Existing and Future Standards for Event-Driven Business Process Management

Rainer von Ammon¹, Christoph Emmersberger², Thomas Ertlmaier¹, Opher Etzion³, Thomas Paulus¹, Florian Springer²

¹ CITT GmbH, Konrad-Adenauerallee 30, D-93051 Regensburg, Germany
² Senacor Technologies AG, Vordere Cramergasse 11, D-90478 Nürnberg, Germany
³ IBM HRL, Haifa University Campus, IL-31905 Haifa

opher.etzion@il.ibm.com

ABSTRACT
The development and operation of modern IT infrastructures requires generally accepted standards. Many standardization efforts are currently ongoing within the service oriented and event processing community. Recently a new discipline entitled “Event-Driven Business Process Management (ED-BPM)” has emerged which takes a synergetic approach within this larger area. Consecutively, topics being discussed in this paper relate to the role of standards in the ED-BPM context, the need for a standard per se and the benefits and shortcomings of standardization in early phases vs. late phases. Within this activities, the most interesting proposal for a reference architecture is the Networked European Software and Services Initiative (NESSI) approach called NESSI Open Service Framework – Reference Architecture (NEXOF-RA) which has to be enhanced in order to support ED-BPM applications. Thereupon a proposal for describing the context and structure of occurring events on a descriptive and non-complex level of events is discussed. Related industry standards like “Notification Event Architecture for Retail (NEAR)” and the possibility to extend that approach to other domains are surveyed. Two possible extensions are exemplified, for the logistics domain (NEAL), and for the finance domain (NEAF). Use cases for the retail, logistics, and finance domains are demonstrated. In the final analysis conclusions are drawn and proposed action items for advancing the ED-BPM standardization are made.

Submission Type & Conference
Industry Paper for the DEBS 2009, 3rd ACM International Conference on Distributed Event-Based Systems, July 6-9, 2009 – Nashville, TN, USA

Keywords

1. INTRODUCTION TO STANDARDIZATION EFFORTS
The development and operation of modern IT infrastructures requires generally accepted standards. In large heterogeneous system landscapes, standards are indispensable for communication and data exchange. Without having networking standards and standardized protocols like TCP/IP or Ethernet and information presentation and structuring standards like GIF, HTML, CSS or XML, the internet would not have been the powerful global network as it has become these days. As described in [15], a standard is “a document, established by consensus and approved by a recognized body that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context”.

The benefit of standards can be characterized considering the following different types of involved adopters:

* A vendor has the possibility to meet the customers demand for interoperability.

* The developer leverages the know-how of other participants by using standardized frameworks and interfaces additionally with a choice of standardized tools and platforms. The implementation speed increases and only one general integration interface has to be supported.

* The end-user profits from reduced costs and risks while adopting a special, pre defined and tested technology and gets insight into the best practices of the industry.

However, it is not always the best solution for a young technology to develop a standard in early phases of its maturity. Having a standard on immature technology can become an
obstacle to the progress of an area. The use of de facto standards is a common way to reduce the risk of influencing the innovation process. In contrast to an early introduction, a late introduction of standards leads also to significant problems, since there already exists a variety of established best practices and proprietary vendor aligned solutions. Standardization should focus the industry on a coherent set of technologies. Without generating standards for new technologies, the innovation process can hardly be controlled which results in a rank growth of the technology. [12]

Many standardization efforts are currently ongoing within the service oriented and event processing community with little agreement on eventing [18] mostly dealing from a technical point of view e.g. the creation of a common event distribution standard for the transmission, transformation and translation of services and events based on existing standards like WS-Eventing and WS-Notification. Even though these efforts often result not only in an academic or industry wide standard, but more often in a rivalry between different standardization initiatives with the outcome of slightly abridged standards with different features on top of the same source, in this case WS-ResourceTransfer and WS-ResourceCatalog. [26]

 Nonetheless, the combination of service oriented and event oriented thoughts are drawn closely together by the emergence of event driven architectures, hence dealing with different issues and problem solving approaches. Within this movement the most interesting proposal for reference architecture is the Networked European Software & Services Initiative (NESSI) [23] approach called NESSI Open Service Framework – Reference Architecture (NEXOF-RA) [24] shown in Figure 1 [25] which has been enhanced with event processing capabilities that are marked in each level of the layered architecture. Event processing may be layered itself to display increasing abstractions of events from a low level (technical layer) to higher level (business process layer) as proposed by David Luckham’s event hierarchies [19].

![Figure 1. NEXOF-RA with event processing enhancements](image)

The following extensions on the existing NEXOF-RA have been proposed in aspects of integrating event processing therein:

Creating the Event Cloud

The first enhancement deals with the ability to produce events that are consumed by other services or by intermediaries that process the events. The "Event Cloud" is a collection of all events that may impact decisions and actions in the enterprise, this can come from instrumentation of services, connecting sensors to the service platform, and applying “pull” mechanism to legacy software, or cases in which instrumentation is not cost-effective.

Harnessing the Event Cloud

The decoupling principle means that services are sending their events and act as producers of the "Event Cloud" without any need to be aware of the event consumers. Harnessing the "Event Cloud" means managing the event formats using an event repository, and route the events to their processing destinations. Some events are routed directly to consumer services in their raw form, and some events are routed to event processing intermediaries, using “event channels”, using either the general pipes that route “event at a time”, or using the sub-case of "event streams" that route a “collection of events at a time” for a set-oriented processing. The routing can be: subscription based (pub/sub protocol), itinerary based (according to event content), intelligent routing (by decision process) or calendar-based routing and the routing service can even possess filtering capabilities to filter out irrelevant events.

Event Transformation

In many cases the events that arrive in the cloud are not in the form that is required by the consumer or the processing entity. The framework requires providing transformation services in the form of: format translation, creating aggregated events, split events, and enrichment of the event from various sources.

Pattern Detection (CEP- Complex Event Processing)

A pattern is a relationship among multiple events that has interpretation as a virtual event in the user's domain, e.g.: the fact that a customer complained three times about the same products, or that at least fifty customers have complained about a single product are patterns whose detection has some significance.

Situation Management

At the end of the event processing chain, there may be the detection of a situation. A situation is a derived event resulted from processing (filtering, transformation, pattern detection or combination of all) and consumed by some consumer; the situation management enables to tie situations with actions, and traces back the lineage of a situation for auditing purposes.

Actions – Events impact on the control flow

Actions are the projection of the situation detection of one or more services. Actions can be notifications to dashboards or individuals, but also can be more active, trigger a business process service, modifying control flows dynamically, or interrupting operating services.

Event processing modeling tools

Enhancements for the set of service oriented modeling tools is performed by adding modeling tools that are specific to capture the event processing part of the application. This should be integrated with other modeling tools; the modeling will also include specific industry models and use patterns.

Semantic Language (EPDL – Event Processing Description Language)
While there are many operational languages for processing events, a more formal, semantic language that will be able to define the end-to-end event processing logic and translate it automatically to execution languages will be added to the toolkit of the framework, in a consistent way to the modeling tools.

**Event-driven Context handling**

The need to use the notion of context for decisions about invocation of services, as well as for their internal decisions, has already been identified in the Reference Architecture, the event-driven enhancements will enable to open, close or modify contexts, based on detected situations, in automatic and dynamic way, and will enable real-time adaptation of the context-sensitive logic.

*The observation perspective (Business Activity Monitoring – BAM)*

Instrumentation of services enables observing the health of business processes by processing events that monitor the running services and feedback on that by means of key performance indicators, and exception detections.

The amount, structure and interaction of events that are emitted inside this framework are much diversified. This leads to the common denominator in the form of similarities and diversities within and beyond business domains like automotive, aviation, logistic, entertainment, finance, banking, insurance or retail. The events, the patterns that can be derived, and the business processes in which they occur should be considered as a coherent part of the reference model, specific to every domain whereby issuing domain oriented Patterns (DoReMoPat) [4].

The modeling notation in the following use cases for the different domains does neither serve as a domain-specific language (DSL) being tailored to a specific application domain [21] nor inherits the capability of presenting the interaction, composition or aggregation of events. Rather, it is intended for describing the context and structure of occurring events on a descriptive and non-complex level hence it is composed by rudimentary elements derived from the Unified Modeling Language (UML) [30], the Models to Text Transformation Language (MOF) [29], and the XML Metadata Interchange (XMI) [28]. Additionally the notation is based on the IXRetail Notification for Event Architecture for Retail (NEAR) standard [22] which is discussed in detail in the next chapter.

---

**Figure 2. Exemplified modeling notation for the use cases**

The modeling notation exemplified in Figure 2 contains the following elements:

1. The name of a data element is embedded in a rectangle whereas the color black is used by default.
2. Blue colored elements are determining common data elements that can occur multiple times within the domain model.
3. Lines with white arrowheads represent a generalization.
4. Content elements are indicated with a prefixed hyphen followed by their name and their possibility of occurrence in brackets in the data element.
5. The at-sign characterizes an attribute of the data element whose enumeration is decoupled from the rectangle and positioned beside it.
6. The plus sign indicates a composite element within a data element.
7. The keyword <choice> in greater-than and less-than signs determines the selection possibility of exactly one element from the group which can vary from instance to instance and ends with </choice>.
8. <option> indicates the grouping of elements in different configurations ending with a slash in front of the options name.
9. Lines with white diamond heads represent a generalization of choice in which exactly one choice per instance can occur.

**2. THE RETAIL DOMAIN AND THE NEAR STANDARD**

Figure 3 shows an example of a typical, distributed, heterogeneous IT infrastructure which can be found inside a retail enterprise.
Figure 3. Typical organizational structure in retail industries

The head office is located on top of the typical organizational structure with the central IT backbone and the data center. All branches of the retail company are connected via the network to one large and complex enterprise bus. Inside the branches there is a wide variety of different devices and software systems from different vendors. To ensure proper communication between all different components, common communication standards are mandatory. A successful standard for processing events in the retail domain from the different devices and software systems is the IXRetail NEAR standard. This standard was designed and released by the Association for Retail Technology Standards (ARTS) of the National Retail Federation. The purpose of this standard is the reduction of time and cost for integrating retail applications by using asynchronous notification events. Therefore the standard defines a framework for XML formatted asynchronous notification event messages and the publish/subscribe methodology. It also defines a reference-architecture to support the notification event framework and describes guidelines and best practices for future development and conformance testing.

From the business point of view the standard aims to simplify the real-time integration of loosely coupled applications and to lower the integration costs thus increase the customer service and operational efficiencies. A typical profile for participating applications in a NEAR based retail infrastructure is characterized as:

- Real time producing applications
- Real time consuming applications
- Message broker systems to handle subscribe / unsubscribe requests for different event types
- No need for common architectures, data models or platforms
- Vendor and source independent producing and consuming applications

The implementation of the reference architecture leads to an asynchronous, non persistent XML notification event architecture which removes the requirement for a direct point-to-point integration [6].

3. EVENT MODEL OF THE NEAR STANDARD

This chapter describes an exemplary domain model of the NEAR standard and shows possible extensions for this retail specific model in terms of using them in a multi domain environment.

Figure 4 shows the Customer Event Domain Model. This model was designed to generate a common data schema for dealing with different customer events. As described in the corresponding charter of the IXRetail NEAR standard, the development of the data model was done in two phases: the first phase of the development was focused on working just with simple customer information, e.g. adding, updating, deleting or identifying a customer. The second phase of the development tries to combine different marketing campaigns with the customer data like customer-specific coupons, loyalty points or promotional messages. The basic concept behind this data model is the possibility to exchange customer data between different software systems inside the retail company to create specialized customer-oriented applications.[5]

The Customer Event Domain Model describes a common data schema for dealing with customer activities. The attribute CustomerEvent holds the version of the model. This is a common element which can be found in all derived data models. The CustomerEventType describes the detailed event type. In this diagram two types of customer events are listed: CustomerIdentifiedEvent and CustomerAddedEvent. The EventCommonData are holding different technical information for processing this event in an appropriate context. The most interesting section in this model is the CustomerCommonData. This field shows the different attributes which have been defined by the IXRetail working group. It consists of rudimentary data which are necessary for processing customer data in the described retail domain.

Figure 4. Customer Event Domain Model [7]

Future applications will not only act inside the enterprise, they will also interact beyond enterprise boundaries. Therefore the business information of the applications should be reusable in different domains. A simple case for such an application can be the event driven intercommunication between online retail stores and logistic companies. The customer information will be
recorded and processed in the online store and have to be reused by the logistics company. For monitoring the payment status of an order process, the exchange of standardized data between the finance and the retail domain will also be necessary.

The shown data model from the NEAR standard is an adequate approach for processing customer events inside a single enterprise. But for cross-domain applications this model has to be extended to become a more common solution. As described later in paragraph 4.2.1, a common data structure for postal addresses is available in the ISO/IEC standards. Similar to the approach of defining a common address structure, a common customer data structure has to be designed which can be used in different domains.

All the discussed extensions of the Customer Event Domain Model can respectively be adapted to further models of the NEAR standard like the Item Event Domain Model or the POSLog Event Domain Model.

4. NEAL: A LOGISTICS EVENTMODEL

The general purpose of the Notification Event Architecture for Logistics (NEAL) is to support enterprises within the logistics domain in creating event driven systems for aligning their business and IT. The idea is to provide a basic set of events to assist in designing and implementing ED-BPM applications.

The first subsection provides an introduction into the logistics domain that points out the companies’ different levels of the logistics knowledge, as it can be found in the literature, touches the basic process knowledge of a common package deliverer, to establish a basic understanding for the use case scenarios to which the event architecture can be applied. After this knowledge is imposed, the next section presents an extract of the current NEAL before the last section tries to point out the benefits as well as the future work which has to be accomplished.

4.1 Introduction to the Logistics Domain

Taking a look back, the basic concepts of logistics can be found in the second half of the 20th century. Since that time, there is still no completed definition for the term “logistics”. Within the scientific discussion, the common understanding of logistics extends the simple transporting, handling, and warehousing operations to broad management functions [9]. “Even when taking all the different opinions into account, a consensus exists (…) that the central function of logistics is the bridging of space-time disparities concerning goods and materials.” [9]

4.1.1 Logistics Process Model

In order to introduce the NEAL extension it is necessary to point out, that the field of logistics has several different characteristics (e.g. package delivery, transportation, production logistics, etc.). Each characteristic has its own processes fitting to their specific business needs. The following use case concentrates on the field of package delivery. To get an impression of the process model in the package delivery domain, Figure 5 illustrates a common logistics process model [31].

Figure 5. Common logistics process model derived from [31] and [14]

The first layer in Figure 5 shows the interaction layer between the package deliverer and the customer. Customers can place orders, e.g. the pickup and delivery of a certain amount of to different destinations. At the second layer commercial processes handle all customer orders from a business view. This means that the actual order administration as well as the billing and invoicing needs to be controlled by these processes. Besides the order and billing tasks, the commercial layer is also responsible for exception handling regarding customer claims as well as production errors (e.g. offenses against service level agreements (SLA) like late package delivery and damaged or lost packages).

The third layer is the actual transportation layer handling transport orders. The use case that demonstrates some capabilities of the event architecture, concentrates on the physical service provision J. Huber divides the physical service provision layer that contains outbound and inbound processing as well as the transportation and execution into three lines, where each line actually executes different tasks [14].

Figure 6. Detailed view of physical service provision layer derived from [14]

The following enumeration points out the characteristics of the physical service provision layer:

- **Supply Line**: Due to the fact that customers can be localized everywhere, the supply line needs to be very widespread. Many small supply points collect the packages from the customers that have ordered a transportation service. Depending on the package deliverer, the supply points can vary from receiving stores, where customers drop a large amount of transportation goods directly on a conveyor belt to agencies, where smaller amounts of packages can be passed on to agency personnel to automates, where customers can drop single packages step by step without any personnel. Besides the business model where the customer takes the
goods to the supply point, there is a second variant of supply point, where the package deliverer picks the transportation goods from the customer. The pickup places can be seen as supply points, because at that point the transfer of perils takes place.

- **Main Line**: The main line incorporates the actual transportation process and is characterized by the transportation of large amounts of packages as bulks loaded into trucks. In this physical service provision section, the packages that have been sorted in the outgoing depots are transported to the incoming depots via several hubs. Several algorithms that are explained in [20] are common in optimizing the transportation processes according to the specific business model. The dependencies on the business models are the amount of time the transportation may take from the transfer of perils until it reaches the package receiver.

- **Delivery Line**: During the delivery line phase, the bulks are sorted in the incoming depots to distribute the single packages in the widespread area. Because the distribution takes also place in the same areas as the supply line, both activities are parallel. This means, that the deliverer distributes and collects packages at the same time, in some cases even at the same customer.

During the whole physical service provision, it is obvious, that plenty of the physical IT systems need to interact in order to plan, control and track the delivery of packages. Looking at the main line distribution systems (Figure 6), it becomes clear that the applications are highly distributed, e.g. handheld scanners used at the supply and delivery points, routing systems planning the actual tours for the trucks, navigation systems being deployed in each transporting vehicle, sorting system in each single hub and depot, just to list some of them.

As a consequence, a huge amount of events is generated by the distributed IT applications in a “(…) partially ordered (…)” way “(…) either bounded or unbounded, where the partial orderings are imposed by the casual, timing and other relationships between the events.”[20] Taking a closer look into the event cloud of the distributed network, it is possible to identify single event streams, as a “(…) linearly ordered sequence of events (…)” [20] which correlate with the partially ordered events in the event cloud.

As the Event Processing Technical Society (EPTS) explains in one of their notes on the event cloud definition, an event cloud may contain not only the already explained event streams but also many event channels and event types.[20] Supporting the architects and developers in the logistics domain, a central data model for the events can help to gain control over the multiplicity of event types within the domain and open up the opportunity to interconnect with other domains.

### 4.2 Proposal: Logistics Event Architecture

The creation of reference models requires both knowledge of the business processes and understanding of the events and their role. After this knowledge is established, complex events can be derived from the basis events to influence the processes in an automated, possibly preemptive way.

The basic idea of having a knowledge base with the data being transferred as events is not new. Taking a look at the SOA trends, there is a strong inclination to centralize all information about the data transferred via interfaces within service repositories or service registries. Service repositories differ from service registries. N. M. Josuttis describes the difference in the following way: Service registries “(…) manage services from a technical point of view (…)”, repositories “(…) from a business point of view (…)” [17]. Technical issues are e.g. signatures, deployment information etc., business issues are interfaces, contracts, SLA’s, dependencies etc. In a SOA, there exists a place containing all information about service interfaces as well as the data which is transferred via the interfaces called service repository. But building these repositories is complex and takes high effort within a company. This leads to the point where a domain specific Notification Event Architecture, a basic set of basis events, can help creating the repositories faster, in a more generalized manner. As a consequence of the generalized repositories, complex events can be derived in a more standardized way and the quality of their utilization should increase.

The following paragraph introduces three basis events of the Notification Event Architecture for Logistics (NEAL). It is the first proposal of this domain specific architecture and can not be seen at the same development level as the current de facto standard in the retail domain (NEAR) for event driven architectures [13].

#### 4.2.1 Common Address Event Architecture

As already mentioned in the introduction, one of the logistics disciplines is transportation. Transportation has as basic task to move goods from a source to a sink. In the world of package deliverers, sources are generally the customers, sinks the recipients. Both, customer and recipient can be localized by their postal address. The postal address identifies both ends worldwide unique, to enable the transportation from any customer to any recipient.

Taking a closer look at the structure of a postal address, it becomes clear, that each region has its own structure of identifying the actual local position of an address. An example for these differences is the structure of a Chinese postal address Structure compared to a German one.

#### Table 1. Chinese postal address example

<table>
<thead>
<tr>
<th>Example</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.R., China 528400</td>
<td>Country, Postal Code</td>
</tr>
<tr>
<td>Guangdong Province, Zhongshan City, East District, Hengda Garden, 7th Building, Room 702</td>
<td>Province, City, District, Building Name, House Number</td>
</tr>
<tr>
<td>To: Mr Xiaoming Zhang</td>
<td>Addressee</td>
</tr>
</tbody>
</table>
Comparing both structures (Table 1 and Table 2) it becomes very clear that each region has its own structure with common and local attributes. The issue of different local standards (e.g. DIN 3008 (DIN: German Institute for Standardization), definition for German address format) is addressed by the ISO/IEC FCD 19773: Information technology - Metadata registries (MDR) Modules.

Table 2. German postal address example

<table>
<thead>
<tr>
<th>Example</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firma ABC</td>
<td>Company</td>
</tr>
<tr>
<td>Kundendienst</td>
<td>Department</td>
</tr>
<tr>
<td>Frau Monika Mustermann</td>
<td>Name (generally preceded by Frau=Ms or Herr=Mr)</td>
</tr>
<tr>
<td>Ostvorstadt</td>
<td>Neighbourhood</td>
</tr>
<tr>
<td>Hauptstraße 5</td>
<td>Streetname + number</td>
</tr>
<tr>
<td>01234 Musterstadt</td>
<td>Postal code + town</td>
</tr>
<tr>
<td></td>
<td>Country (if other than Germany)</td>
</tr>
</tbody>
</table>

The ISO/IEC 19773 is written by the Technical Committee ISO/IEC JTC1, Information Technology, Subcommittee SC32, Data Management and Interchange, and specifies basically “small modules that can be used or reused in applications” [16]. The purpose of the NEAL is the use and reuse of applications. Besides this congruent goal there are several benefits in integrating an existing standard into the event architecture. It increases the chance of the user acceptance because the interoperability with existing applications. Another benefit is that the time consuming process of coordination has already taken place. Otherwise it would take again a lot of time to talk to all different parties for making a final agreement on the data structure.

Figure 7 shows an extract of the ISO/IEC 19773 postal address data structure in the notation of notification event architectures. For modeling event architectures of different domains, the idea is to take different common events. The Common Postal Address event is one of the candidates to become a common event over all domains. The reason for this is that address data is used in every domain and a common standard would increase the chance of an unproblematic data exchange. The only restriction on the postal address model is the flexibility that only needed attributes are used in the specific domain and the architecture does not force all domains to implement all attributes.

4.2.2 Order Event Architecture

Recalling the logistics process model of paragraph 4.1.1 it becomes obvious that the starting activity that entails all actions is an order placed by a customer.

Analyzing an order of a general package deliverer, an order (Figure 8) consists of a shipment and on top of it an additional, optional service. Services are not the core business of a package deliverer. As a consequence these services are executed by partners. The core transportation business is executed within the physical service provision line. To enable the execution of the physical process, it is necessary to provide the postal address information of the delivery point as well as the address of the recipient.

Each order is placed by a customer whereby each customer is identified uniquely by a postal address. The whole event structure is wrapped by the event common data that provides some internal system information for further processing as well as organizational information.

The physical service provision needs to keep the commercial department informed about the shipment status for keeping track about the actual order status.
value of the package transportation. They are used in the commercial service provision to dispose the package streams with a maximum rate of return. The tracking events also provide information that is labeled on the package like the destination as a postal address.

Besides the functional information, the event type is also wrapped with technical information.

### 4.3 Conclusions about NEAL

Pointing out the benefits of NEAL, it is best to focus first on the interaction between the commercial and the physical service provision. The commercial service provision knows in almost real time, what the current state of the physical service provision is and can react on that. Reactions on e.g. a delay during a package delivery can entail customer interactions as well as new disposes at service partners.

### 5. NEAF – A POTENTIAL FINANCIAL EVENTMODEL

The purpose of the Notification Event Architecture for Finance (NEAF) is to support enterprises within the finance domain in introducing event driven systems. The idea is to provide basic concepts in designing, implementing and managing event based systems.

The first section provides a short introduction in finance with the focus on banking and possible applicable areas for an event based approach. After that the first event model of a typical account transaction and the benefits of the standardized model are explained.
5.1 Introduction to the Finance Domain

The use of Complex Event Processing (CEP) within business processes promises, especially in the financial industry a great potential [32]. “Event-Driven Business Process Management (ED-BPM) is a combination of actually two different disciplines: Business Process Management (BPM) and CEP (…)” [2]. An example for an applicable area of ED-BPM is the financial sector. One of the major applications of the financial sector is managing credits, both credits provide and credits granted to a financial institution as illustrated in Figure 11. A bank is expected to introduce an adequate risk management to avoid situations emerging in the financial crisis. This requires on the one hand regulations and strategies providing policies for banks, and on the other hand technologies helping to get insight into complex processes. ED-BPM has the potential and technologies to provide a better insight into business processes and to allow reactions to potential exceptions before they actually occur [10]. Another use case with this approach is the detection of fraud [34]. For monitoring business processes and for detecting and reacting on problems, errors or fraud in real time, CEP uses special mechanisms for pattern recognition. A hurdle for introduction of ED-BPM in finance and also other domains is that currently no standards, guidelines, or practical experience exist. This is necessary to allow a smoothly and quick implementation of ED-BPM. The Notification Event Architecture for Finance (NEAF) aims to provide a set of reusable, standardized basic events and patterns tailored for finance business processes.

5.1.1 Application Area of NEAF

The financial sector deals, among other business scenarios, with managing funds of the private and public sector. This section has its focus on a typical banking use case. Figure 12 shows a common value chain of a bank [31]. In each of the areas of the value chain it is conceivable to use ED-BPM to support business processes. One task of future research will be to analyse business processes and find out in which areas ED-BPM achieves the most valuable improvement. Another task is to define a collection of common events for specific use cases in finance. Therefore a standardized way of describing various events is necessary.

5.1.2 Foundation of NEAF

NEAF is based on the NEAR standard of the Association for Retail Technology Standard (ARTS) introduced in chapter 3. The concepts and symbols for describing these events are largely reused. That allows a uniform description of events across various domains. Thereby commonalities and differences between various domains, e.g. finance, logistics and retail can be identified.

5.2 Transactions as an Example for NEAF

The first example of NEAF describes an account transaction or posting. For each money transfer, direct debit, cash withdrawal, car payments and many more, a transaction on the account of a customer has to be processed. The Postbank for example, a German institute with focus on the private sector, has to deal with more than 10.000.000 transactions per day [33]. Hence account transactions are the largest proportion of events in the banking sector. These transactions are primarily part of the area “Operations and Execution” (Figure 12). The amount and the proportion of specific transactions are depending on the orientation of the bank. A bank specialized on checking accounts for private customers, has the biggest part of transactions assigned to business processes of “Cash & Liquidity Management”. A bank specialized on trading has the largest proportion in the area of “Security, Funds and Derivates”. These areas provide very different business processes.

Figure 13 shows examples for actions which are possible triggers for a transaction and a “TransactionAccountEvent” from the perspective of a banking institute. The actions, responsible for a transaction, are part of different business processes with almost no commonality. For example, the interest for a deposit is based on a totally different business process than payment via credit card. Further insights in specific processes are not relevant for understanding the concepts of NEAF and therefore not further explained in this paper. The only significant commonality is that a transaction on the account needs to be executed. This transaction represents either a debit posting or a credit posting.
5.2.1 An Event Pattern for Transactions of Various Business Processes

The aim was to model a common and standardized event for an account transaction, which could be used in various business processes and for various banking institutes. Therefore transactions of specific business processes were analyzed concerning their commonalities and their embedded characteristic. This is necessary, because each transaction has specific attributes, e.g. a credit card payment needs other attributes than a money transfer to another account. These specific characteristics were analyzed and evaluated in respect to their relevance. Not relevant details were omitted and mandatory special cases were generalized and added. Result of this task was the common “AccountTransactionEvent” model which can be used in various business processes in the financial domain.

5.2.2 AccountTransactionEvent - Excerpt of NEAF

Figure 14 describes the „AccountTransactionEvent“, which represents a common account transaction, as a first example of NEAF. The model is based on the notation introduced in chapter 1. It consists on common data which are also necessary for other event models of NEAF. These data are highlighted blue and contain the keyword “Common”. E.g. bank account, credit or debit card and customer data are also used in other use cases and event models. The data for the customer address is based on the ISO standard explained in paragraph 4.2.1. The “EventCommonData” consists of primarily technical attributes, necessary for processing the event in a CEP engine.

The “AccountTransactionType” is a template data model proposal for any kind of bank transactions. It is extended with common data of financial domain and specific data of business processes. The “TransactionID” is the unique identifier for every instance of a transaction. “TransactionType” describes the type, e.g. credit payment, withdrawal, etc. whereas “Name” describes the initiator of the transaction. Depending on the “TransactionType” several text fields provide additional information about the transaction. The value of the transaction is represented by the attribute “Amount”. This field contains always a positive, two digits floating-point value. If the transaction marks a “payment” or a “deposit”, is defined by the attribute “PostingType”. “PostingDate” describes the date of incoming and “ValutaDate” the date of execution of the transaction. The transaction is always assigned to a customer represented by “Customer”. The “AddressCommonData” represents the customers address. “AccountSource” and “AccountDestination” describe the source and destination account of the account movement.
5.3 NEAF Conclusions
The „AccountTransactionEvent“ is a first attempt in describing basic events for ED-BPM in the financial domain. In the future NEAF will provide a collection of templates and models helping to improve the introduction and management of event based systems for finance to improve design and implementation. This approach is already achieved in the retail domain with NEAR. For each specific use case, the event templates (e.g. “AccountTransactionEvent“) can be enhanced with specific attributes. The proposals should also facilitate the exchange and processing of events by IT-systems between companies, e.g. in outsourcing projects, what becomes more and more necessary [33].

6. CONCLUSION AND FURTHER WORK
In this paper the state and the needed enhancements of the NEXOF-RA as a basis of exploitation for the ED-BPM oriented use cases from different domains was discussed. Furthermore the question of early and late standardization approaches with the corresponding impact on the technological evolution was discussed. We discussed NEAR as an existing standard for the retail domain and the appropriate value from the development and adopters point of view. The idea of defining basic event models in an as for now rudimentary and simplified notation as proposals for future standards was applied to other domains. From these domain-specific models of the basic events the event patterns and event processing logic shall be largely automatically generated for the catalogs of the domain-specific use cases, depending on the EPL approach. This EPL-logic can become quite complex, as it was shown in [8]. Approximately 400 lines of code for a SQL-like EPL approach in the case of the loss pattern of a bank application were needed to implement the prediction of impending losses from cancelled loan applications. This context is based on a predictive business with proactively detected situations and prevented by appropriate actions and measures which shall already be pre-modeled in the event patterns of use cases. [8]

The impact of the mentioned project ED-BPM/DoReMoPat is to show that the future internet of services on the basis of the ED-BPM-reference model [3] and the enhanced NEXOF-RA allows to build applications much faster than today and allow to change respectively enhance and manage applications in a very flexible way. The project exploits SSAl (service and software architectures and infrastructures) in order to verify SSAl and the impacts mentioned above, to build working solutions and products for the different domains as “breakthrough applications”. The number of different domains is important for the exploitation and verification of the generic applicability of the concepts and architectures. Different domains are also needed to derive a meta-model as an abstraction of the domain specific models.

The project will demonstrate that applications based on ED-BPM are additionally predicated on Grid- or Cloud Computing:

- ED-BPM is based on an enhancement of NEXOF-RA which shall be the basis of applications for Grid- and Cloud Computing.
- ED-BPM will be the basis for future Internet services, because the real-time aspects and the
flexibility will mainly be based on the new role of Event Processing.

- ED-BPM is an enhancement of the “classical” BPM and will be the basis for dynamically reacting on event patterns coming from a global event cloud of external or native events.
- ED-BPM must therefore enhance standards like BPMN [27] for integrating complex events in the modelling process at design time.
- ED-BPM must enhance BPEL [1] for making such processes executable at runtime.
- ED-BPM must define and create new domain specific standards like the existing NEAR (Notification Event Architecture for Retail), we have to define NEAL (logistics), NEAF (finance), etc.
- ED-BPM, especially the part of CEP, is the basis for real-time processing of mass data in event clouds.
- ED-BPM and the different approaches of Event Processing Languages and their standardization depending of their applicability in the different domains are sophisticated problems which have to be investigated. The appropriate EPL respectively the definition and flexible and easily changeable implementation of event processing logic will be the basis of triggering and changing the control flow of business processes.

7. REFERENCES


