# THE ADIWA PROJECT

## *On the Way to Just-in-Time Process Dynamics based on Events from the Internet of Things*

Markus Schief, Christian Kuhn, Birgit Zimmermann, Philipp Rösch
*SAP Research, Bleichstraße 8, Darmstadt, Germany*

Walter Waterfeld, Jens Schimmelpfennig, Dirk Mayer
*Software AG, Darmstadt, Germany*

Heiko Maus
*German Research Center for AI (DFKI GmbH), Kaiserslautern, Germany*

Jörn Eichler
*Fraunhofer SIT, Darmstadt, Germany*



Keywords: Event-based system, Event processing, Context awareness, Dynamic business processes, Internet of things.

Abstract: In this paper, we introduce a concept, which focuses on innovative commercial system implementations reflecting process-embedded events from the Internet of Things. The developed concepts are derived from experiences applying recent research advances to industry scenarios. The rationale behind the overall concept is twofold: while transparency is increased by event-based methodologies in the context of the Internet of Things, the agility of business processes is fostered by enhanced business process models, orchestration support, execution control, and user assistance.

# 1 INTRODUCTION

In recent years, the speed of change in business environments has been accelerating. To enable agile business processes, transparency is a crucial prerequisite. While a great deal of research has been performed in the field of dynamic business processes, a lack of acceptance can be constituted as, up to now, the data basis has not been sufficient (Pesic and van der Aalst, 2006). Our concept enables dynamic design, planning, and execution of business processes based on the identification, processing, and usage of real world events from the *Internet of Things* (IoT).

By analyzing four industry scenarios in the area of logistics, business services, retail, and production, we discovered requirements for our technical components and the concept architecture. With regards to transparency, the challenge is to identify the relevant real world events and transform them into consumable information pieces for business processes, e.g. (Heusler et al., 2006). A systematic and consistent concept is required to transform the IoT information and, hence, to increase the information value within business processes. In terms of business processes, information can be leveraged within the generic design of business processes as well as during the planning and execution of specific business process instances. Internal triggers, such as business process deviations, as well as external triggers, such as changed environmental factors, need to be covered by the targeted event-based information system.

To alleviate these challenges, an extension of business process modeling and execution is needed, which integrates IoT data as well as the description of process variance. Orchestration tooling as well as knowledge worker support have to be provided in order to

Schief M., Kuhn C., Zimmermann B., Rösch P., Waterfeld W., Schimmelpfennig J., Mayer D., Maus H. and Eichler J..
THE ADIWA PROJECT - On the Way to Just-in-Time Process Dynamics based on Events from the Internet of Things.
DOI: 10.5220/0003417503710377
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 371-377
ISBN: 978-989-8425-56-0

realize the full potential of IoT information. Supporting users is important for two main aspects: To ensure that humans are involved in business processes that are closer to the real world and, secondly, in order to foster process dynamics, process participants' know-how from the actual process execution needs to be exploited. The second aspect ensures a steady process evolution and adaptation to the real world.

# 2 FUNCTIONALITIES

## 2.1 Event Processing

Event processing plays a central role in the overall ADiWa approach of connecting business processes to the real world via IoT technologies. It is essential to complement the traditional top-down realization of business processes with a bottom-up approach of events. Only in this way can business processes cope with the unpredictability of the real world.

The connecting of business processes with IoT technologies results in a wide spread of the technology stack in ADiWa. It reaches from business process management, via service and event management, to IoT sensors and devices. Events occurring at all layers are often intertwined. Thus, there is a strong need to process and aggregate low-level IoT events to high level business events (Schmidt and Schief, 2010). This challenge usually requires not only a relatively complex multi-step processing of events into business events but also complex structured events during and as the result of the processing. Thus, there is a strong need for a complex event processing (CEP) component. In our event concept, we distinguish between two main event processing components: Firstly, the CEP component generates and processes complex structured events and, secondly, the event bus provides a scalable distribution via publish-subscribe-mechanisms.

Our work leverages state of the art research in these areas. The field of complex event processing has been initiated by (Luckham, 2002). (Hinze et al., 2009) gives an overview of the currently-developed technologies and identifies characteristics of a large number of applications. It shows that there exist two approaches for defining event processing language (EPL) rules: streaming queries, e.g., according to the SQL-style, and rule-based approaches. (von Ammon et al., 2008) describes an integration of event processing with Business Process Management (BPM) and gives examples of practical applications. Although it describes the first approach for the integration of event processing with BPM, it does not address the integration with IoT technologies. There is also no approach for an integrated event tooling for the business user from BPM to the IoT. Based on the Java Messaging Service (Hapner et al., 2002), a lot of work has been done on message bus functionality for the distributing and brokering of events. (Muhl et al., 2006) gives an overview of many relevant systems. Though quite a large number of techniques exists in different systems, they focus on the level of a messaging systems and not completely on events.

Beyond the general complex event processing, we additionally consider several quality criteria as well as analysis capabilities to further enhance the value of information provided by the complex events. To understand and trust the output of the system, the user requires reliability and guarantees. In ADiWa, we achieve these goals by a holistic quality determination and propagation approach. We consider four different layers associated with quality. At the lowest layer, the event producers emit events with a certain **production quality**. Typical event producers are sensors or devices (e.g., RFID readers); corresponding quality dimensions are, for example, accuracy and completeness of the measured values. Next, the events are transmitted and processed with a certain **notification** and **processing quality**, respectively. We determine the notification quality based on a large set of features of the event bus (Hinze et al., 2009). Examples for such features and their impact on the quality are: does the event bus support validity intervals of events, and what is the granularity? Is the event bus capable of early filtering, and how efficient and accurate is this filtering? Is there support for distributed processing? Is there support for privacy? The processing quality mainly targets the properties of the CEP engine like the above-mentioned throughput, latency, or accuracy of complex event detection. Finally, the events affect dynamic business processes with a certain **controlling quality**. Here, key performance indicators (KPIs) given by the business process management indicate the success of the dynamic adaptation of business processes utilizing events.

To realize the holistic quality determination and propagation, we make use of an **extended metadata schema**. There, we track the individual quality criteria as well as lineage information to additionally provide an uninterrupted and persistent trail of events for downstream analyses. For efficiency reasons, only the event-specific metadata are directly attached to the events (such as its age), while the remaining information is stored in external repositories (quality criteria of the event producers and event bus). Further, by benchmarking and monitoring, we evaluate, observe, and ensure the quality of the event transport.

With several analysis capabilities, we provide both the event processing and the user with valuable supplementary functionalities. These functions include the handling of missing events as well as the detection of new complex events. Further, we enable the proactive detection of occurring complex events; here, a user is notified with an alert-event if the occurrence probability of a specific complex event exceeds a given threshold. This alert allows the user to dynamically initiate appropriate actions at an early stage.

## 2.2 Process Modeling and Execution

Usually, only a low degree of flexibility is supported in business process management systems. There are a few approaches dealing with the question of how to enhance the flexibility of business process execution (Dadam and Reichert, 2009; Hallerbach et al., 2009). In addition to offering change operations, rules can be used to offer a more flexible business process execution. An overview of the current state of the art can be found in (Paschke and Kozlenkov, 2009) and (Graessle et al., 2006). Based on the enhanced transparency provided by IoT events, ADiWa enables companies to support flexible adaptation and dynamic execution of business processes. Four phases in the lifecycle of a process are considered in our concept: design, planning, execution and controlling.

The lifecycle of business processes starts with the **design phase**. In this phase, business processes are documented as semi-formal process models. In ADiWa, we design business processes using an extended event-driven process chain (Keller et al., 1992) method, allowing the definition of complex events and providing a means of integrating real-world IoT events into business processes. This approach allows process owners to define process-relevant events and to specify appropriate process responses. The resulting process models are transformed into technical representations and are used as templates for both the planning of individual process instances and the configuration of a process-controlling component.

In the **planning phase**, results of the modeling phase are instantiated (respectively adjusted) for specific process instances. Planning is an iterative process in which a large amount of intermediate results is created and interpreted. For example, all data occurring for one process instance and the matching rules have to be taken into account. In addition, it must be possible to adjust plans at a later time, for example, to adapt them to a suddenly-occurring event. Planers must be able to specify the required operations. They expect immediate reactions in order to be able to continue with planning directly after the adaptation. To enable this rapid interaction, the planning component must be able to read, analyze, and write data as quickly as possible. The goal is to enable users to plan processes flexibly. Thus, it must be possible to define several alternative planning sequences based on potential variations of (strategic and operative) objectives. Likewise, required activities and resources required need to be adjusted to reflect IoT events.

In the **execution phase**, the focus is on a flexible process execution in terms of IoT intelligence. To be able to offer this functionality, an execution environment is needed that is based on the results of both the design and the planning phase. In addition, the execution environment must provide a rule-based process execution as well as the interpretation of events from the IoT. All this information must be combined in order to respond to events showing up during runtime. As such, during runtime, the execution of a process instance can be adapted to its actual context. If an upcoming event necessitates a change of planning or even execution of certain process steps, the execution environment supports this flexibility, for example, by offering change operations similar to the ones defined by (Hallerbach et al., 2009). Adaptations in process execution or process planning are communicated as process-relevant events. According to the event context, three kinds of business process-relevant events can be distinguished: execution-, planning-, and design-relevant events. If the execution or planning of a certain process needs to be changed frequently during runtime, this fact may indicate that the original process design, on which planning and execution is based upon, is insufficient or outdated. Thus, the design needs to be adjusted to ideally fit to the runtime requirements. Such optimization potential is detected by a CEP engine (Section 2.1) reacting to both execution- and planning-relevant events, which are aggregated to the above-described design-relevant events. Design-relevant events trigger governance processes, which support the process owner in the optimization of business process models, providing an automated change management approach for execution-driven process evolution.

Another key stage for flexible processes is the **controlling phase**. The results of monitoring and controlling affect all process lifecycle phases. The ADiWa monitoring and controlling component collects event data from the execution platform as well as identified business process-relevant events from the CEP engine. It reassembles single as-is process instances and visualizes them as event-driven process chains. By combining and aggregating the data of IoT objects belonging to business process instances with other process-related data (e.g., semantic data), the controlling component is the process memory. Process analysts and process managers are able to analyze single process instances as well as aggregated instances and even process types. In case of deviations in process performance, an alert-event is generated and communicated to the person in charge (Section 2.3) or other subscribers. The employee in charge can react to the identified issues to control the process in a flexible way, or change the process.

### 2.3 User Process Interaction

One major topic in ADiWa is the process-embedded support of users and the preservation of their process know-how. Leveraging the users' know-how from daily process work is one of the challenges for business process management (Riss et al., 2005). Therefore, several projects tackle this issue, such as Apodle[1], ACTIVE[2], and Nepomuk (Riss et al., 2007), to name a few. However, no approach has yet to consider exploiting know-how from processes traceable via the IoT.

Nonetheless, supporting users through the IoT, such as smart items in instrumented environments, has recently gained momentum. Applications range from product tracing, such as in SemProM (Schneider and Kroener, 2008), to knowledge-based assistance, such as memory externalization (Kawamura et al., 2007). The novelty in ADiWa consists of the comprehensive approach to the use of the IoT to let users intuitively participate in, intervene, and modify business processes. This approach is complemented by means for process-embedded information support based on IoT-enriched user and process context.

Investigating the project's industry scenarios, we find processes which are often coarsely modeled (if at all), partly unspecific, dynamic, and human-centric, as well as human-individual parts. In those cases, process participants must react to new situations (e.g., those which can be tracked by events from the IoT) and often have to adapt a running process. Participants range from office workers to process-embedded mobile workers interacting with smart objects in an instrumented environment. ADiWa empowers users with tools to understand processes and provides detailed information on involved objects in order to allow controlling, optimization, or different kinds of intervention such as in exceptional cases. It supports users in coping with the vast amount of newly-available information from the IoT.

ADiWa's **interactive user workspace** (see Figure 1) enables process participants to flexibly work on process steps by introducing agile task management, collaboration, and knowledge work support. Here, a configurable process information cockpit provides required information, such as process instance data, data from legacy systems, and occurred events relevant to the user. This approach is combined with proactive information delivery to satisfy the presumed information need inferred from the process and user context. The workspace presents relevant information objects (such as documents, notes, guidelines, and data from smart objects and their product memory) as well as alternative actions from process know-how (such as process guidelines, former process instances, and similar problem solutions). Besides supporting knowledge reuse, the environment is designed to automatically capture process knowledge during process execution.

ADiWa explicitly focuses on enabling users to execute processes in reality while tracking their working activities automatically, and thus, keeping up-to-date process information in the system. This function ranges from giving feedback (e.g., noting reasons for exceptions) and supporting solution-finding in the task management environment (e.g., making a checklist for handling an exception) to directly interacting with smart objects (e.g., to solve an exception). Moreover, it is also tracked as process audit data. For instance, in the retail scenario, rearranging goods in a store due to an exception handling does not require a manual reporting of transferred goods, because this is already traced by the IoT events sent by the instrumented shelves. Comparing the process model to the actual execution based on traced data, provides a valuable basis for performance analysis and optimization.

ADiWa provides a platform for multi-modal interaction with a wide range of usage paradigms and interfaces. Within the latter, participants see their tasks, required information, and supposed actions, as well as any changes in the dynamic processes. They are able to interact with smart objects, retrieve additional information, and provide feedback on process steps.

Another innovative paradigm applied is the digital pen and paper interaction in mobile scenarios. Writing with the digital pen on a specially-prepared paper can be interpreted by software either after manual synchronization or by direct streaming (via Bluetooth). Applications range from simple note-taking or filling out a form with handwriting recognition to selecting commands on a command sheet to control software. By introducing this as IoT source in ADiWa, users are able to do their process work on paper while being seamlessly embedded in the business process and providing direct feedback. In conjunction with the controlling phase explained in Section 2.2,

[1] www.aposdle.tu-graz.at
[2] www.active-project.eu

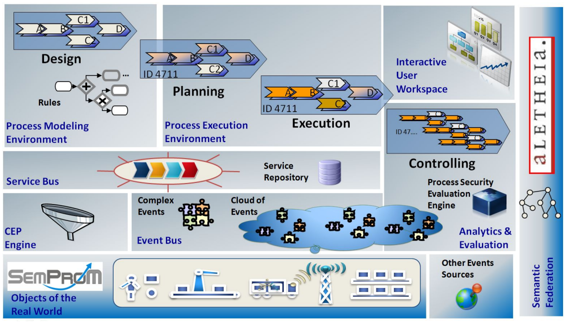


Figure 1: ADiWa big picture of components.

process participants are also able to intuitively combine context data from various sources including IoT and process data during the planning and execution phase. They can aggregate performance indicators and include them in the interactive user workspace as widgets. These widgets are used as working tools and can be shared between users and jointly evolved. A widget configuration can be done without the intervention of IT departments, as it is required in business process management nowadays.

## 3 ARCHITECTURE AND SECURITY

The core architectural paradigm of ADiWa is **event-driven business process management**. To enable this vision, a reference architecture (see Figure 1) is defined, which defines and integrates the most important and influential architectural concepts and mechanisms. All modules within the reference architecture are self-contained and modular entities with clearly-defined external interfaces. The interoperability will be maintained through the development of Web services conforming to WS-I[3] guidelines. Partner projects of ADiWa within the *Digital Product Memory* Innovation Cluster play an important role. The SemProM[4] project focuses on the collection and representation of information during an object's life-cycle. The Aletheia[5] project provides a holistic view of an item's state. The ADiWa reference architecture consists of some key components which are essential for event-driven processes.

- The **event bus** and the **CEP** enable the routing, distribution, and the event processing algebra-based analysis of events.
- The **service bus** serves as central orchestration and integration entity of the different components based on a system-wide service registry.
- The **process modeling environment** is the main component for design-time business process management, including maintenance of business rules.
- The **process execution environment** is the component for business process run-time, especially for planning and execution of process instances.
- An **interactive user workspace** assists the human with appropriate context information and consumable information for effective decision-making.
- The **analytical and evaluation component** collects information from events and processes to derive meaningful representations.

Dynamic business processes consume a large number of events from intelligent objects and base decisions on those events. Planned as well as ad hoc changes in complex environments must not violate security policies. The architecture needs to enforce security polices despite of weak security mechanisms of intelligent objects and dynamic environments. Security properties of events must be checked in spite of their large number and high frequency.

There are several business process-based approaches for the analysis and specification of security requirements and controls. Rodriquez et al. (Rodriguez et al., 2007) present their method for secure business process specification to annotate UML activity diagrams with an UML profile. Modeling security goals in business processes more explicitly enables Wolter et al. (Wolter et al., 2009) to adapt their policy transformation process to different target security infrastructures. Basin et al. (Basin et al., 2006) present with SecureUML a flexible approach developing special purpose UML dialects for security specification and analysis. Neither of them seamlessly integrates multiple models horizontally or incorporates support for cross-phase modeling as is proposed in ADiWa.

Accordingly, we define a guideline for the systematic analysis of security requirements within the context of dynamic business processes. Further, we derive a framework for an integrated security modeling of dynamic business processes. Using weaving models (Bezivin et al., 2006), we link security-related artifacts of dynamic business processes. The interconnectivity allows for the analysis of security-related dependencies and serves as a basis for immediate impact analysis, policy derivation, and security status evaluation. Moreover, we formulate a method to infer and enforce secure interactions with smart objects at runtime. Low-level security polices are decoupled from security expert knowledge and

[3] www.ws-i.org
[4] [http://www.semprom.org/]
[5] [http://www.aletheia-projekt.de/]

(frequently changing) implementation details utilizing description logics and ontologies. Thus, we are developing a process security evaluation engine providing operators with insight concerning the current security status of a process. The engine analyzes the events consumed or generated by the dynamic business process using the integrated security model and the derived security policies. It is capable of simulating possible execution paths and detecting potential security breaches in the near future. Trustworthiness of the events to be processed is at the heart of secure event processing. In ADiWa, we are developing methods to establish trusted relationships between event processing components to avoid the overhead of authenticating each event individually. Plausibility checks are used to detect compromised components.

## 4 OUTLOOK

In this paper, we provide a very brief overview of the motivation, progress, and goals of the ADiWa project. In line with the overall concept, the different facets of the research activities deliver a holistic solution framework that covers the various challenges in the field of event-based system implementations. While the described event processing functionalities increase the needed transparency, the enhanced business model design and execution improve the flexibility of business process adaptations. By that, we aim to close the gap between technical possibilities and their business usage and, thus, to demonstrate the business value of our overall concept. In this respect a deep and thorough evaluation of the concept and the functionalities is our benchmark. Leveraged by our research activities, we intend to raise the research field of event-based system implementations one step higher and to provide a fertile ground for future research projects.

## ACKNOWLEDGEMENTS

The project was funded by means of the German Federal Ministry of Education and Research under the promotional reference 01IA08006. The authors take the responsibility for the contents.

## REFERENCES

Basin, D., Doser, J., and Lodderstedt, T. (2006). Model driven security: From uml models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology*, 15(1):39–91.

Bezivin, J., Bouzitouna, S., Del Fabro, M., Gervais, M., Jouault, F., Kolovos, D., Kurtev, I., and Paige, R. (2006). A canonical scheme for model composition. *Lecture Notes in Computer Science*, 4066:346–360.

Dadam, P. and Reichert, M. (2009). The ADEPT project: a decade of research and development for robust and flexible process support. *Computer Science-Research and Development*, 23(2):81–97.

Graessle, P., Schacher, M., Grc, P., et al. (2006). Agile Unternehmen Durch Business Rules. *Springer*.

Hallerbach, A., Bauer, T., and Reichert, M. (2009). Configuration and management of process variants. *Handbook on BPM*.

Hapner, M., Burridge, R., Sharma, R., Fialli, J., and Stout, K. (2002). Java Message Service Specification v1. 1. *Sun Microsystems, Inc*.

Heusler, K., Stolzle, W., and Bachman, H. (2006). Supply Chain Event Management Grundlagen, Funktionen und potenzielle Akteure. *Wirtschaftswissenschaftliches Studium*, 35(1):19.

Hinze, A., Sachs, K., and Buchmann, A. (2009). Event-based applications and enabling technologies. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, page 1.

Kawamura, T., Fukuhara, T., Takeda, H., Kono, Y., and Kidode, M. (2007). Ubiquitous memories: a memory externalization system using physical objects. *Personal and Ubiquitous Computing*, 11(4):287–298.

Keller, G., Nuettgens, M., and Scheer, A. (1992). Semantische prozessmodellierung auf der grundlage ereignisgesteuerter prozessketten (epk). *Veroeffentlichungen des Instituts fuer Wirtschaftsinformatik*, 89.

Luckham, D. (2002). The power of events: an introduction to complex event processing in distributed enterprise systems. *Springer*.

Muhl, G., Fiege, L., and Pietzuch, P. (2006). Distributed Event-Based Systems. *Springer*.

Paschke, A. and Kozlenkov, A. (2009). Rule-Based Event Processing and Reaction Rules. In *Proceedings of the 2009 International Symposium on Rule Interchange and Applications*, pages 53–66. Springer.

Pesic, M. and van der Aalst, W. (2006). A declarative approach for flexible business processes management. *LNCS*, 4103:169.

Riss, U., Rickayzen, A., Maus, H., and van der Aalst, W. (2005). Challenges for Business Process and Task Management. *Journal of Universal Knowledge Management*, 0(2):77–100.

Riss, U. V., Jarodzka, H. M., and Grebner, O. (2007). Pattern-based task management & implicit knowledge. In *4th Conference on Professional Knowledge Management*. GITO Verlag Berlin.

Rodriguez, A., Fernandez-Medina, E., and Piattini, M. (2007). M-BPSec: A method for security requirement elicitation from a UML 2.0 business process specification. *Lecture Notes in Computer Science*, 4802:106.

Schmidt, B. and Schief, M. (2010). Towards Agile Business Processes Based on the Internet of Things. *Advanced Manufacturing and Sustainable Logistics*.

Schneider, M. and Kroener, A. (2008). The smart pizza packing: An application of object memories. In *IE'08*.

von Ammon, R., Greiner, T., Paschke, A., Springer, F., and Wolff, C. (2008). Event-Driven Business Process Management. *OBJEKTSpektrum*.

Wolter, C., Menzel, M., Schaad, A., Miseldine, P., and Meinel, C. (2009). Model-driven business process security requirement specification. *Journal of Systems Architecture*, 55(4):211–223.