

# Das Paradigma der Event-Driven Architecture als Grundlage für ein entscheidungsunterstützendes IT-System im Zivil- und Katastrophenschutz

Robin Marterer, Benedikt Birkhäuser, Rainer Koch

Universität Paderborn

Computeranwendung und Integration in Konstruktion und Planung (CIK)

Pohlweg 47-49

33098 Paderborn

{marterer, b.birkhaeuser, r.koch}@cik.uni-paderborn.de

**Abstract:** Entscheidungsprobleme, denen sich die strategischen Führungsebenen der Feuerwehr im Krisenmanagement stellen müssen, sind als hoch komplexe, stark vernetzte, intransparente und dynamische Entscheidungsprobleme anzusehen. Im Zuge vorangegangener Projekte hat sich herausgestellt, dass insbesondere im Bereich der strategischen Ressourcenkoordination Unterstützungspotential liegt. Aktueller technologischer Fortschritt im Bereich der Informations- und Kommunikationstechnologie in der Domäne der zivilen Sicherheit ermöglichen die Weiterentwicklung dieses Ansatzes. Das Projekt PRONTO verfolgt einen ereignisbasierten Ansatz um die skizzierten Herausforderungen durch ein echtzeitfähiges IT-System zu lösen. Aufbauend auf diesem Projekt werden in diesem Paper Aspekte der Anwendung einer Event-Driven Architecture in der Domäne der zivilen Sicherheit erörtert und diskutiert.

## 1 Einleitung

Entscheidungsträger der Feuerwehr stehen im Kontext der Einsatzführung vor komplexen Herausforderungen. Vor diesem Hintergrund stellt sich die Frage, wie eine angemessene Unterstützungsfunktionalität umgesetzt werden kann. Dieses Paper erörtert und diskutiert, inwieweit der Ansatz der Event-Driven Architecture einen Beitrag zur Realisierung liefern kann. In Abschnitt 2 werden dazu Forschungsergebnisse und Grundlagen beschrieben, die eine Einordnung der auftretenden Herausforderungen ermöglichen. Abschnitt 3 erläutert die derzeitigen, technologischen Strömungen, auf denen das Konzept der Event-Driven-Architecture aufbaut. Die Umsetzung der skizzierten Ansätze im Projekt PRONTO wird in Abschnitt 4 erläutert. Abschliessend wird versucht auf Grundlage der ersten Projektergebnisse ein Fazit zu ziehen.

## 2 Grundlagen und Herausforderungen

Das Projekt PRONTO betrachtet dazu unter anderem die Fallstudie 'Emergency Rescue Operation' (ERO), in der die Ansätze für den Bereich der Feuerwehr aufgegriffen und implementiert werden. Forschungsschwerpunkt des CIK ist der Bereich der zivilen Sicherheit, der somit im Rahmen dieses Papers fokussiert wird.

Im Folgenden werden die wesentlichen Grundrisse der Einsatzstrukturen im Zivil- und Katastrophenschutz am Beispiel der Feuerwehr Dortmund erläutert, die auf unterschiedlichen gesetzlichen Regelungen basieren. [FWD03, NW, Dor] Der Gesamtzuständigkeitsbereich der Feuerwehr ist in Einsatzbezirke aufgeteilt, für die jeweils eine Feuerwache zuständig ist. Wie viele und welche taktischen Einheiten und Einsatzfahrzeuge alarmiert werden, wird durch eine Alarm- und Ausrückeordnung (AAO) definiert. Im Einsatzfall ist die Feuerwehr hierarchisch organisiert, folglich existiert zu jedem Einsatz genau eine Person als Einsatzleiter, welche die Verantwortung trägt. In einer einfachen Betrachtungsweise können operative, taktische und strategische Führungsebenen unterschieden werden. Auf der strategischen Entscheidungsebene kommt in großen Schadenslagen ein strategischer Einsatzstab der Feuerwehr zum Einsatz, der im administrativen Bereich mit dem Krisenstab zusammenarbeitet.

Ziel von PRONTO ist es, Entscheidungsträgern der beschriebenen Domäne eine IT-basierte, strategische Einsatzunterstützung zur Verfügung zu stellen. Einsatzlagen, in denen strategische Entscheidungsträger und Stäbe zum Einsatz kommen, zeichnen sich durch hohe Komplexität, starke Vernetztheit, Intransparenz und große Dynamik aus. [PO92, BPK09] Eine Grundannahme von PRONTO ist es, die wesentlichen Informationen aus der realen Welt als Ereignisse aufzufassen und diese als Software-Objekte in ein IT-System zu überführen, um sie sammeln und miteinander in Bezug setzen zu können. Viele der für Entscheidungsträger bedeutungsvollen Informationen sind oft prinzipiell vorhanden, stehen ihnen aber in den durch hohen Entscheidungsdruck geprägten Situationen nicht (schnell genug) zur Verfügung oder sind unerreichbar in unterschiedlichen Systemen verborgen. Die Verwendung eines ereignisorientierten Ansatzes bringt eine Entlastung der Entscheidungsträger durch die überwiegend interaktionslose Bereitstellung von Informationen mittels Push-Ansatz mit sich.

## 3 Stand der Technik

Zu den Zwecken der Anwendungsintegration und der Abbildung von Geschäftsprozessen auf IT-Systeme hat sich in den letzten Jahren für viele Anwendungsbereiche das Paradigma der *Service-Oriented Architecture* (SOA) als zielführend erwiesen. Sprachen wie die XML-basierte *Business Process Execution Language for Web Services*<sup>1</sup> (WS-BPEL) dienen dazu, Geschäftsprozesse flexibel zu modellieren und durch die Komposition von *Web Services* zu realisieren. Auf der Implementierungsseite stehen SOAP, WSDL und UDDI als Mittel der Wahl zur Verfügung.

---

<sup>1</sup><http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> (Zugriff: 2010-01-09)

Das Konzept der *Real-Time Enterprise* (RTE) [TYPM09] verfolgt das Ziel, Geschäftsprozesse permanent zu überwachen und kontinuierlich zu verbessern, um Zeit und Kosten einzusparen. Aus einer solchen Unternehmenskultur ergeben sich erweiterte Anforderungen an eine Systemlandschaft. Neben der Anwendungsintegration und der Beschreibung von Geschäftsprozessen erfordert die RTE Echtzeiteigenschaften, die durch herkömmliche SOAs nicht erfüllt werden können. Genau diese Echtzeiteigenschaften sind es, die neben der Anwendungsintegration für die Fallstudie ERO im Projekt PRONTO von besonderer Bedeutung sind.

Das Paradigma der *Event-Driven Architecture*<sup>2</sup> (EDA) [TYPM09] stellt einen aktuellen Ansatz dar, um diesen erweiterten Anforderungen zu begegnen. Eine Referenzarchitektur [DEF<sup>+</sup>08, S. 133] soll nachfolgend der Beschreibung der zentralen Konzepte einer EDA dienen, deren zentralen Artefakte erwartungsgemäß Ereignisse darstellen. Diese können aus beliebigen an das System angeschlossenen Ereignisquellen stammen und werden über einen Ereigniskanal, realisiert durch eine *Message-Oriented Middleware* (MOM), an eine Softwarekomponente für das *Complex Event Processing* (CEP) [Luc02] weitergeleitet. Die MOM aus dem Bereich *Enterprise Application Integration* (EAI) dient neben ihrer Funktion als Transportkanal für Ereignisse auch als Bindeglied zwischen allen an das System angeschlossenen Softwarekomponenten. Hierzu zählen nicht nur verschiedene Systeme, die Ereignisse erzeugen, sondern auch Softwarekomponenten aus Vorgängerprojekten. Zentrales Konzept ist das *Publish-Subscribe*-Entwurfsmuster, das eine flexible Weiterleitung von Ereignissen, basierend auf verschiedenen Ereignistypen, an verschiedene Softwarekomponenten ermöglicht. Das CEP realisiert eine regelbasierte Verarbeitung eingehender Ereignisse. Dies geschieht anhand von durch eine *Event Processing Language* (EPL) formulierten Regeln (auch: Event Pattern), die meist logische, zeitliche oder auch räumliche Zusammenhänge ausdrücken. Die Anwendung dieser Regeln hat die Erzeugung von Ereignissen zur Folge, die einen höheren Informationsgehalt aufweisen. Ein Beispiel für eine EPL stellt *Rapide*<sup>3</sup> [Luc02, S. 145ff] dar. Eingehende oder durch das CEP erzeugte Ereignisse können dazu verwendet werden, Entscheidungsträger in einem Unternehmen über beliebig komplexe Zustände und Vorgänge zu informieren oder Geschäftsprozesse anzustoßen. Die aggregierte Darstellung von Ereignissen und ihren Zusammenhängen lässt sich über Dashboards realisieren und ist dem *Business Activity Monitoring* (BAM) zuzuordnen. Mit [LS08] stellt die *Event Processing Technical Society*<sup>4</sup> (EPTS) eine aktuelle Zusammenfassung der Begrifflichkeiten im CEP-Bereich zur Verfügung. Die Tatsache, dass alle großen – wie auch kleinere Anbieter – von Middleware-Produkten EDA- und CEP-Konzepte in ihre Produkte integrieren, lässt auf das Vorhandensein einer entsprechenden Nachfrage nach solchen Technologien auf dem Markt schließen. Im Open-Source-Bereich steht *N/Eesper*<sup>5</sup> zur Verfügung.

Eine andere Bezeichnung für eine Ereignisquelle innerhalb eines EDA-Systems ist auch 'Sensor'. Dieser Begriff spiegelt den Charakter wider, der die meisten Ereignisquellen beschreibt. Oft handelt es sich um Hardware mit assoziierten Softwarekomponenten, die im Zusammenspiel Ereignisse der realen Welt (physikalische Phänomene) erkennen und in

---

<sup>2</sup>auch: *Event-Driven Service-Oriented Architecture* (ED-SOA), *SOA 2.0* oder *advanced SOA*

<sup>3</sup><http://complexevents.com/rapide> (Zugriff: 2009-01-09)

<sup>4</sup><http://www.ep-ts.com> (Zugriff: 2010-01-09)

<sup>5</sup><http://esper.codehaus.org> (Zugriff: 2010-01-09)

durch Computer verarbeitbare Formate (digitale Werte) überführen. Beispiele hierfür sind Temperatur-, Beschleunigungs- und Wasserstandsensoren – aber auch Mikrofone, Videokameras, RFID-Transponder und GPS-Geräte.

Eine Initiative, die sich der standardisierten Verknüpfung von Sensoren und Web Services verschrieben hat, ist das *Sensor Web Enablement*<sup>6</sup> (SWE). Hier bedient man sich des Konzepts *Data Fusion* [LHL09] zum Zweck der „Kombination von Daten (oder Informationen), um den Zustand eines Aspekts der Welt zu beschreiben oder zu prognostizieren“<sup>7</sup>. Die Parallelen zu EDA und CEP sind offensichtlich. Die Anwendbarkeit des SWE auf die Domäne des *Zivil- und Katastrophenschutzes* wurde bereits im EU-Projekt OSIRIS<sup>8</sup> erforscht.

## 4 Adaption auf PRONTO

Die Anforderungsanalyse im Projekt PRONTO ergab zahlreiche Herausforderungen, die an dieser Stelle nur grob skizziert werden können und im Rahmen des Systemdesigns Berücksichtigung finden mussten: Ein wesentlicher Aspekt ist die Notwendigkeit der Anbindung zahlreicher heterogener Softwarekomponenten. Diese werden von unterschiedlichen Partnern auf zum Teil unterschiedlichen Plattformen entwickelt. Zusätzlich sind diese über unterschiedlich robuste und performante Netzwerke an das System angebunden. Berücksichtigt werden müssen zusätzlich Legacy-Module, deren Anbindung ebenfalls sichergestellt werden muss. Der breite Ansatz von PRONTO mündet im Bedarf einer hohen Flexibilität in Bezug auf die kurzfristige Anpassung des Systems an sich geänderte Anforderungen. Dies wird im Rahmen der Softwareentwicklung durch einen agilen Ansatz aufgegriffen, muss sich allerdings ebenso in den Designentscheidungen bezüglich der Systemarchitektur widerspiegeln.

Zentrale Aspekte der beschriebenen Referenzarchitektur ließen sich direkt auf die Softwarearchitektur von PRONTO übertragen. Abbildung 1 beschreibt das PRONTO-Gesamtsystem bestehend aus Subsystemen und im Kontext externer an das Demonstrator-System angeschlossener Systeme. Die Architektur unterteilt sich grob in zwei Teile: die verteilte Seite, welche mobile Einheiten umfasst, die Ereignisse über drahtlose Netzwerke an das System übermitteln und die Serverseite, die ebenfalls Komponenten umfasst, welche Ereignisse in das System einspeisen. Zusätzlich dient die Serverseite der Verarbeitung aller eingehenden Ereignisse. Eine zentrale Annahme ist, dass jede Softwarekomponente sowohl in der Rolle des Ereignisproduzenten als auch des Ereigniskonsumenten auftreten kann.

Es wird davon ausgegangen, dass externe Systeme sowohl auf der verteilten als auch auf der Serverseite in der Rolle von Ereignisproduzenten auftreten. Die von ihnen erzeugten Ereignisrohdaten (1a, 1b) werden von den mit ihnen assoziierten Komponenten

<sup>6</sup><http://www.opengeospatial.org/ogc/markets-technologies/swe> (Zugriff: 2010-01-09)

<sup>7</sup>[http://52north.org/joomla/components/com\\_publications/publications/2007/52N\\_EinfuehrungSensorWeb.pdf](http://52north.org/joomla/components/com_publications/publications/2007/52N_EinfuehrungSensorWeb.pdf) (Zugriff: 2010-01-09)

<sup>8</sup><http://osiris-fp6.eu> (Zugriff: 2010-01-09)

*Event Recognition* in den Subsystemen *Event Creation* in ein vom System akzeptiertes Ereignisformat transformiert und an die Komponente *Ereigniskanal* (MOM) im Subsystem *Integration* übermittelt (2a, 2b).

Zu den typischen Sensoren auf der verteilten Seite zählen u. a. GPS-Geräte, Mikrofone, Videokameras, Beschleunigungs-, Treibstofftank- und Wassertanksensoren, die allesamt für gewöhnlich auf verschiedenen Feuerwehrfahrzeugen untergebracht sind. Typische Sensoren auf der Serverseite greifen auf zentrale Systeme wie das Einsatzleitsystem, den Sprech- und Datenfunk sowie externe Services (Wetter, Hochwasserpegel) zu.

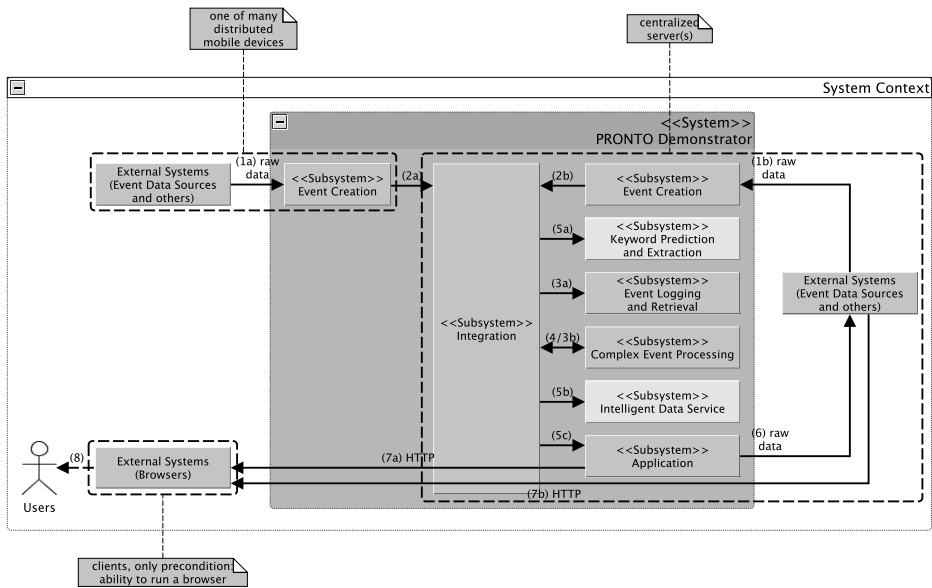


Abbildung 1: Gesamtarchitektur des PRONTO-Systems

Das Subsystem *Event Logging and Retrieval* abonniert über den *Publish-Subscribe*-Mechanismus des Ereigniskanals Ereignisse, die archiviert werden sollen und speichert diese in einer Datenbank, um sie zu einem späteren Zeitpunkt, beispielsweise zu Test- oder Trainingszwecken, wieder abrufen – und ggf. erneut in das System einspielen – zu können (3a). Die dem Subsystem *Complex Event Processing* zugeführten Ereignisse werden als *Low-Level Events* (3b) bezeichnet und innerhalb dieses Subsystems zu *High-Level Events* (4) kombiniert. Sowohl die Software zur Ereignisverarbeitung als auch die zugehörige *Event Processing Language* werden von Projektpartner NCSR entwickelt. Die resultierenden Ereignisse werden in erster Linie vom Subsystem *Application* verarbeitet (5c), welches die Komponenten *Intelligent Resource Management* (IRM), MAP, *Decision Support* (DS) und CHAT enthält. Die Subsysteme *Keyword Prediction and Extraction* (KPX) sowie *Intelligent Data Service* (IDS) werden ebenfalls von Projektpartner NCSR entwickelt. KPX dient der Unterstützung der Spracherkennung durch entsprechende Komponenten in den Sub-

systemen *Event Creation*, die durch Projektpartner FhG geliefert werden. IDS enthält eine Ontologie der Anwendungsdomäne *Zivil- und Katastrophenschutz*, welche für die Berechnungen in den Subsystemen CEP sowie KPX dient. Zunächst wird davon ausgegangen, dass die Benutzer des Systems webbasiert auf das System zugreifen. Hierzu werden die Komponenten im Subsystem *Application* in einer einheitlichen Anwendung integriert und über die Komponente *Web Server* im Subsystem *Integration* zur Verfügung gestellt (7, 8).

Während der gesamten Konzeption des Systems wurde die bundesweite Einführung des TETRA-Digitalfunks für *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS) [Lin08] berücksichtigt. TETRA wird Datenübertragungsmöglichkeiten mit sich bringen, die in der bisherigen, analogen Welt des BOS-Funks so nicht verfügbar waren. Das TETRA zu Grunde liegende TCP/IP-Protokoll ermöglicht eine leichtere Anbindung verteilter Ereignisproduzenten an die auf dem selben Protokoll aufbauenden und bereits etablierten Webtechnologien, die im EDA-Bereich Anwendung finden. Somit können Ansätze der Event-Driven Architecture nun prototypisch auf die Domäne der Feuerwehr übertragen werden. Die erste Generation des BOS-Funks bringt Datenübertragungsmöglichkeiten mit sich, die anfangs noch sehr begrenzt sein werden. Die Übertragungsraten der zweiten Generation werden mit den *TETRA-enhanced Data Services* (TEDS) in etwa im Bereich von *Enhanced Data Rates for GSM Evolution* (EDGE) liegen und somit deutlich performanter sein.

Derzeit befindet sich PRONTO in der Phase der ersten Prototypentwicklung. Für den Ereigniskanal wird derzeit der von JBoss abgeleitete Message Channel HornetQ verwendet. Die aktuellen Prototypen basieren auf HornetQ und verwenden JMS-Nachrichten, um Ereignisse im JSON-Objekt-Format zu versenden.

## 5 Fazit

In weiteren Teilprojekten müssen Fragen der konkreten Ereignisdefinition, der Ereignismodellierung, der Ereigniserkennung sowie der Darstellung der Ereignisse auf der GUI gelöst werden. Zusätzlich wird versucht, die erreichten Ergebnisse auf eine zweite Fallstudie 'City Transport Management' zu übertragen.

Dazu werden in den nächsten Projektschritten die einzelnen Komponenten durch die beteiligten Partner entwickelt werden und in einem ersten, lauffähigen Demonstrator integriert werden. In den weiteren Phasen des Projektes werden die ereigniserkennenden Komponenten sukzessive durch Testläufe auf der Basis von Realdaten verbessert werden. Zusätzliche Herausforderungen stellen sich im Bereich des GUI-Designs, der Systemperformance sowie der Sensoranbindung. Das Gesamtergebnis des Projektes wird im Rahmen einer Übung mit den Endanwenderpartnern evaluiert werden.

Nimmt man die aufgenommenen Anforderungen als ein Bewertungsmaßstab, so konnten bisher alle wesentlichen Anforderungen durch die Entscheidung für eine Event-Driven Architecture erfüllt werden. Eine endgültige Evaluation der getroffenen Designentscheidung kann zu diesem Zeitpunkt nicht erfolgen. Diese kann erst im Zuge des weiteren Projektfortschrittes vor dem Hintergrund der gestellten Anforderungen und zum Beispiel

in Anlehnung an gängige Kriterien für Softwarequalität [ISO01] erfolgen.

## Literatur

- [BPK09] Benedikt Birkhäuser, Jens Pottebaum und Rainer Koch. Unterstützung von Einsatzentscheidungen der Feuerwehr auf Basis IT-unterstützter Kräftekoordination. In Stefan Fischer, Hrsg., *Informatik 2009: Im Focus das Leben: Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 28.9. - 2.10.2009 in Lübeck*, Jgg. 154 of *GI-EditionProceedings*, Seiten 1393–1406. Gesellschaft für Informatik, Bonn, 2009.
- [DEF<sup>+</sup>08] Jürgen Dunkel, Andreas Eberhart, Stefan Fischer, Carsten Kleiner und Arne Koschel. *Systemarchitekturen für verteilte Anwendungen*. Carl Hanser Verlag, München, 2008.
- [Dor] Feuerwehr Dortmund. Brandschutzbedarfsplan der Stadt Dortmund.
- [FWD03] *Feuerwehr-Dienstvorschrift 100: Führung und Leitung im Einsatz - Führungssystem*. Deutscher Gemeindeverlag, Stuttgart, 2003.
- [ISO01] ISO/IEC. ISO/IEC 9126-1:2001, Software Engineering - Product Quality - Part 1: Quality model, 2001.
- [LHL09] Martin E. Liggins, David L. Hall und James Llinas. *Handbook of Multisensor Data Fusion*. CRC Press, Boca Raton, second edition. Auflage, 2009.
- [Lin08] Christof Linde. *Aufbau und Technik des digitalen BOS-Funks*. Franzis Verlag, Poing, 2008.
- [LS08] David Luckham und Roy Schulte. Event Processing Glossary - Version 1.1, 2008.
- [Luc02] David Luckham. *The Power of Events*. Addison-Wesley, Pearson Education, Boston, 2002.
- [NW] Landtag Nordrhein-Westfalen. Gesetz über den Feuerschutz und die Hilfeleistung Nordrhein-Westfalen: FSHG: 10.02.1998.
- [PO92] Wiebke Putz-Osterloh. Entscheidungsverhalten. In Erich Frese, Hrsg., *Handwörterbuch der Organisation*, Jgg. 2 of *Enzyklopädie der Betriebswirtschaftslehre*, Seiten 585–599. Poeschel, Stuttgart, 1992.
- [TYPM09] Hugh Taylor, Angela Yochem, Les Phillips und Frank Martinez. *Event-Driven Architecture*. Addison-Wesley, Pearson Education, Boston, 2009.

