Event-Driven Services: Integrating Production, Logistics and Transportation

A. Buchmann[1], H.-Chr. Pfohl[2], S. Appel[1,*], T. Freudenreich[1,*], S. Frischbier[1,*], I. Petrov[1], C. Zuber[2],

[1] Databases and Distributed Systems Group, TU Darmstadt
{*lastname*@dvs.tu-darmstadt.de}

[2] Chair of Management & Logistics, TU Darmstadt
{pfohl@bwl.tu-darmstadt.de}

Abstract. Today's production processes are characterized by global supply chains, short lifecycles, and an increasing personalization of goods. To satisfy the demands for agility we must integrate the production with the logistics processes and knowledge about the underlying transportation services and infrastructure. This requires continuous monitoring and reacting to events. Service-oriented architectures have provided a platform for structuring services within and across enterprises. However, for an effective monitoring and timely reaction to emerging situations it is crucial to integrate event processing and service orientation. In this position paper we show how event processing and service orientation can be combined into an effective delivery platform for an integrated coordination of the flow of goods. We show how simple events, e.g. RFID tag detections or simple sensor readings, can be integrated into abstract events that are meaningful to invoke logistics services and improve the celerity of responses. We propose filtering, aggregating, and on-the-fly analysis of the continuous flow of events and make events persistent in an event warehouse for auditability and input to future planning processes.

Keywords: Events, complex event processing, monitoring, services, event warehouse, agility, production, logistics, transportation.

1 Introduction and Status Quo

Integrating Production, Logistics and Transportation: Current trends in manufacturing show shortened product lifecycles and an increased personalization of goods with the resulting smaller lot sizes and more frequent retooling. New concepts like just in sequence production or the possibility of global sourcing and distribution have increased the complexity of logistics and the integration of transport systems [8]. In addition the recent consciousness about new objectives in the value creation – like the

*) Funding by German Federal Ministry of Education and Research under research grants ADiWa (01IA08006) and Software Cluster EMERGENT, by Hessen Ministry of Higher Education, Research and the Arts under research grant LOEWE Dynamo PLV, and by Deutsche Post AG.

product's carbon footprint – have led to an extension of the cost-based optimizations of production and distribution strategies by ecological and social aspects. As a result, we must consider the problems of production, logistics and the most efficient use of the underlying transportation services and infrastructure in a comprehensive manner that allows us to optimize for multiple goals. Decentralized decision making and the flow of information across multiple domains and enterprise boundaries play a crucial role in this situation. Tomorrow's software systems have to be agile in terms of integration and adaption to meet the requirements of modern supply chain management.

Service-oriented Architecture (SOA) - Drawbacks and Opportunities: On this way towards dynamic processes, service orientation has become a key concept to enable modularization of information systems and the integration of legacy systems. By encapsulating a given functionality as a service and providing a standardized interface it becomes much easier to build new systems and to adapt to rapidly changing demands. However, current enterprise software systems based on SOA are mostly custom-tailored to the vital needs of a particular organization, reflecting individual workflows, semantics and contexts. Although SOA is a promising approach to reduce functional redundancy and syntactical complexity [4], inter-organizational integration is still hard to achieve. Furthermore, service orientation was conceived with a classical request-reply invocation paradigm in mind: the consumer of a service requests a service from a service provider and receives a response. This is the paradigm used in client-server architectures, such as typical database systems. In the case of services the invocation of a service can be either direct or through some form of mediator that helps in locating the proper service, but the initiation of an interaction is still an explicit request.

Introducing the Event-based paradigm: Pure service orientation with a request-reply interaction is ill-suited for today's cyberphysical systems in which streams of heterogeneous events from different domains and their associated data are continuously produced by sensors and embedded systems and can provide real time information on many operational entities. Typical low-level events are the RFID tag readings detected at a warehouse gate, the GPS coordinates of a truck, or the readings of a temperature sensor in a container with food or pharmaceutical goods. These simple, low-level events can be combined and more abstract and application relevant events can be derived from them. For example, knowing the GPS position of a truck and the traffic conditions ahead, we can derive a complex event that tells us that the truck will arrive two hours late at its destination. The responses to such detected situations can and should be packaged as services. However, the manner in which they are *automatically* invoked is different. Events are produced and many different parties within different contexts may show interest in certain events. They will be notified that an event happened rather than having to ask for it. This reduces the latency between the occurrence of an event and the proper reaction. It also avoids unnecessary polling cycles.

Integrating SOA with Events: These event-driven services combine the benefits of complex event processing (CEP) and SOA [7]. The resulting event-driven architecture (EDA) [1] encompasses both approaches and tries to provide a common framework to cover requirements both from inside an organization as well as from the outside. Companies are starting to recognize the benefits of this approach. However, there is still a huge gap between potential and realization. Software providers are trying to develop the necessary event processing platforms. These are needed, just as database management systems were needed as common platforms decades ago. The end users are trying to understand how to use these platforms to their advantage; at this many open issues exist on both sides [2].

Technological Requirements: From a technological point of view there is the need for powerful notification services that can be distributed and scale appropriately [3], well-defined and standardized languages for event definition, operations on event streams, efficient persistence mechanisms for massive streams, integration of transactional behavior in stream processing, heterogeneity, and security and privacy are just some issues.

Application Requirements: From an application point of view the issues are, among others, the definition of relevant domain-specific events, mechanisms for combining the flood of events with operational data kept in databases, persisting events for auditability purposes and for future planning, decide on an appropriate level of aggregation for events, and how to design and deploy in an effective manner software systems that take full advantage of event-driven services.

Related Projects: In this position paper we sketch an approach that we are investigating in three joint projects between university researchers, major software companies, and big players in the logistics, transportation and production domain. In the Dynamo PLV project we address the issues how to integrate the three domains, production, logistics and transportation, and how to provide the proper infrastructure both from an IT perspective as well as domain-specific. In the ADiWa project we concentrate on issues of quality of service in event delivery and composition. In the Software Cluster project EMERGENT we concentrate on the lifecycle of events and software engineering of event-driven systems to ensure a high level of interoperability between systems based on different domains and platforms.

2 Events

The logistics domain has pioneered many of the automated tracking and tracing applications to afford a more efficient supply chain management. Some sectors, such as air cargo, have successfully defined and implemented a set of domain-specific events and milestones in the context of the Cargo 2000 program to simplify quality management [6]. Examples of such milestones are “Pick Up from Customer” or “Departure” (airline event). Existing software can monitor standardized messages and raise failure alerts or produce route maps with comparisons of planned and actual milestones. This is a very good first step, but in its present form is only applicable to the air cargo sector. Our goal is to develop a platform that would allow us to extend the Cargo 2000 approach to other transportation sectors, such as rail, trucks and combined traffic & transport as well as the integration of intralogistics, and to provide the necessary event definition language.

Further, Cargo 2000 milestones represent logistics processes only with a coarse granularity. Thus another goal is to integrate low-level events that are automatically detected into the whole process. For example, GPS data or events from the German toll system that tracks all trucks above 12 tons when they use the German Autobahn system. Besides the technical problems derived from such a use of truck sightings by the network of installed sensor nodes, it raises a series of interesting privacy and security issues that must be addressed.

The goal is to exploit the same events in many different applications. To this end, a flexible subscription mechanism is needed to enable new services to subscribe to already available events. Furthermore, different services may require existing events as input to compose more complex events in different constellations. We show this process schematically in Figure 1. Underlying this simplified schematic is the need for a well-defined event algebra, an event consumption policy, and an event lifecycle management. The event algebra defines the operators that allow us to combine events, the consumption policy determines in which order events must be consumed (e.g. chronologically or always the latest) and the lifecycle management determines, when an event can be discarded and what events must be made persistent. In [5] we define all the basic terms and identify the event processing requirements of many applications in different domains, including logistics, supply chain management and traffic monitoring and management. This work must be expanded to include event flows in production environments and shop floor control.

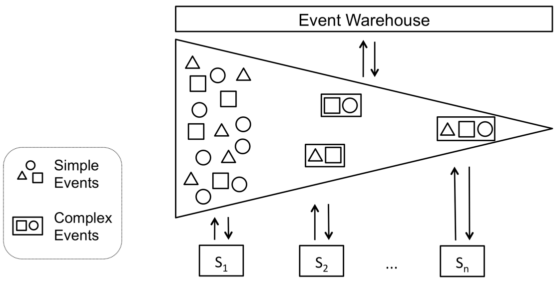

Figure 1: Complex Event Processing by Services S_1 to S_n

3 Event Warehouse

While traditional data management is concerned with storing transactional data (OLTP) and data warehousing is concerned with consolidating operational data for analysis purposes (Business Intelligence), event processing deals with on-the fly processing of streams of heterogeneous events or rather their representation, the event objects. Events are filtered, aggregated and analyzed on the fly. The resulting business events must be reported in a timely manner.

Many events have only transient importance or do not have subscribers and can be discarded. It would unnecessarily clutter the database to store every RFID tag reading or very dense traces of truck trajectories. On the other hand, for planning and simulation purposes as well as auditability all the pertinent events must be recorded.

Finding the right granularity for making events persistent is one of the goals of our current research. Equally important is to determine what underlying events and what context information must be made persistent when storing higher-level business events that were derived and used in the decision making process.

A comprehensive approach to production, logistics and transportation requires the design and execution of complex simulation models. The weak point of many models is the unreliable and incomplete set of input data and the assumptions that are made to compensate for it as well as the missing interfaces of each model to communicate with each other. Therefore, the idea of event warehouses (EW) is a new trend in research [9, 10]. The EW plays a central role in providing a complete and realistic set of events and traces that can be used to execute what-if scenarios, to calibrate the models that are being used by comparing their predictions against recorded data, and in the development of new adaptive strategies.

4 Outlook

Event-driven services are the key to integrating SOA and CEP. Services that are triggered by events rather than invoked, be it manually or on a timed basis, offer a much better responsiveness to business needs. By making events into first class citizens, we can capture the dynamics of complex production, logistics and transportation systems. The event warehouse serves as a repository of meaningful business events as substructure of decisions with all its impacts. It thus captures the result of on-the-fly analytics as well as the underlying system dynamics and context information required for reconstruction of relevant situations. It provides the necessary inputs to models for comprehensive simulation of complex production, logistics and transportation systems.

References

1. Chandy, K. and W. Schulte, *Event Processing: Designing IT Systems for Agile Companies*. 2010: McGraw-Hill, Inc.
2. Frischbier, S., K. Sachs, and A. Buchmann. *Evaluating RFID Infrastructures*. in *2nd Workshop RFID Intelligente Funketiketten - Chancen und Herausforderungen*. 2006. Germany.
3. Guerrero, P., et al. *Pushing Business Data Processing Towards the Periphery*. in *23rd IEEE International Conference on Data Engineering*. 2007. Turkey.
4. Herr, M., U. Bath, and A. Koschel, *Implementation of a Service Oriented Architecture at Deutsche Post MAIL*, in *Web Services*, L.-J. Zhang and M. Jeckle, Editors. 2004, Springer Berlin / Heidelberg. p. 227-238.
5. Hinze, A., K. Sachs, and A. Buchmann. *Event-based applications and enabling technologies*. in *3rd ACM International Conference on Distributed Event-Based Systems*. 2009. USA.
6. IATA - International Air Transport Association. *Cargo 2000*. 2010; Available from: http://www.iata.org/whatwedo/cargo/cargo2000/Pages/index.aspx.
7. McGovern, J., et al., *Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations*. 2006: Springer-Verlag New York, Inc.
8. Pfohl, H.-C. and H.P. Buse, *Inter-organizational logistics systems in flexible production networks: An organizational capabilities perspective*. International Journal of Physical Distribution & Logistics Management, 2000. **30**(5): p. 388-408.
9. Roth, H., et al. *Event data warehousing for Complex Event Processing*. in *4th International Conference on Research Challenges in Information Science*. 2010.
10. Winter, R. and P. Kostamaa. *Large scale data warehousing: Trends and observations*. in *26th IEEE International Conference on Data Engineering*. 2010. USA.