

Smart Items in Ereignisgesteuerten Prozessketten

Peter Ibach
Institut für Informatik
Humboldt Universität
Rudower Chaussee 25
D-12489 Berlin
ibach@informatik.hu-
berlin.de

Dirk Bade
Department Informatik
Universität Hamburg
Vogt-Kölln-Straße 30
D-22527 Hamburg
dirk.bade@informatik.uni-
hamburg.de

Steffen Kunz
Institut für Wirtschaftsin-
formatik
Humboldt Universität
Spandauer Straße 1
D-10178 Berlin
steffen.kunz@wiwi.hu-
berlin.de

Abstract: Smart Items sind Objekte, die anhand von Sensoren ihren Zustand sowie Umweltinformationen ermitteln und weitergeben können. Um in ereignisgesteuerten Prozessketten auf diese feingranularen Zustandsinformationen reagieren zu können, ist es notwendig, die Sensordaten zu aggregieren (Objekt-Ebene), so dass ihnen Ereignisse auf höherer Ebene (Prozess-Ebene) zugeordnet werden können. Da dieser Abgleich aus Komplexitätsgründen kaum allgemein möglich ist, verwenden wir in unserem Ansatz eine halbautomatische Lösung, bei der Objektzustände erfasst, zusammengeführt und visualisiert werden, so dass Anwender eine Entscheidung über den weiteren Prozessablauf vornehmen können. Der Ansatz wird beispielhaft anhand eines Logistikszenarios auf Basis der Funkortungssoftware MagicMap vorgestellt. Dabei werden Smart Items durch ihre Funkcharakteristik verortet und auf einer Karte, zusammen mit Zustandsinformationen visualisiert sowie Abweichungen gegenüber der Prozessplanung gekennzeichnet.

1 Einleitung

Radio Frequency Identification (RFID) ist eine der Schlüsseltechnologien des Ubiquitous Computing und dient der computergestützten, eindeutigen Identifikation physikalischer Objekte. In Verbindung mit anderen Sensortechnologien eröffnet sich weiteres Potential. Mit entsprechenden Sensoren ausgestattete „Smart Items“ – wie diese elektronisch aufgewerteten Objekte genannt werden – sind in der Lage, ihren eigenen Zustand zu bestimmen und diesen in einem Netzwerk, dem Internet der Dinge, auszutauschen. Intelligente, informationstechnische Anwendungen erfordern eine Abbildung von diesen Zuständen auf abstrakte vereinfachte Schemata, um auf diese, in großen Mengen anfallenden Zustandsdaten, geeignet reagieren zu können.

Diese Schemata bzw. Modellierungssprachen können je nach Verwendungszweck unterschiedliche Formen annehmen; eine dieser Modellierungsmethoden sind ereignisgesteuerte Prozessketten, mit deren Hilfe sich Prozessabläufe darstellen lassen. Beim Durchlaufen der Prozess-Instanzen (Run-Time) stellt sich jedoch das Problem, diese vereinfachten ex ante vormodellierten (Build-Time) Schemata mit den erfassten Objektzuständen in Übereinstimmung zu bringen. Dabei entsteht in der Praxis oftmals eine Differenz zwischen den physisch erfassten Werten und dem Prozessmodell. Bei einer solchen

Abweichung ist es notwendig, korrigierend einzugreifen, um die Konsistenz der Prozesse wieder herzustellen bzw. zu erhalten.

In diesem Zusammenhang stellen sich drei wichtige Fragen, die wir in diesem Artikel adressieren: (a) Wie lassen sich Sensordaten so aggregieren (Objekt-Ebene), dass ihnen Ereignisse auf höherer Ebene (Prozess-Ebene) zugeordnet werden können, (b) wie lassen sich Abweichungen von (Build-Time) Prozessen und (Run-Time) Prozess-Instanzen feststellen bzw. darstellen und (c) wie ist es möglich, diese Abweichungen zu korrigieren?

Unser Lösungsansatz beinhaltet die Verwendung der Funkortungslösung MagicMap in Kombination mit der Ereignisverarbeitungs-Software JESPA. Damit ist es möglich, Objektzustände aus Sensordaten zu aggregieren und so zu visualisieren, dass Anwender eine integrierte Sicht des räumlichen, zeitlichen und sachlogischen Kontextes der Smart Items erhalten. Hierdurch lassen sich Inkonsistenzen der Objektzustände gegenüber den zur Build-Time definierten Prozessen erkennen und in Interaktion mit dem Nutzer korrigieren.

Der Rest dieses Artikels ist wie folgt aufgebaut: In Abschnitt 2 wird auf aktuelle Forschungsschwerpunkte im Prozessmanagement eingegangen und das Konzept der Ereignisgesteuerten Prozessketten (EPKs) vorgestellt. Abschnitt 3 befasst sich mit der Integration von Smart Items in EPKs, um anschließend einen Lösungsansatz für die zuvor formulierten Fragen vorzustellen. In Abschnitt 4 wird die Umsetzung mit Hilfe des an der Humboldt-Universität entwickelten Funkortung-Systems MagicMap und des an der Universität Hamburg entwickelten Systems JESPA, einer agenten-basierten Architektur zur Verarbeitung von Ereignisströmen, beschrieben. Diese Lösung wird anhand eines konkreten Anwendungsbeispiels aus der Logistik näher erläutert (Abschnitt 5). Im letzten Abschnitt wird der Ansatz zusammengefasst und ein Ausblick auf darauf aufbauende Erweiterungen gegeben.

2 Prozessmanagement mit Ereignisgesteuerten Prozessketten

Um die komplexen Aktivitäten und Interaktionsabläufe heutiger Unternehmen zu beschreiben, bedarf es ausdrucksstarker Modelle. In der Disziplin des Prozessmanagements haben sich in den letzten Jahren verschiedene Modellierungssprachen etabliert, die je nach Verwendung unterschiedlich geeignet sind. Nachfolgend gehen wir kurz auf die aktuellen Probleme im Prozessmanagement ein und stellen eine spezifische Modellierungssprache, Ereignisgesteuerte Prozessketten (EPKs), vor, die in dieser Arbeit für die Darstellung von Prozessabläufen verwendet wird.

2.1 Prozessmanagement

Bei der Modellierung von Prozessen lassen sich zwei Phasen unterscheiden: (i) die Build-Time, zu der ein Prozess definiert wird und (ii) die Run-Time zu der er ausgeführt wird [ARD07]. Der Prozess-Entwurf zur Build-Time stellt dar, wie der Prozessablauf

unter normalen Bedingungen stattfinden sollte. Im Praxiseinsatz kann es jedoch zu Abweichungen von dieser idealisierten Betrachtung kommen, was eine Anpassung des Prozesses zur Run-Time erforderlich macht [WNL08]. Diese Problematik wird in der Prozessmanagement-Forschung sehr aktuell und viel diskutiert [Ad06, ARD07, HCK06, RRF08]. Ein Problem besteht vor allem darin, dass aktuelle Workflow Management Systeme nur geringe bis keine Flexibilität in Bezug auf ad-hoc Änderungen bieten; eine Anpassung der zur Build-Time modellierten Prozesse an die Run-Time-Gegebenheiten ist nicht vorgesehen [RD98]. Diese Flexibilität ist mit zunehmender Vielfalt und Dynamik von Prozessen in allen Unternehmensbereichen und Wirtschaftssektoren jedoch von sehr großer Bedeutung. Eine schnelle und kontextbezogene Anpassung von Prozessen ist für die Wettbewerbsfähigkeit vieler Unternehmen essentiell.

2.2 Ereignisgesteuerte Prozessketten

Um Prozessabläufe darzustellen, werden in diesem Dokument EPKs [KNS92], eine semi-formale Modellierungstechnik zur Darstellung und Dokumentation von Geschäftsprozessen, verwendet. Eine EPK besteht aus:

- (a) **Ereignissen:** zeitpunktbezogene Zustandsübergänge.
- (b) **Funktionen:** Tätigkeiten bzw. Prozessschritte, die sich innerhalb eines Zeitraums vollziehen.
- (c) **Operatoren:** Verknüpfungen wie „und“, „oder“ sowie „exklusivoder“, um mögliche Entscheidungen auf Basis von Bedingungen und Regeln definieren zu können.
- (d) **Kontrollfluss:** Definition von zeitlichen Beziehungen, die sich innerhalb eines Prozesses zwischen Ereignissen, Funktionen und Operatoren ergeben.

Um die Modellierungsmöglichkeiten von EPKs zu erhöhen, wurde deren Funktionalität erweitert. Die so genannten erweiterten EPKs (eEPKs) verfügen über vier zusätzliche Elemente:

- (e) **Informationsobjekte:** Objekte aus denen Informationen geladen bzw. in die Informationen gespeichert werden können, beispielsweise ein Kundenauftrag.
- (f) **Organisationseinheiten:** führen Funktionen aus und benötigen dafür Zugriff auf Informationsobjekte.
- (g) **Prozesswegweiser:** zeigt Verbindungen von bzw. zu einem Prozess auf, so dass zum Beispiel auf vor- oder nachgelagerte EPKs verwiesen werden kann.
- (h) **Anwendungssysteme:** verweisen auf Datenbanken oder andere Anwendungsprogramme.

Die mit Hilfe von EPKs modellierten Prozesse lassen sich in Workflow Management Systeme überführen, wo sie zur Ausführung des Prozessablaufs dienen. Es sei an dieser Stelle jedoch angemerkt, dass hierfür andere Modellierungssprachen, wie Yet Another Workflow Language (YAWL) [AH07] oder Petri-Netze, besser geeignet sind, da die Beschreibungslogik dieser Modellierungssprachen präziser ist. Da EPKs jedoch momentan die wohl am meisten verbreitete Modellierungssprache für Prozesse ist und da deren Verwendung recht intuitiv ist, haben wir uns in dieser Arbeit für diese entschieden.

3 Integration von Smart Items in Ereignisgesteuerte Prozessketten

Das Internet der Dinge, das aus der elektronischen Vernetzung von Smart Items hervorgeht, ermöglicht über die erfassten Sensordaten eine durchgängige Rückkopplung zwischen Prozesszuständen und Objektzuständen. Dies erschließt einerseits einen höheren Automatisierungsgrad in vielen Anwendungen und andererseits neue Möglichkeiten der Prozessüberwachung und -kontrolle [Wo02]. Eine besondere Herausforderung stellt dabei die korrekte Abbildung der Objektzustände auf Prozesszustände dar, welche durch den unterschiedlichen Granularitätsgrad der beiden Abstraktionsebenen begründet ist.

3.1 Smart Items und Prozessketten

Abbildung 1 zeigt die wichtigsten Abstraktionsebenen des Prozessmanagements und setzt sie miteinander in Beziehung. Auf unterster Ebene findet sich die Realität mit einem kontinuierlichen, unendlichen Zustandsraum. Sensoren auf der zweiten Ebene können bestimmte Aspekte der Realität punktuell erfassen. Ihre Messdaten bilden die Grundlage für die Objekt-Ebene, in der sich Objektzustände durch Verdichtung und Aggregation der Sensordaten ermitteln lassen. Die Prozess-Instanz-Ebene wiederum beschreibt zur Run-Time angepasste Arbeitsabläufe für konkrete Objekte. Im Gegensatz dazu wird auf der obersten Ebene, der Prozess-Ebene, weiter von konkreten Instanzen abstrahiert. Auf dieser Ebene spezifizieren zur Build-Time entworfene Prozessbeschreibungen generelle Aktivitäten und Zustandsübergänge von Objekten.

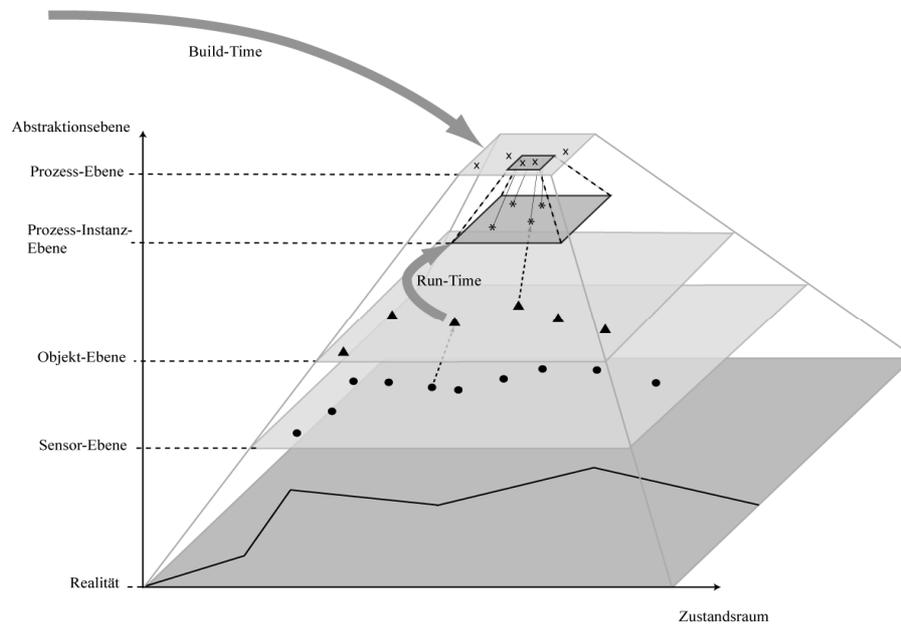


Abbildung 1: Abstraktionsebenen beim Prozessmanagement.

Die Unterscheidung in (Run-Time) Prozess-Instanzen und (Build-Time) Prozesse ist besonders dann von Bedeutung, wenn es möglich ist, kontext- und zeitbezogene Informationen zu sammeln, daraus die sich zur Run-Time ergebende Situation auf Objekt-Ebene zu bestimmen und diese mit dem geplanten Prozessablauf zu vergleichen. Im Falle von Inkonsistenzen kann eine Anpassung der Prozess-Instanz notwendig sein. Genau diese Voraussetzung bieten Smart Items: Sie können mit Hilfe von Sensoren Informationen über sich und ihre Umgebung erfassen und diese in einem Netzwerk mit anderen Objekten austauschen.

3.2 Herausforderungen und Lösungsansatz

Es können eine Reihe grundlegender Fälle identifiziert werden, die eine Abbildung des aus den Sensordaten ermittelten Objektzustandes auf einen bestimmten Prozesszustand und umgekehrt verlangen:

- In welchem Zustand bzw. in welchem Prozessschritt befindet sich ein bestimmtes Objekt der realen Welt?
- Welche Bedingungen zur Ausführung eines Prozessschrittes sind (nicht) erfüllt bzw. von welchen Objekten hängt die Ausführung ab?
- Auf welche Objekte wirkt die Ausführung eines Prozessschrittes?
- Wie lassen sich aus vielen einfachen Prozessereignissen komplexe Vorhersagen für den weiteren Prozessverlauf herleiten?

Hieraus ergeben sich vielfältige Herausforderungen: Informationen über die physische Situation, hier die Positionen und Zustände der Smart Items, müssen wahrgenommen und in ein Kontextmodell überführt werden. Dieses Modell muss kontinuierlich aktualisiert werden, wobei zu entscheiden ist, welche Informationen aktiv von den Ereignisquellen (den physischen Objekten) aktualisiert werden (Push-basiert) und welche Informationen von ereignisverarbeitenden Einheiten abgefragt werden (Pull-basiert).

Des Weiteren bedarf es Methoden, erfasste Sensordaten auf allgemeinere Objektzustände und schließlich auf Prozesszustände abzubilden. Oft wird dies auch als Ableitung von höherem Kontext aus niederem (oder Basis-) Kontext formuliert, siehe beispielsweise [Ni08] (higher-level context, lower-level context). Aufgrund der auf höheren Abstraktionsebenen verdichteten Daten soll für beliebige Objekte entschieden werden können, ob Soll- und Ist-Zustand hinsichtlich des Prozessfortschritts übereinstimmen und welche Prozessschritte, ausgehend von der aktuellen Situation, weitergehend möglich sind. Da der Zustandsraum der möglichen Messungen sehr groß ist, Prozesse jedoch nur wenige diskrete Zustände haben, ist die Abbildung besonders anspruchsvoll. Schließlich müssen dem Anwender die Vorgänge in der physischen Umwelt, die Zusammenhänge zwischen Objekten und Prozessen sowie Abhängigkeiten zwischen Objekten untereinander in geeigneter Weise präsentiert werden, so dass dieser sich einen Überblick über den globalen Zustand sowohl der physischen Welt als auch der Modell- und Prozesswelt verschaffen kann, um gegebenenfalls eingreifen zu können.

Um nun festzustellen, ob die aktuelle Situation noch konsistent mit der geplanten Prozessausführung ist, müssen zunächst die Objektzustände bestimmt werden. Hierfür werden Sensordaten aggregiert und in ein entsprechendes Objektmodell überführt. Anschließend wird versucht, die Konstellation der Objektzustände einem der möglichen Prozessschritte zuzuordnen. Gelingt diese Abbildung nicht, ist dies ein Hinweis darauf, dass sich Prozess-Aktivitäten möglicherweise nicht wie erwartet auf die Realität ausgewirkt haben bzw. unvorhergesehene Ereignisse die Situation so verändert haben, dass für eine erfolgreiche Prozessfortsetzung eine Korrektur erforderlich ist. Korrekturmaßnahmen können dazu die Zuordnung der Zustände zu bestehenden Prozessschritten betreffen oder auch Änderungen der Prozesse selbst erfordern. In einfachen Fällen kann dies automatisch erfolgen, oft aber wird ein manueller Eingriff nötig sein. Um dem Benutzer die Situation zu veranschaulichen und ihn bei dem notwendigen Eingriff zu unterstützen, bedarf es weiterhin einer entsprechenden kontextuellen und räumlichen Visualisierung der Objekte sowie des Prozesszustandes.

Mit Hilfe von MagicMap und JESPA (vgl. Abschnitt 4) lassen sich Objektzustände aus den Sensordaten ermitteln und gegen die Prozess-Ebene verifizieren. Für den Fall, dass sich Zustände der Objekt-Ebene nicht auf Zustände der Prozess-Ebene abbilden lassen, wird der Benutzer auf die Abweichungen aufmerksam gemacht. Er kann dann anhand der räumlichen und kontextuellen Visualisierung in MagicMap die Bedingungen erfassen und geeignet reagieren. Dieser halbautomatische Ansatz ermöglicht eine Anpassung der Prozessausführung an die konkreten Gegebenheiten zur Run-Time. Es sei an dieser Stelle angemerkt, dass mit der Run-Time-Anpassung der Prozess-Instanzen normalerweise keine Anpassung der zur Build-Time entworfenen Prozesse erfolgt. Falls regelmäßig ähnliche Prozessschritte in der angepassten Prozess-Instanz erforderlich werden, so wäre auch eine Anpassung des generellen Prozessmodells sinnvoll.

4 Umsetzung mit MagicMap und JESPA

Im Folgenden werden mit MagicMap und JESPA zwei wesentliche Bausteine unseres Lösungsansatzes vorgestellt, bevor anschließend auf das Zusammenspiel dieser Komponenten sowie die daraus resultierende Prozess- und Benutzerunterstützung eingegangen wird.

4.1 MagicMap

MagicMap ist eine Softwarelösung zur Ermittlung und Visualisierung von Positionen sowie weiteren Zustandsdaten mobiler Objekte [Br07]. Dies umfasst Standard-Geräte wie Smartphones, PDAs oder Laptops, aber auch Smart Items, die durch spezielle Transponder ausgestattet sind. Im Gegensatz zu konventionellen Funkortungslösungen, die lediglich eine Technologie zur Ortung nutzen, unterstützt MagicMap unterschiedliche Technologien durch eine modulare Architektur (vgl. Abbildung 2). Sensordaten werden erfasst, normalisiert und zur Positionsbestimmung ausgewertet. Gegenwärtig unterstützt MagicMap die Ortung über GPS, WLAN, Bluetooth, GSM, ZigBee und RFID und erreicht dadurch eine flächendeckende Ortung und Erfassung sensorischer

Daten sowohl außerhalb als auch innerhalb von Gebäuden. Über die gekennzeichneten Schnittstellen können externe Systeme angebunden werden. Daneben gibt es die Möglichkeit, die Funktionalität über Plugins innerhalb von MagicMap modular zu erweitern.

Dank verbesserter Algorithmen zur Berücksichtigung von Umgebungsstörungen ist mit solchen Standard-Funktechniken eine adäquate Ortungsgenauigkeit möglich. MagicMap erreicht bei Indoor-Tracking-Szenarien in der Logistik Genauigkeiten von 3-5m [Ib05]. Die Vielfalt unterstützter Kommunikationsschnittstellen bietet dabei die Möglichkeit, Sensordaten auszutauschen, beispielsweise Temperatur- oder Beschleunigungswerte. Mit Hilfe von Annotationen lassen sich ferner Informationen zu physikalischen Objekten zuordnen.

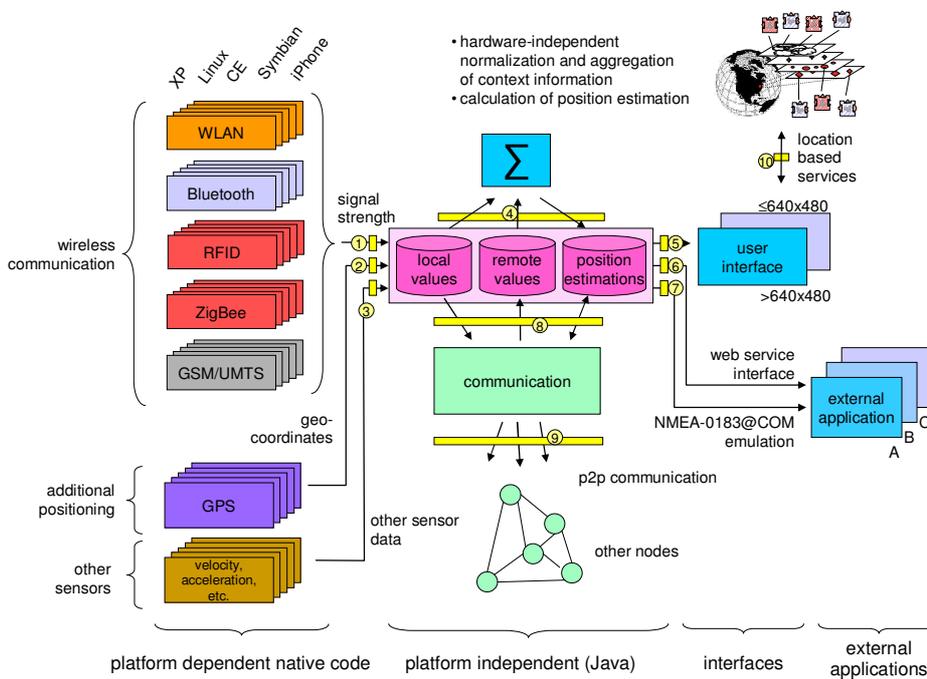


Abbildung 2: MagicMap Architektur

4.2 JESPA

JESPA (Jadex Event Stream Processing Architecture) ist eine verteilte, agenten-basierte Architektur zur Verarbeitung von Ereignisströmen, insbesondere für Sensor- bzw. RFID-Netzwerke [Ba09]. Die Verarbeitung umfasst dabei auf der einen Seite die Detektion von komplexen Ereignismustern (*Complex Event Processing, CEP* [Lu02]), mit Hilfe derer Korrelationen (z.B. zeitliche und kausale Beziehungen) zwischen primitiven Ereignissen (*lower level events*) identifiziert werden können, um daraus komplexe Ereignisse (*higher level events*) zu folgern. Auf der anderen Seite umfasst die Verarbeitung

durch JESPA aber auch die Nachverarbeitung dieser komplexen Ereignisse, z.B. das Filtern, Gruppieren, Transformieren oder Anreichern mit weitergehenden Informationen, um schließlich Anwendungsereignisse (*Application Level Events, ALE*) zu erzeugen, welche die für eine Anwendung relevanten Informationen in einem gewünschten Format enthalten.

Voraussetzung dafür, dass eine Anwendung mittels ALEs über eine bestimmte Korrelation von Ereignissen informiert werden kann, ist die Registrierung eines komplexen Ereignismusters (vgl. Abbildung 3). Im Kern von JESPA arbeitet eine *Esper CEP-Engine* [Esp09], welche für das Identifizieren der registrierten komplexen Ereignismuster in Strömen primitiver Ereignisse verantwortlich ist und eine eigene, an SQL angelehnte Beschreibungssprache für Ereignismuster besitzt. Wurde ein Ereignismuster erkannt, generiert Esper als Ergebnis ein komplexes Event, welches bspw. Informationen der zugrunde liegenden primitiven Ereignisse aggregiert und mittels Attributen repräsentiert.

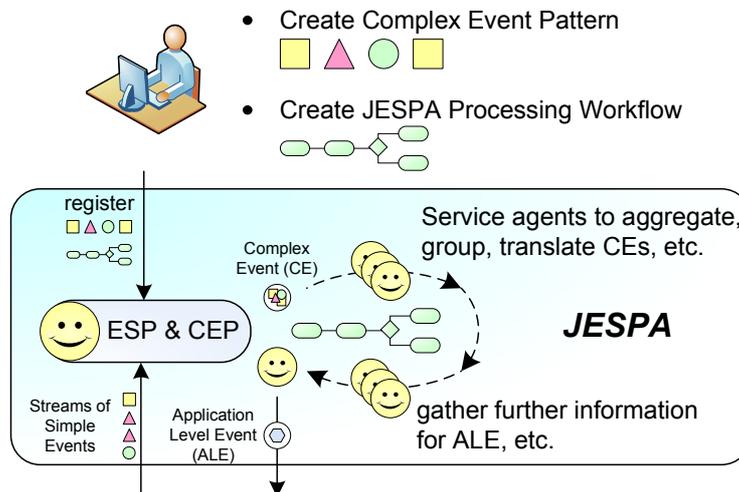


Abbildung 3: Datenverarbeitung in JESPA

Da bestehende Anwendungen dieses Ergebnis unter Umständen nicht direkt handhaben können, kann optional ein *JESPA Processing Workflow* spezifiziert und zusammen mit dem Ereignismuster registriert werden. Dieser Workflow beschreibt die Orchestrierung von Service-Agenten, welche ein komplexes Ereignis in mehreren Schritten in ein ALE transformieren, welches schließlich an die registrierten Anwendungen zurückgegeben wird und von diesen direkt weiterverarbeitet werden kann.

Für die Entscheidung, JESPA zur Verarbeitung von Ereignissen in MagicMap zu integrieren, sprechen verschiedene Gründe, die im Folgenden anhand des Zusammenspiels näher erläutert werden.

4.3 Zusammenspiel von MagicMap und JESPA

Sowohl MagicMap als auch JESPA sind modular aufgebaut und besitzen für ihre wesentlichen Funktionen Schnittstellen, was das Zusammenspiel beider Lösungen unkompliziert macht. MagicMap ist hierbei für die Erfassung von Messwerten zuständig. Diese können lokal auf einem Client erhoben sowie zusätzlich mit anderen Clients ausgetauscht werden und dienen MagicMap zur Berechnung und Visualisierung von Objektpositionen. JESPA zeichnet sich dann für die weitergehende Verarbeitung der Messwerte verantwortlich und versucht Muster in den Messwerten zu erkennen, um daraus den aktuellen Zustand von Objekten abzuleiten. Die Steuerung des Zusammenspiels wurde über ein Plugin „MagicTracker“ für MagicMap realisiert (vgl. Abbildung 4), welches neben JESPA noch eine weitere Komponente zur Abbildung von Objektzuständen auf einen Prozesszustand enthält. Durch diese Komponenten werden die Messwerte der Sensoren sukzessiv zu Objektzuständen und letztlich zu Prozesszuständen abstrahiert, was den Abstraktionsebenen beim Prozessmanagement (vgl. Abbildung 1) entspricht.

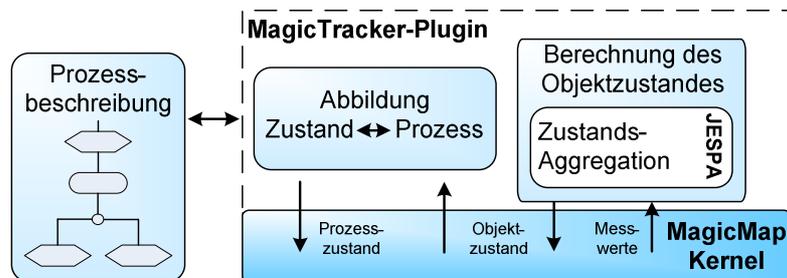


Abbildung 4: Das MagicTracker-Plugin realisiert das Zusammenspiel von MagicMap und JESPA

Um die Messwerte verarbeiten zu können, registriert JESPA „Software-Sensoren“ bei den MagicMap-Schnittstellen, die bei der Erfassung von Messwerten beteiligt sind. Hierzu gehören:

- die Schnittstelle zur Aufzeichnung lokal empfangener Signalstärken, Geokoordinaten und weiterer Sensordaten (vgl. Abbildung 2, Punkt 1, 2 und 3),
- die Schnittstelle zur Positionsberechnung (vgl. Abbildung 2, Punkt 4) sowie
- die Webservice-Schnittstelle, über die ein MagicMap-Client Objektdaten mit anderen Clients austauscht (vgl. Abbildung 2, Punkt 9).

Diese Software-Sensoren leiten die Messwerte als Ereignisse an JESPA weiter, welches den Kern des MagicTracker-Plugins darstellt. Zum Beispiel registriert sich ein Software-Sensor als ein *NodeModelListener* bei MagicMap und wird informiert, sobald aus den Messwerten resultierend neue Objekte hinzugefügt oder Attribute vorhandener Objekte aktualisiert werden. Basierend auf diesen Informationen werden primitive Ereignisse erzeugt und an JESPA übergeben.

Um aus diesem Strom primitiver Ereignisse ein komplexes Ereignis zu generieren, muss ein entsprechendes Ereignismuster bei der Event Processing Engine registriert werden. Das Ereignismuster, dass eine Palette p von Gabelstapler pc an Position pos zum Zeit-

punkt t stärker als der maximal zulässige Wert a (hier 3,0) erschüttert wurde, sieht in der *Esper Event Pattern Language* [Esp09] bspw. folgendermaßen aus:

```
SELECT current_timestamp(), pc.id, pc.pos, p.id, p.sensor.acc
FROM PalletEvent AS p, PalletCarrierEvent AS pc
WHERE pc.load.id = p.id AND p.sensor.acc > 3.0
```

Sobald ein solches Muster im Strom der primitiven Ereignisse erkannt wird, erzeugt die Event Processing Engine ein komplexes Ereignis, welches z.B. die Projektions-Attribute des Ereignismusters enthält, im obigen Beispiel also den aktuellen Zeitstempel, die ID und Position des Gabelstaplers sowie die ID der Palette und den gelesenen Beschleunigungswert.

Um weitergehend die Abbildung der Situation auf die aktuelle Prozess-Instanz zu ermöglichen und ggf. eine Abweichung des Ist-Zustandes vom Soll-Zustand mit MagicMap zu visualisieren, müssen mit Hilfe der Informationen des komplexen Ereignisses Attribute der Objekte in MagicMap geändert werden. Durch Angabe eines JESPA Processing Workflows kann JESPA aus einem komplexen Ereignis eine Attribut-Tabelle erzeugen, welche sich auf die Objektzustände in MagicMap abbilden lässt. Anhand der Attribute eines Objektzustandes kann MagicMap schließlich die Visualisierung der Objekte anpassen, bspw. die grafische Repräsentation von Objekten ändern oder einzelne Objekte in den Fokus der Aufmerksamkeit des Benutzers holen, um diesen bei der Identifikation von Problemen oder Abweichungen vom Build-Time Prozess zu unterstützen.

Um einen Brückenschlag zwischen Objektzuständen und Prozessbeschreibung zu erlangen, ist außerdem eine Komponente vorgesehen, welche die Objektzustände auf das Prozessmodell abbilden kann (vgl. Abbildung 4). Durch diese Abbildung können z.B. Differenzen zwischen dem aktuellen Ist-Zustand und dem zur Build-Time modellierten Prozess aufgedeckt werden. Weitergehend kann das Resultat der Abbildung aber auch eine Auflistung möglicher Prozessschritte sein, die vom aktuellen Prozesszustand aus aktiviert werden können. Diese Folgeschritte können dem Benutzer in MagicMap als mögliche Handlungsoptionen präsentiert werden, wobei zusätzlich die Abhängigkeiten zwischen Objekten und einzelnen Optionen visualisiert werden.

5 Anwendungsbeispiel

Wir betrachten einen Ausschnitt aus dem Transportweg eines Paketes eines Logistikdienstleisters. Ein LCD-Fernseher, als Paket verpackt, wird von einem zentralen Distributionshub zu einem dezentralen Distributionsspoke transportiert. Aufgrund besonderer Anforderungen des Produktes an den Transport (z.B. geringe Erschütterung), kann auf dem Transportweg durch unvorhergesehene externe Einflüsse eine Anpassung der zur Build-Time definierten Prozesse an die konkreten Gegebenheiten nötig werden. In Abbildung 5 werden in der ersten Spalte die Prozesse zur Build-Time ohne die Möglichkeit zur Informationserfassung aus der Umwelt dargestellt, in der zweiten Spalte werden Umwelteinflüsse dargestellt, die diese Prozesse beeinflussen können und in der dritten

Spalte werden die Prozesse zur Build-Time mit der Möglichkeit der kontextbezogenen Informationsgewinnung durch Smart Items abgebildet.

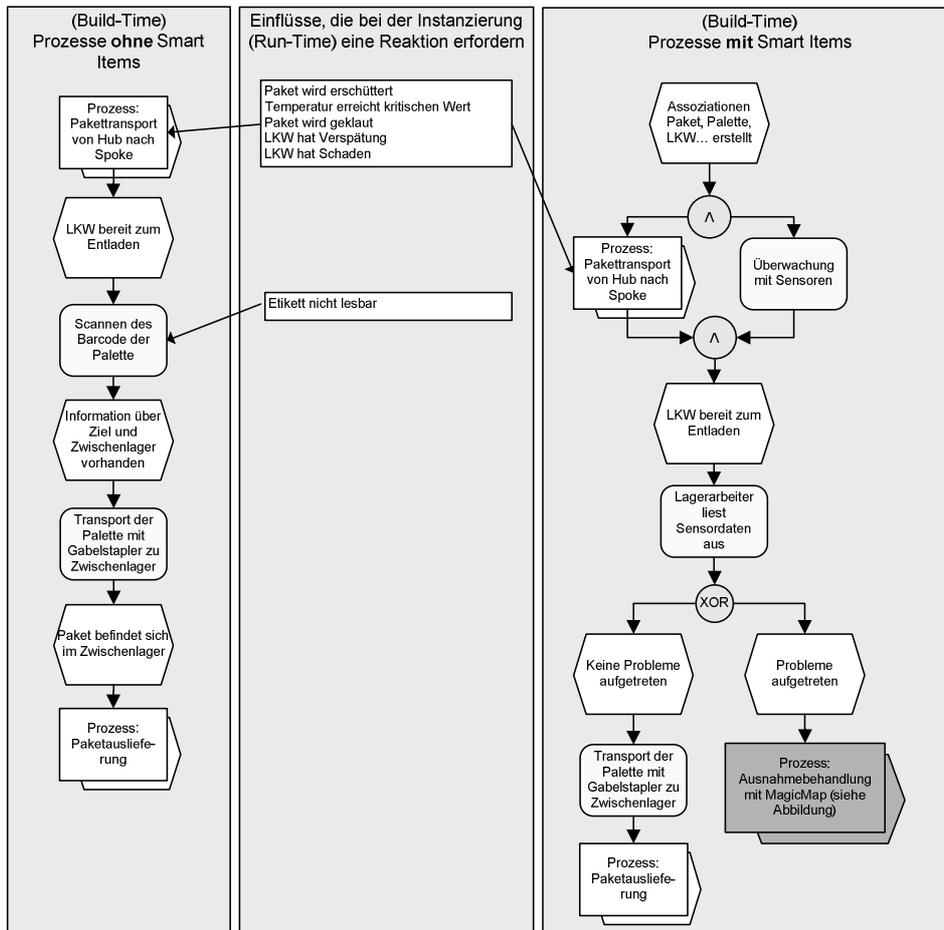


Abbildung 5: Prozesse in einem LogistikszENARIO – links ohne und rechts mit Smart Items

Die halbautomatische Ausnahmebehandlung (in Abbildung 5 als Prozess-Wegweiser rechts unten dargestellt) geschieht mit Hilfe des MagicMap-Clients. Der Screenshot in Abbildung 6 setzt das in Abbildung 5 dargestellte Prozessmodell mit Smart Items fort. Dort werden die Objekte übersichtlich in einer Liste (linker Fensterbereich) und im Detail mit Kontextinformationen auf einer Karte dargestellt (oberer rechter Fensterbereich). Daneben werden Sensorwerte in der Messungs-Ansicht (Fensterbereich unten rechts) angezeigt. Anhand der Visualisierung lässt sich die Abweichung der Prozess-Instanzen zur Run-Time gegenüber den zur Build-Time festgelegten Prozessen schnell und detailliert erfassen und ein genauer Ablauf für die Ausnahmebehandlung zuordnen.

Der in Abbildung 6 dargestellte Screenshot zeigt einen LKW (links) mit drei Behältern im Laderaum, die gerade entladen werden. Ein Lagerarbeiter steht bei den eingegangenen Paketen und nimmt die Lieferung entgegen. Bei der Erfassung der Sensoren mit einem Lesegerät werden die Daten an MagicMap übermittelt. MagicMap erzeugt bzw. aktualisiert entsprechende Objektmodelle für die Visualisierung und übergibt die Daten an den MagicTracker bzw. JESPA, wo sie mit benutzerdefinierten und domänenspezifischen Ereignismustern abgeglichen werden. Wird ein Muster im Strom der Ereignisse identifiziert z.B. das Überschreiten eines Beschleunigungswertes, falls das Paket mit dem LCD-Fernseher während des Transports zu stark erschüttert wurde, werden in den nachfolgenden Verarbeitungsschritten entsprechende Hinweisattribute, z.B. dass ein Objekt besondere Aufmerksamkeit erfordert, in den korrespondierenden MagicMap-Objekten gesetzt. Abhängig von den Attributen eines Objektmodells wird dessen grafische Repräsentation schließlich auf der Karte von MagicMap markiert – hier mit einem Ausrufezeichen. Bereits während des Entladens kann dies berücksichtigt werden, z.B. per Vermerk bei der Quittierung oder Änderung des folgenden Prozessablaufs für diesen Behälter.

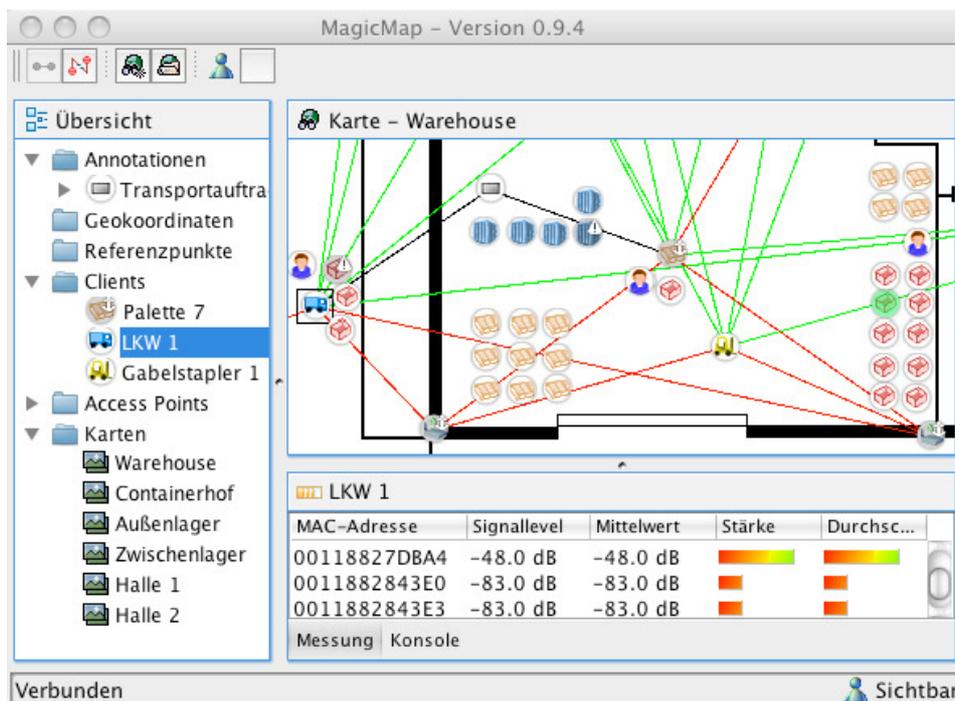


Abbildung 6: MagicMap-Screenshot – räumliche Visualisierung von Smart Items und ihrer Zustände in einem LogistikszENARIO

Darüber hinaus soll eine zur Abholung bereitstehende Palette zum LKW transportiert werden – dies ist durch die Annotation „Transportauftrag“ ersichtlich, die LKW 1 und Palette 7 zugeordnet ist. Außerdem wird ein weiterer Lagerarbeiter, der am Packplatz bei der Palette steht, ein Gabelstapler der sich dem Packplatz nähert, um die Palette zum

LKW zu transportieren, sowie ein Behälter (rechts), der gerade zum Packplatz transportiert wird, dargestellt. Bei einem der Kühlcontainer, die ebenso in der Abbildung dargestellt sind, wurde die Temperatur kurzzeitig überschritten; entsprechende Hinweisattribute wurden im Objektmodell gesetzt und eine Markierung – hier wieder ein Ausrufezeichen – wird angezeigt. Den jeweiligen Objektattributen lassen sich zur Visualisierung frei definierbare Markierungen zuordnen. Bis zu vier solcher Overlays können mit Offset (links oben, rechts oben, links unten, rechts unten) semitransparent über ein Objekticon gelegt werden sowie ein weiteres Farboverlay. Durch die Aggregation der Sensordaten zu Objektzuständen, die räumliche Visualisierung der Objekte und die Kennzeichnung von Objekten, die besondere Aufmerksamkeit erfordern – hier weil ihre Ist-Zustände von den durch die Prozessschritte vorgegebenen Soll-Zuständen abweichen –, wird das vorhandene Kontextwissen für die Nutzerinteraktion aufbereitet, so dass die Akteure des Szenarios geeignet unterstützt werden.

6 Zusammenfassung und Ausblick

In dem vorliegenden Dokument wurden die Möglichkeiten und Herausforderungen der Integration von Smart Items in ereignisgesteuerte Prozessketten behandelt. Die Berücksichtigung des Zustandes solcher Smart Items erlaubt dabei eine frühzeitige Reaktion auf Gegebenheiten der Umwelt, die nicht im Prozessablauf vorgesehen waren. Hierzu wurde ein Lösungsvorschlag vorgestellt, der die Aggregation von Sensordaten, daraus die Ableitung von Objektzuständen und anschließend die Abbildung auf Prozesszustände in nahezu Echtzeit erlaubt. Differenzen zwischen tatsächlichen Zuständen auf Objektebene und erwarteten Zuständen laut Prozessplanung können so aufgedeckt werden. Da solche Abweichungen oft manuelle Eingriffe erfordern, werden Objekte auf einer Karte in ihrem räumlichen Kontext zusammen mit Echtzeit-Zustandsinformationen sowie unerwarteten Abweichungen von der Prozessplanung visualisiert. Dadurch lassen sich die Situation und entsprechende Handlungserfordernisse schnell erfassen, so dass Zustandszuordnungen bzw. Prozessabläufe durch den Anwender zeitnah korrigiert werden können.

Weitere Herausforderungen für die Zukunft sehen wir unter anderem in der Unterstützung und Untersuchung weiterer Prozessbeschreibungssprachen, wie YAWL oder Petri-Netze. Hierzu müssen einheitliche Schnittstellen definiert und weitergehend ausgearbeitet werden. Auch der Einsatz von CEP-Mechanismen muss vereinfacht werden, da diese teils domänenspezifisch sind und somit ein hoher Änderungsaufwand beim Wechsel der Anwendungsbereiche zu erwarten ist. Von besonderem Interesse ist außerdem der Einsatz von Software-Agenten, einem Softwareentwicklungs-Paradigma, das zunehmend Einzug auch in den Logistikbereich hält [Po08]. Agenten könnten einerseits Smart Items, andererseits Prozesse repräsentieren und kooperativ Prozesszustände bestimmen sowie gegebenenfalls autonom Änderungen im Prozessablauf aushandeln.

Literaturverzeichnis

- [Ad06] Adams, M.; Hofstede, A.; Edmond, D.; van der Aalst, W.: Worklets: A service-oriented implementation of dynamic flexibility in workflows, *Lecture Notes in Computer Science* (4275): S. 291, 2006.
- [AH07] Van der Aalst, W.; Ter Hofstede, A.: YAWL: yet another workflow language, *Information Systems* (30:4): S. 245-275, 2005.
- [ARD07] Van der Aalst, W. M. .P.; Rosemann, M.; Dumas, M.: Deadline-based Escalation in Process-Aware Information Systems, *Decision Support Systems* 43(2): S. 492-511, 2007.
- [Ba09] Bade, D.: Towards an Extensible Agent-based Middleware for Sensor Networks and RFID Systems, 3rd International Workshop on Agent Technology for Sensor Networks (AAMAS-09), Budapest, Hungary, May 12, 2009.
- [Br07] Brüning, S.; Zapotoczky, J.; Ibach, P. K.; Stantchev, V.: Cooperative Positioning with MagicMap, *Workshop on Positioning, Navigation and Communication 2007 (WPNC'07)*, Hannover, March 22, 2007.
- [Esp09] Espertech Inc.: Esper – Complex Event Processing, esper.codehaus.org, June 2009.
- [HCK06] Han, J.; Cho, Y.; Kim, E.; Choi, J.: A ubiquitous workflow service framework, *IC-SCSA* (4), *Lecture Notes in Computer Science* (3983), 2006.
- [Ib05] Ibach, P.K., Stantchev, V., Lederer, F., Weiß, A., Herbst, Th., Kunze, T.: WLAN-based Asset Tracking for Warehouse Management, *IADIS International Conference e-Commerce*, Porto, Portugal, Dec. 15-17, 2005.
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK), *Universität des Saarlandes*, Heft 89, 1992.
- [Lu02] Luckham, D.: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison-Wesley Professional, 2002.
- [Ni08] Nicklas, D.; Grossmann, M.; Mínguez, J.; Wieland, M.: Adding High-level Reasoning to Efficient Low-level Context Management: A Hybrid Approach, *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, Hong Kong, March 17-21, 2008.
- [Po08] Pokahr, A.; Braubach, L.; Sudeikat, J.; Renz, W.; Lamersdorf, W.: *Simulation and Implementation of Logistics Systems based on Agent Technology HICL 2008: Logistics Networks and Nodes*, Erich Schmidt Verlag, 2008.
- [RD98] Reichert, M.; Dadam, P.: ADEPTflex - Supporting dynamic changes of workflow without loosing control, *Journal of Intelligent Information Systems (JIIS)*, Special Issue on Workflow and Process Management, 10(2): S. 93-129, 1998.
- [RRF08] Rosemann, M.; Recker, J.; Flender, C.: Contextualisation of business processes, *International Journal of Business Process Integration and Management* 3(1): S. 47-60, 2008.
- [WNL08] Wieland, M.; Nicklas, D.; Leymann, F.: Managing Technical processes using smart workflows, *Towards a service-based Internet*, *Proceedings of the 1st European Conference Service Wave*, S. 287-298, 2008.
- [Wo02] Wong, C.Y.; McFarlane, D.; Zaharudin, A.; Agarwal V.: The Intelligent Product Driven Supply Chain, *IEEE International Conference on Systems, Man and Cybernetics*, 2002.