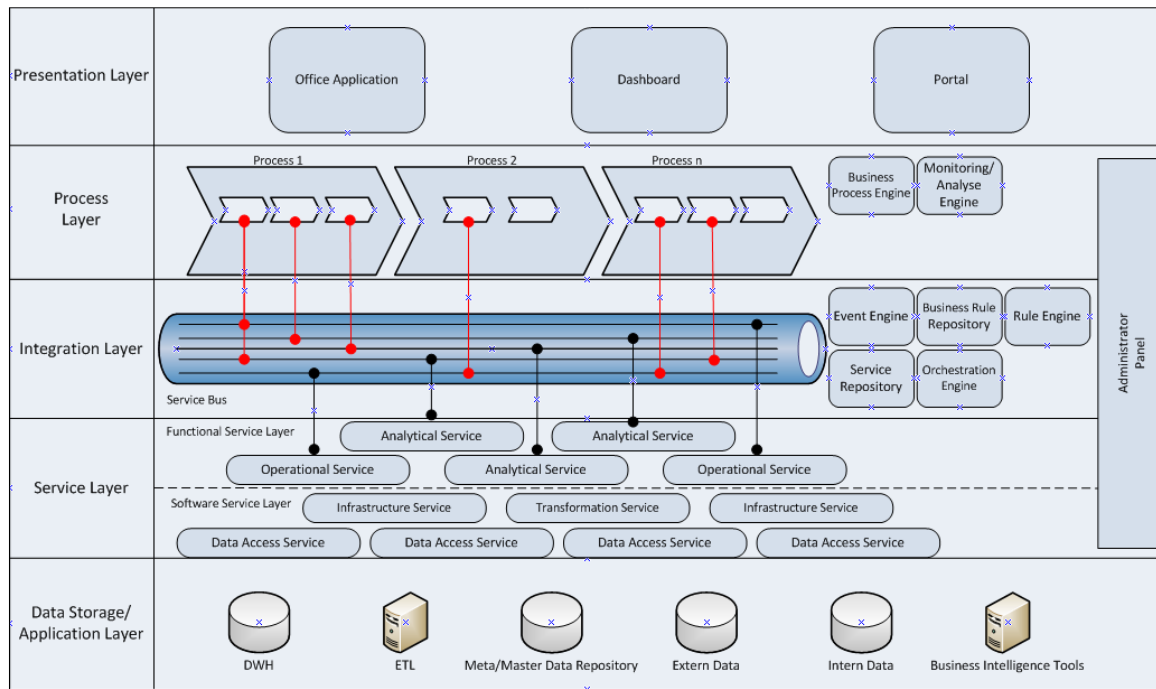


Fig. 2. Service-oriented Business Intelligence Reference Architecture

Source: Author’s own.

Rys. 2. Architektura inteligentnego system komputerowego ukierunkowanego na klienta

Źródło: praca własna autora.

The *Data Storage/Application layer* serves as a central foundation for the subsequent levels. The classes (information systems) are *participants* that can take the *role* as *service user* or *service provider*. Each information system includes operations, which allow the receiving and sending of events. Examples of such systems are the DWH, BI tools, or source systems. The event is a *ModelElement* and contains the *EventParameter* as attribute. *SetEreignisParameter* corresponds to an abstract operation, which expresses the fact that any operation leads to an event. The *event engine* is part of the *service bus* and sends and receives events from information systems. It appears adversely to follow the traditional BI architecture concept, since the DWH exists no longer in a monolithic pillar, but rather in an embedded IT infrastructure [22]. The historical data hold by the DWH must be combined and synchronized with real time information from operational systems and should flow into processes. A redundant and distributed storage can lead to individual application terminologies, this ends in inconsistencies and duplications. Due to this reason, a central *Meta/Master Data Repository* is required. It is accessible through *Data Access services* and a *service bus* [22].

The *Service layer* reflects all encapsulated functionalities of the *Data Storage/Application layer* to meet the requirements of cross-system working processes. [35] Thus, the usefulness of services should be discussed and due to this reusability should be proved [20]. In this context, the attribute can be verified for *Data Access*, *Transformation* [32], *Infrastructure* [20], *Analytical* and *Operational services* [36]. In addition, the role concept of a SOA must be integrated into the reference architecture. Hence, a *participant* can take the *role* as *service*

user or *service provider*. In coherence to this, the provider delivers the services to the user. This requires a *service contract*. The *contract* is part of the *service*, contains the *service description* and allows a *message exchange*. *Messages* expand the *ModelElementDataTyp* and implicate a container that holds the *message* in a specific format and sends or receives it. E.g. an *event* can be one *message*, but also a combination of several messages. The *service* itself can be understood as a *technical activity* and can be specialized in *Functional* or *Software services* [20]. Using the included service operations allow to do refined specializations. *Analytical services* are the core of the presented reference model and obtain transformed content from underlying services. Operating systems are part of the reference architecture and have to be designed in a service-oriented way [35] to gain their functionalities as *Operational services*. Here, operational services belong for example to scheduling, or material management [22]. The service integration occurs through the *integration layer*.

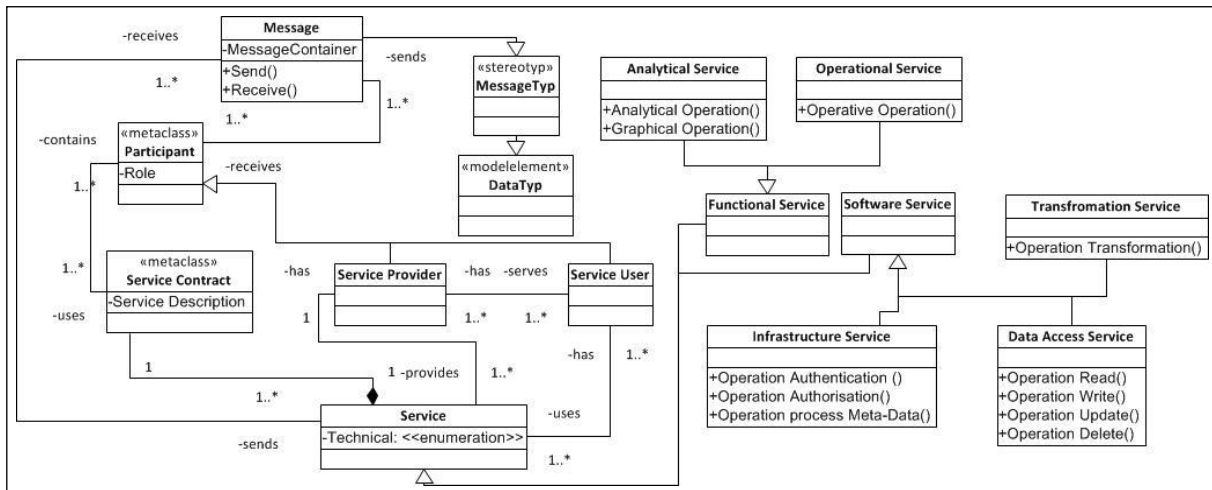


Fig. 3. The Service Layer

Source: Author's own.

Rys. 3. Warstwa usługowa

Źródło: praca własna autora.

The *Integration layer* realizes the coordination of all services and events in the total system. The *service bus* consists of *service repository*, *orchestration engine*, *rules engine*, *event engine*, *business rule repository*, and *administrator panel*. The *service repository* manages and publishes all *service descriptions*, which are presented in form of *service contracts*. Furthermore, the *service repository* selects services in cooperation with the *orchestration engine*. The selection is based on the *service description*, which makes the service available for the *orchestration engine*. The *orchestration engine* uses the operation orchestrating to execute a service sequence. A sequence contains ordinarily different service types and can be embedded in other systems or processes. These sequences are classified in context of this paper in *business* and *technical processes*. The *orchestration engine* allows just the orchestration of *technical processes*. In addition, the *orchestration engine* contains

mechanisms that allow state management, logging and monitoring of sequences. The *orchestration engine* is requested by events that are triggered from the *event engine*. Here, the *event engine* is based on publish-and-subscribe and receives and processes events from all layers, which are sent to registered users. Furthermore, the *event engine* holds analytical operations. That analysis is enriched by business process events from the *monitoring/analysis engine* [35] and requires defined business rules which are obtained from the *rule engine*. In this context, a simultaneous processing of multiple events is feasible to detect causal, temporal or spatial relationships between events, in order to identify predictive problem scenarios. These proposed functions follow the ECA rule and allow an automated decision making and execution. The *rule engine* provides business rules that are obtained from the central *business rule repository*. This prevents a redundant implementation of business rules, increases their reusability and separates the process logic from the decision logic. Business rules can be understood as expertise, management policy [22], and decision logic of processes [35]. Already modeled processes are also stored in the *business rule repository* and can be queried by the *rule engine*. The *administrator panel* presents the administrative component of the SoBI reference architecture. Accordingly, the panel creates *business rules* and *service descriptions*, which are related to the *service* and *business rule repository*, delivers a service development interface and defines human activities. In addition, the *administration panel* holds process modeling capabilities.

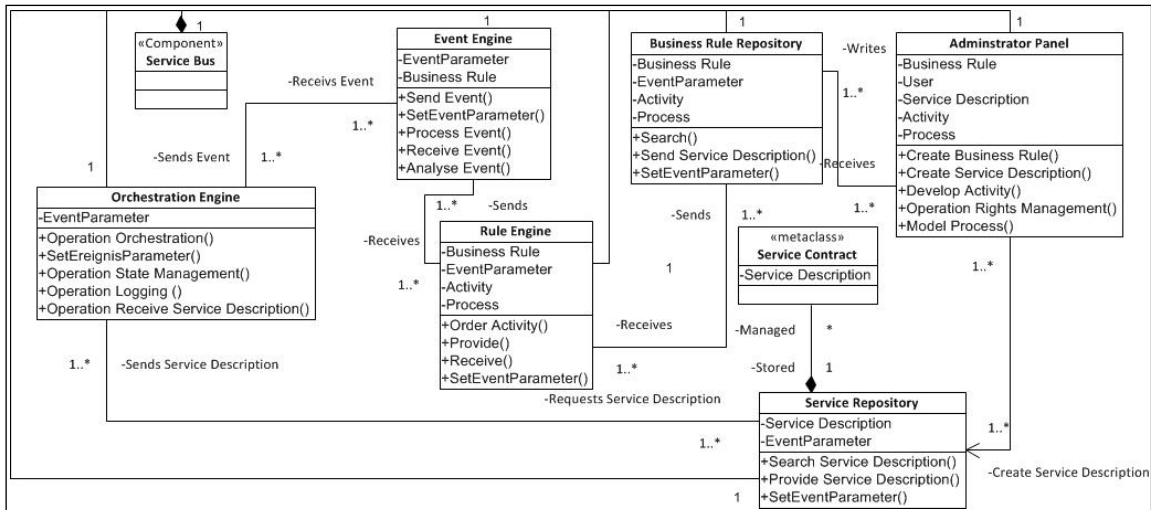


Fig. 4. Integration Layer

Source: Author's own.

Rys. 4. Warstwa integracyjna

Źródło: praca własna autora.

The central element of the *process layer* is the process. A process is formed by activities that can be executed technically or manually. In this context, a technical activity is run by a service and a manual activity is performed by a user. The second part of a process is business rules. The *rule engine* puts the activities on basis of business rules in the right order (process). We are separating between technical and business process. A business process can

hold manual and technical activities; in contrast, a technical process is composed exclusively by technical activities. In this context, the *orchestration engine* takes the responsibility of the technical process orchestration. Unlike to this, the *business process engine* performs the execution of business processes. In case of a business process the *orchestration engine* is cooperating with the *business process engine*, so the technical activities of the business process are provided by services. Processes are triggered by events from the *event engine*. Accordingly, the *event engine* sends an event to the *rule engine*, which searches for business rules by using the *business rule repository*. The result is an orchestrated process that is executed by the *orchestration engine* or performed by the *business process engine*. Alternatively, the process can be modeled in advance in the administration area. In this case, the process is provided by the *rule engine*, which obtains the process from the *business rule repository*. Therefore, the process itself is triggered by an event in the *orchestration* or *business process engine*.

Furthermore, the *monitoring/analysis engine* observes business processes and shares the results closely with the *business process* and *event engine*. Consequently, the event engine analyzes continually business process events in real time. The required process states are received by the *monitoring/analysis engine* taken from the *business process engine*. Each business process change initiates an event in the *monitoring/analysis engine*, which is passed by the *event engine* through potentials consumers. Special business process states or circumstances can be sent as well as an event and trigger defined reactions in the *event engine*. This requires business rules that are related to the *rule engine*. Thus, decisions can be automated and done in shortest time. Consequently, the *monitoring/analysis engine* enables real time BI analyses and the monitoring of critical process key figures.

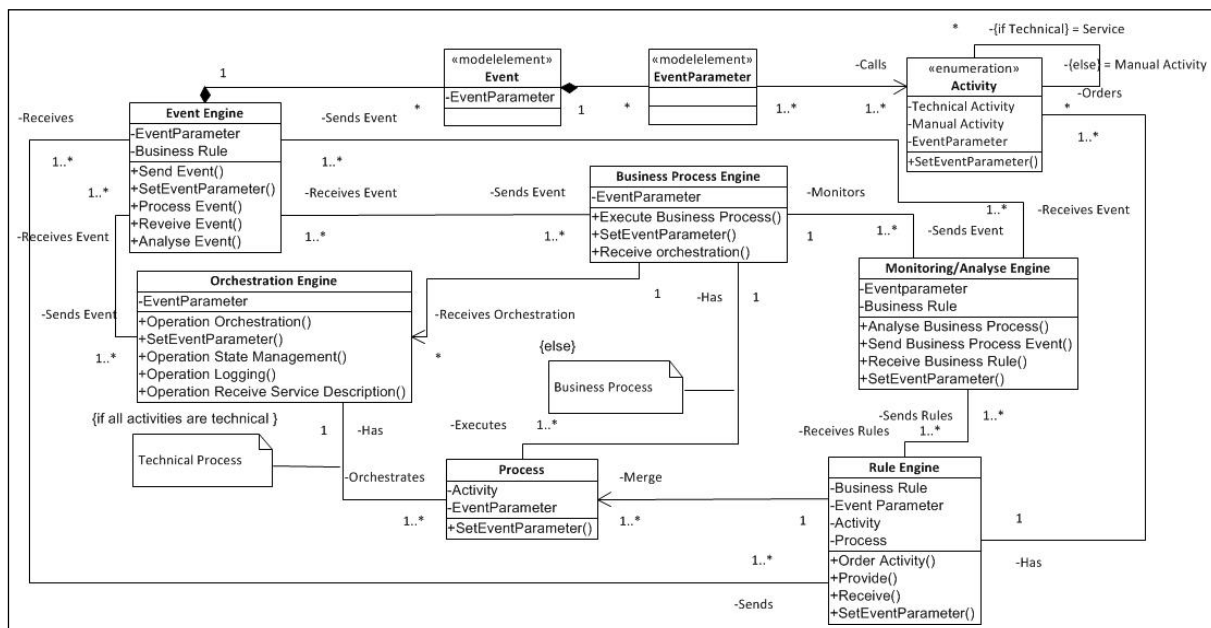


Fig. 5. Process Layer

Source: Author's own.

Rys. 5. Warstwa procesów

Źródło: praca własna autora.

The *presentation layer* enables a flexible integration of analytical information by different visualization options. The fundamentals are *analytical services*, which allow graphical representations in portals, dashboards, or office applications [35]. In this context, the classes portal, dashboard, and office application inherit from information system. Thus, the user can be informed about circumstances by event or send its decision directly to the *event engine*.

5. Validation

The validation of the proposed reference architecture follows the approach of Felden and Buder. They validated their conceptual reference model through qualitative face-to-face expert interviews. [37] This method seems to be suitable, because we are still on a conceptual modeling stage, where a software based implementation is not realized. We are also aware that expert interviews are not free of subjectivity. But to address this issue, the proceeding and suggestions for expert interviews based on [38] are used. Four practitioners and three research experts were asked in an iterative cycle of three rounds. Opinions and suggested changes about validation criteria and reference architecture were considered and modified. To ensure the qualification of each expert, pre-questions were applied. The interviews happened separately, varied from two to three hours and followed an interview guideline. The interviews were protocolled and afterwards analyzed by a four-eye principle. Afterwards, the participants checked the written conversation records for correctness. In a final round each expert was asked to validate the importance (I) of the remaining requirements and the grad of fulfillment (F) of the proposed reference architecture based on a five point (1 = strongly disagree, 5 = strongly agree) likert-scale [39]. Here, the reference model was presented and discussed. The validation results are summarized in Table 2. An argumentative-deductive validation is attached to demonstrate the realization of proposed BI concepts. The results are proofed against significance through a one tailed t-test. Here, the null hypothesis is rejected, whether the p-value fulfillment (PF) or p-value importance (PI) does not exceed the chosen significance value of $\alpha = 0.05$. The null hypothesis $H_0: \mu \leq 3.0$ $H_1: \mu > 3.0$ expresses the assumption that the experts disagree in the importance or fulfillment criteria (below 3.00).

In summary, almost all experts believe in a successful implementation of the concepts. In addition, several importance criteria are not significant. Common BI requirements like OLAP or reporting reach top marks and new concepts are demanded as well. The particular needs of the experts are apart from each other. This can be explained through different business cases. Here, the reference architecture enables a customized solution for specific requirements. Some experts state that the realization of the real time BI paradigm could be critical. "The proposed method can lead to an overload of the used bus or involved systems. Services, usually web services, are designed for small data sets so that the substitution of the ETL

process appears doubtful.” A potential compromise could be the proposed request-response model, which achieves real time partially. Another solution regards to in-memory technologies. Other experts predict a growing level of system complexity. “The emerging meta-model will lead to confusing mappings, difficult error handlings and complex maintenance. Even BI has rapid changing requirements. The administration of the Service Repository will be tough job.” At least one expert disbelieve in the reuse of services. “My experience leads to the conclusion that BI functions are to specialized to achieve a re-use in operational processes.” Future developments and case studies will reveal whether the skepticism is justified or not.

Table 2

Expert Interview Validation

Concept	Criteria	(F)	(PF)	(I)	(PI)	Validation
Enterprise-wide DWH	Universal data access method	4.14	0.00	4.57	0.00	The heterogeneous interface problems get solved by Data Access service, which is provided by the plugged system itself or a service bus [20]. The services allow BI a universal access to all connected systems and all systems access to BI analyses [35]. The collected data is modified in a source conform structure by Transformation and Infrastructure services. Here, a source can be a service or an information system like the DWH. The DWH is one of many components and stores historical data. Although, the system wide data will be updated via a service bus by using publish-and-subscribe. If persistently storing is not required, the data can be promptly accessed by a service from any operational system on demand and can used for further analysis in order of a so called request-response model.
	Integration various data sources	4.14	0.00	4.29	0.00	
	Data Manipulation/Cleansing	3.43	0.04	4.57	0.00	
	Meta-Daten-Management	3.57	0.02	4.14	0.00	
	Mapping capabilities	3.57	0.02	4.14	0.00	
Operational BI	BI functions	4.29	0.00	3.71	0.00	The service bus achieves a fusion between the operational and analytical layer thus an immediate supply of data and responsiveness is acquired. Consequently, Analytical services are available for operational questions [35].
	Operational functions	4.14	0.00	3.43	0.10	
	Operational data	4.14	0.00	3.71	0.00	
Real Time BI	Event-based (CDC)	4.71	0.00	3.71	0.00	Data delivery in real time requires publish-and-subscribe. The data holding system generates an event in case of a data modification, which is forwarded by the publisher component through the bus into the event engine. The engine gets the event
	Real time data acquisition	3.86	0.02	3.29	0.08	
	Real time reports	3.43	0.10	3.14	0.17	

	Loads only changed records	4.71	0.00	3.61	0.01	and sends it to the rule engine. Hence, the rule engine receives the business rule or process from the business rule repository and sends the corresponding event through the orchestration engine which orchestrates the right data acquisition process. Thus, the orchestration engine obtains the service description by the service repository. The orchestration starts and the Data Access service activates a read operation in the source system. The executed operation is notified by the orchestration engine, which trigger the Transformation and following services till a Data Access service write the data into the DWH.
Active BI	Automatic tasks	3.29	0.09	3.71	0.00	The event engine allows the realization of active BI, which processes simple and complex (CEP) events from all layers [35]. This requires a centralized business rule repository and a rule engine [22]. As a result, the event engine informs decision maker in a semi or fully automated way for primarily routine tasks. In the first case, the information is forwarded to a user through an Analytical service. In the latter case the event engine can trigger a process or service, which realizes the final decision for example in an operational system. One variant could be the embedding of Analytical services in processes. Here the flow is orchestrated in such a way that an Operational service forms the end of the process and executes the analytical results.
	Closed-loop	4.29	0.00	4.86	0.00	
	Alert delivering	4.43	0.00	3.86	0.00	
	Complex event processing	3.57	0.02	3.23	0.17	
Embed- ded BI	Reuse of functions	3.29	0.26	3.57	0.05	The encapsulated BI functionalities are provided in processes or other applications and expands the range of users. In this context, BI becomes a service provider. Here, the embedding is not limited to Functional services. Software services can be invoked as well. Especially the open interface simplifies the integration. Here, the possibilities are ranging from simple reporting to data mining service that could be integrated into office applications, enterprise portals, and search masks. The services are provided by the orchestration engine. In addition, embedded BI can occur in processes, because operational processes can be controlled by analytical information.
	Portals	4.43	0.00	3.86	0.00	
	Flexible analytics	4.14	0.00	2.86	0.04	

BPM	View actual process status	4.14	0.00	3.86	0.00	To enable BPM, BI has to work as service user. Thus, BI can request timely information from business processes, which will feed the DWH for analytical purposes. The required process key figures are sourced from the monitoring/analysis engine. For analytical purposes the process data has to be historicized in the DWH to allow time series analysis or Balance Scorecard implementations. [1] Thus, a synchronization of the business process and DWH is achieved, which enables timely decision. Ultimately, the monitoring/analysis engine delivers special process conditions or circumstances to initiate further analysis or decisions by the event engine. [35] Here, the coordination of business objects and business process is achieved and BPM realized.
	Process can be defined or mapped	3.86	0.00	3.43	0.03	
	Business monitoring activity	4.57	0.00	3.43	0.14	
General	OLAP	4.86	0.00	4.86	0.00	Analytical service representing the encapsulated BI frontend-tools (OLAP, analytics, reports). They even contain graphical abilities. The administrator panel creates business rules and service descriptions, which are related to the service and business rule repository, delivers a service development interface, models processes, and defines human activities.
	Analytical capabilities	4.71	0.00	4.43	0.00	
	Visualization	4.58	0.00	4.71	0.00	
	Reports	4.71	0.02	4.71	0.00	
	Administration	3.57	0.00	4.86	0.00	

Source: Author's own.

6. Conclusion and implications

The aim of the paper was the initial step of a development of a SoBI reference architecture to gain a latency reduction between data acquisition and decision, to support the changing analytical and to allow an enterprise wide integration of BI. Real Time BI decreases acquisition latency and is achieved through publish-and-subscribe. Here, data changes are recognized by the event engine in time and forwarded into the DWH by orchestrated ETL services. BPM address acquisition too. Thus, process informations are delivered from the monitoring/analysis into the event engine. Active BI addresses the decision latency. Defined business events are detected through the cooperation between event and rule engine. Regarding to the event a predefined order of services will be executed to write an automatic decision into the executing system. Additionally, embedded and operational BI focus on the

analysis latency. Thus, analytical or operative services can be embedded into operational or analytical systems to allow managers a faster and objective decision. We verified our results through expert interviews. SoBI integrates BI into processes and external systems, supply real time, extends the range of user, reduces decision latency, and provides flexibility. No existing approach was identified that satisfies all requirements. The strength of the proposed solution represents the multiple fulfillments of all BI concepts and a detailed recommendation for action. Enterprises can apply the reference model as pattern regarding to their needs. In this context, the reference architecture is extensible as well as reducible. E. g. elements like Monitoring/Analysis Engine can be omitted, if BPM is not needed or Predictive Analytics [9] can be integrated as service if demanded. A next step is the realization of a prototype to gain valuable design insights for further applications.

Bibliography

1. Gluchowski P., Gabriel R., Dittmar C.: Management Support Systeme und Business Intelligence; Springer; Berlin 2008.
2. Hackathorn R.: Minimizing Action Distance; at: <http://www.tdan.com/view-articles/5132/>; last access 2012-12-11; 2003.
3. Dayal U., Wilkinson K., Simitsis A.: M. Castellanos; Business Processes Meet Operational Business Intelligence;IEEE Data Eng. Bull; 2009.
4. Golfarelli M., Rizzi S., Cella I.: Beyond Data Warehousing; DOLAP; USA; November; 1-6; 2004:
5. Davis J., Imhoff C., White C.: Operational Business Intelligence: The State-of-the-Art; at: <http://www.b-eye-network.com/files/Operational%20BI%20Research%20Report.pdf>; last access 2012-12-20; 2009.
6. Chaudhuri S., Dayal U.: An Overview of Data Warehousing and OLAP Technology; ACM SIGMOD Record; 26; 1; March;65-74; 1997.
7. Fettke P., Loos P.: Referenzmodellierungsforschung; Wirtschaftsinformatik; 46; 5; May; 331-340; 2004.
8. Chamoni P., Gluchowski: Integrationstrends bei Business-Intelligence-Systemen; Wirtschaftsinformatik; 46; 2;February; 119-128; 2004.
9. Schulze K.D.: Besbak U., Dinter B.: Business Intelligence-StudieSteriaMummert Consulting; Hamburg 2009.
10. Wilkes S., Harby J.: A move to drive industry standardization of SOA;at: <http://xml.coverpages.org/SOA-blueprintsConceptsV05.pdf>; last access 2012-12-04; 2004.
11. Martens B., Teuteberg F.: Risk and Compliance Management for Cloud Computing Services: Designing a Reference Model; AMCIS; USA; August; 1-10; 2011.

12. Frank U.: Evaluation of Reference Models; in Peter Fettke and Peter Loos (Eds.) Reference Modeling for Business Systems Analysis; IGI Publishing; 118-139; 2007.
13. Sacu C., Spruit M.: BIDM: The Business Intelligence Development Model; ICEIS; Portugal; 288-293; 2010.
14. Eckerson W.: TDWI Report Series Best Practices in Business Performance Management; at: http://www.olap.it/Articoli/2004_Best_Practices_Business_Report.pdf; last access 2012-12-12; 2004.
15. Eckerson W.: Pervasive business intelligence Techniques and Technologies to Deploy BI on an Enterprise Scale; at: http://www.strategycompanion.com/_Download/TDWI_BPR_PerBI_StrategyCompanion.pdf; last access 2012-12-12; 2008.
16. Russom P.: TDWI best practices Report operational datawarehousing; at: <http://tdwi.org/research/list/tdwi-best-practices-reports.aspx>; last access 2012-12-13; 2010.
17. Russom P.: TDWI Next Generation Data Integration; at: <http://tdwi.org/webcasts/2011/04/next-generation-data-integration.aspx>; last access 2012-12-11; 2011.
18. Chaudhari M., Dietrich W.: A Distributed Event Stream Processing Framework for Materialized Views over Heterogeneous Data Sources; VLDB; Singapore; September; 13-18; 2010.
19. Hackathorn R.: What is Active Business Intelligence?; at: <http://www.information-management.com/issues/20020501/5171-1.html>; last access 2012-12-12; 2001.
20. Dinter B.: Einsatzmöglichkeiten serviceorientierter Architekturen in der Informationslogistik; in Jochen Töpfer and Robert Winter (Eds.) Active Enterprise Intelligence; Berlin; Springer; 221-242; 2008.
21. Nijkamp E., Oberhofer M.: Embedded Analytics in Front Office Applications; in Johann Freytag (Eds.) LNI; 144; Springer; 449-459; 2009.
22. Martin W.: Performance Management and Analytics; at: <http://www.wolfgang-martin-team.net/BI-BPM-SOA.php>; last access 2012-12-11; 2012.
23. Buder J., Felden C.: Deducing demands at business-intelligence-systems beyond unbundling within the European energy markets; ECIS; Verona; June; 3305-3317; 2009.
24. TEC; Business Intelligence and Data Management; at: <http://bi.technologyevaluation.com/>; accessed 2012-12-17; 2012.
25. Cooper H.M.: Synthesizing Research: A Guide for Literature Reviews. Thousand Oaks, Sage Pubn.
26. Webster J., Watson R.T.: Analyzing the past to prepare for the future: writing a literature review; MIS Quarterly; 26; 2; 13-23; 2002.
27. Wu L., Barsh G., Bartolini C.: A Service-oriented Architecture for Business Intelligence; SOCA; 279-285; 2007.

28. Awad M., Abdullah M.S., Ali A.B.M.: Extending ETL framework using service oriented architecture; *Procedia Computer Science*; 3; 110-114; 2011.
29. Bucher T., Gericke A., Sigg S.: Process-centric business Intelligence; *BPM Journal*; 15; 3; 408-429; 2009.
30. Chan L.K., Sim Y.W., Yeoh W.: A SOA-Driven Business Intelligence Architecture; *IBIMA*; 1-7; 2011.
31. Dinter B., Stroh F.: Design Factors for Service-oriented Architecture Applied to Analytical Information Systems: an Explorative Analysis. ECIS, June 1-12, Verona 2009.
32. Gordon S., Grigg R., Horne M., Thurman S.: Service-Oriented Business Intelligence; *The Architecture Journal*; 6; 1; January; 23-32; 2006.
33. Nguyen T.M., Schiefer S., Tjoa A.M.: Sense & response service architecture (SARESA); *DOLAP*; New York; 77-86; 2005.
34. Panian Z.: How to Make Business Intelligence Actionable through Service-oriented Architecture; *WSEAS*; 5; 5; 210-221; 2008.
35. Vogt T., Neuhaus S., Linden M., Chamoni P.: Business-Intelligence-Konzept auf Basis einer Event-Driven Service-Oriented Architecture; in Barbara Dinter and Robert Winter (Eds.) *Synergiendurch Integration und Informationslogistik*; 217-229, 2008.
36. Müller R.M., Linders S., Pires L.F.: Business Intelligence and Service-oriented Architecture; *Information Systems Management*; 27; 2; August; 168-187; 2010.
37. Felden C., Buder J.: Integrated Information Supply for Decision Support in Grid Companies; *BISE* 4; 1; February; 15-29; 2012.
38. Flick U.: *An Introduction to qualitative Research*; London: SAGE; 2009.
39. Likert R.: A Technique for the Measurement of Attitudes; *Archives of Psychology*; 22; 140; 1-55; 1932.

Omówienie

Artykuł zajmuje się elastycznym podejściem zmniejszającym opóźnienia w dynamicznym środowisku rynkowym, występujące podczas implementacji i użytkowania inteligentnych systemów komputerowych ukierunkowanych na klienta. Systemy te są pomocne w procesie wspomagania decyzji do zmieniających się wymagań biznesowych. Problemy praktycznej realizacji systemów BI dotyczą integracji i wydajności system. W artykule zidentyfikowano i zdefiniowano możliwe do wystąpienia problem, stworzono model architektury referencyjnej, referencyjny model inteligentnych systemów komputerowych, ukierunkowanych na klienta z uwzględnieniem podziału na warstwy i zaproponowano procedure weryfikacji i walidacji.