

Das dynamische Mindesthaltbarkeitsdatum – Auf dem Weg zu einer Echtzeitereignisverarbeitung in der Lebensmittellogistik

¹Falk Alexander, ¹Thomas Hipp, ¹Tobias Scholze,
¹Felix Wagner, ¹Thorsten Schöler, ²Michael Krupp

¹Fakultät für Informatik, ²Fakultät für Wirtschaft
Hochschule für Angewandte Wissenschaften Augsburg
An der Hochschule 1
86161 Augsburg

{falk.alexander, thomas.hipp1, tobias.scholze,
felix.wagner, thorsten.schoeler, michael.krupp}@hs-augsburg.de

Abstract: Das dynamische Mindesthaltbarkeitsdatum stellt neue Herausforderungen an logistische Prozesse und deren IT-Unterstützung. Es wird gezeigt, dass die Ereignisverarbeitung in Echtzeit eine wertvolle Unterstützung in Logistik-IT-Systemen darstellt. Eine Softwarearchitektur zur Überwachung von Logistikprozessen wird vorgestellt und im Anwendungsfall Dynamisches Mindesthaltbarkeitsdatum angewandt.

1 Einleitung

Ein aktuell vieldiskutiertes Problem im Lebensmitteleinzelhandel, ist das massenhafte Wegwerfen von Nahrungsmitteln. Nur ein Bruchteil der erzeugten Lebensmittel landet wirklich auf dem Teller der Konsumenten. In Deutschland werden pro Jahr elf Millionen Tonnen Lebensmittel entlang der Versorgungskette weggeworfen [Ins12]. Dies beginnt bereits beim Erzeuger, mit Produkten, die der Norm und den Anforderungen der Verbraucher nicht standhalten, setzt sich fort in der logistischen Kette zum Handel, wo z. B. Druckstellen zu Aussortierung führen und endet im Kühlschrank der Verbraucher, wo Produkte oft grundlos rigoros entsorgt werden.

Bei Lebensmitteln und verderblichen Produkten, spielt in diesem Zusammenhang das Mindesthaltbarkeitsdatum (MHD) eine wesentliche Rolle. Jedes Lebensmittel trägt dieses Datum, das oftmals ein Richtwert für den Verbraucher ist, ab wann er Lebensmittel wegwirft [Bun12]. Für den Einzelhandel ist es wesentlich, nur Produkte im Regal anzubieten, die noch ausreichend Zeit bis zum Erreichen des MHD haben. Jeder kennt den Reflex, die „jüngere“ und vermeintlich bessere Milch von hinten aus dem Regal zu nehmen, um diese dann doch am gleichen Tag zu trinken.

Das Management der Güterbewegungen berücksichtigt heute bereits das MHD, so ist in

Großhandelslagern bekannt, welche Produkte welches MHD haben. Üblicherweise werden die „älteren“ Produkte zuerst ausgelagert und in den Verkauf gebracht. Allerdings ist nicht bei allen Produkten das MHD wirklich aussagekräftig für die Qualität und die verbleibende Zeit bis zum Verfall. So ist bei verderblichen Produkten, die oft nicht mit Datum gekennzeichnet sind, wie Obst und Gemüse, die Optik für den Verbraucher wichtiger als die angegebene Daten [Ins12]. Hinzu kommen objektive Qualitätskriterien wie bereits genannte Druckstellen oder Geruch. Je nach Reifegrad einer Frucht bleibt bis zum Verkauf einer Ware ein gewisses Zeitfenster. Ist dieses überschritten, kann die Ware nach heutigen Kundenanforderungen nicht mehr verkauft oder nur vergünstigt verkauft werden. Die Länge dieses Zeitfensters ist abhängig von der korrekten Behandlung der Früchte im logistischen Prozess. Das Schlüsselwort heißt hier Kühlung: Wird eine durchgängige Kühlung bei der richtigen Temperatur erreicht, dann spricht der Logistiker von der geschlossenen Kühlkette. Erschwert wird dies durch unterschiedliche Temperatur-Anforderungen. So benötigen Schokolade ca. 15 °C, Äpfel ca. 4 °C und Südfrüchte ca. 12 °C. In Lagerhallen werden meist nicht ausreichend unterschiedliche Temperaturzonen eingerichtet. Resultat ist, dass nicht alle Produkte bei optimaler Temperatur gelagert werden können. Da lautet die Devise: Geschwindigkeit, also Produkte möglichst schnell zu distribuieren und möglichst kurz suboptimal zu kühlen.

Die Verfolgung der Kühlung mittels RFID wird seit langem diskutiert. RFID-Transponder werden hierzu mit Thermometern kombiniert. Jederzeit kann ausgelesen werden, welcher Temperatur das jeweilige Produkt entlang der logistischen Kette ausgesetzt war. Das Konzept des dynamischen MHD geht einen Schritt weiter: Ausgehend von der maximalen Haltbarkeit bei optimaler Kühlung, wird die Haltbarkeit reduziert, wenn in der logistischen Kette ungünstige Verhältnisse auf das Produkt eingewirkt haben. Das heißt, Temperaturüberschreitungen werden identifiziert, ausgewertet und in der Berechnung der restlichen Lebenszeit des Produktes berücksichtigt. Dies führt dazu, dass eventuell Palette A, die zwar später in ein Lager eingelagert wurde als Palette B, aber ungünstigere Rahmenbedingungen ausgesetzt war, als erste in die Verkaufsräume geht, oder gleich als B-Ware verkauft wird, um die Entsorgung zu vermeiden. Bei der aktuellen Entwicklung der Sensoren ist dieses Konzept auch in Kombination mit anderen Verfallsindikatoren, wie Ethylengasen (Reifegas bei Fruchtreife) oder auch Lichteinfall o. ä., denkbar [Fra12].

Die weiteren Abschnitte sind wie folgt gegliedert: Zunächst wird im Abschnitt 2 der Stand der Technik in der IT-Unterstützung von Lebensmittellogistikketten, speziell unter dem Gesichtspunkt RFID und Ereignisverarbeitung in Echtzeit, kurz dargestellt. Anschließend werden in Abschnitt 3 einige Anwendungsfälle präsentiert, welche in Abschnitt 4 im Rahmen eines Hardware-/Software-Co-Systems umgesetzt werden. Abschließend werden die mit der Umsetzung gesammelten Erfahrungen dargestellt (Abschnitt 5) und in Abschnitt 6 eine Zusammenfassung mit Ausblick auf nachfolgende Forschungsaktivitäten gegeben.

2 Stand der Technik aus IT-Sicht

Der Einsatz von Ereignisverarbeitungstechnologien, wie z. B. Complex Event Processing (CEP), in Logistikanwendungen und dem Internet der Dinge ist nicht neu, wie beispiels-

weise in [SSMW10] beschrieben. Ebenso wenig wie der Einsatz von RFID in der Lebensmittellogistik bereits gut untersucht ist (siehe auch [GSC⁺07] und [GSBB08]). Allerdings erscheint der in diesem Beitrag vorgestellte Ansatz, RFID mit CEP zu einem Echtzeitsystem für die Lebensmittellogistik zu kombinieren, erfolversprechend, wie es die Studie ähnlich gelagerter Untersuchungen zeigt.

Einen Überblick über den Einsatz von RFID-Technologie in der Lebensmittel-Supply-Chain gibt [Jon06]. Am Beispiel der Fleischproduktion in Japan wird der Einsatz von RFID deutlich und der Informationsfluss über die einzelnen Teilnehmer der Logistikkette hinweg verdeutlicht. [GMRS07] beschreibt ebenfalls einen RFID-basierten Ansatz für die Lebensmittellogistik (in diesem Fall Früchte), allerdings mehr auf Funk- und Hardwareaspekte fokussierend.

In [HCC06] wird eine Softwarearchitektur beschrieben die dem hier vorgestellten Ansatz ähnlich ist. Es werden Informationen von RFID-Tags mittels Datenfusion ausgewertet. Im Gegensatz zum hier beschriebenen Ansatz wird ein regelbasiertes System mit Fuzzyfizierung verwendet, ein Ansatz der dem Fuzzy Classifier System unter [SMS05] nicht unähnlich ist.

Der Begriff eines „Real-Time Food Receiving operations management Systems (RFRS)“ aus [LCTP10] beschreibt den eingangs vorgestellten Anwendungsfall. Im Gegensatz zu den in [LCTP10] eingesetzten Case-Base-Reasoning- und Methoden der Fuzzy-Logik [HCC06], sowie dem Vorschlag aus [LCTP10], verwendet der hier im Anschluss vorgestellte Ansatz Standard-CEP-Komponenten zur Ereignisverarbeitung und steht somit den Untersuchungen in [HHX09] näher. Die beschreibungslogische Modellierung der Ereignisse in einer Ontologie, wie in [HHX09] vorgeschlagen, soll in Zukunft ebenfalls untersucht werden, wie es im Abschnitt 6 kurz beschrieben wird.

Um das Szenario Dynamisches MHD als Anwendungsfall in der IT-Infrastruktur eines Unternehmens umsetzen zu können, sind Softwarearchitekturen notwendig, die schnell und dynamisch – am besten in Echtzeit – auf aktuelle Gegebenheiten im Logistikprozess reagieren können. Bei der Lebensmittellogistik darf eben nicht erst in einem nachts laufenden Batch-Prozess festgestellt werden, dass auf Palette A schnell verderblicher Fisch gelagert ist, der am Vortag noch verkauft hätte werden können.

Um zu einer tragfähigen IT-Architektur zu gelangen, sollen nun einige exemplarische Anwendungsfälle analysiert werden.

3 Anwendungsfälle

Aus dem Logistik-Anwendungsfall Dynamisches MHD lassen sich Anwendungsfälle (Use cases) ableiten, aus denen wiederum die Anforderungen an eine Softwarearchitektur zur Logistikunterstützung abgeleitet werden können. Im Folgenden sollen exemplarisch Anwendungsfälle vorgestellt werden, die direkten Einfluss auf die in Abschnitt 4 vorgestellte Softwarearchitektur haben.

Warenfluss im Umschlagsplatz: Waren, bestehend aus beispielsweise verschiedenen Früchten und Gemüse, erreichen einen Umschlagplatz. Beim Einlagern der Waren werden diese auf Vollständigkeit geprüft. Dies geschieht beispielsweise durch einen Abgleich der Liste der ankommenden EPCs¹ und der vorher eingereichten Liste durch den Erntebetrieb. Zusätzlich werden die Messdaten über den bisherigen Kühlstatus ausgelesen. Ist die Kühlung nicht durchgehend optimal gewesen, so verringert sich automatisch das MHD. Die Waren müssen früher ausgelagert werden, damit sie beim Endverbraucher das gleiche MHD besitzen. Die Transportwege müssen kurzfristig adaptiv sein. Automatisierte Transportsysteme, z. B. Rollbänder, holen bevorzugt Waren mit geringer MHD. Personell besetzte Lieferanten wie Gabelstaplerfahrer nutzen mobile Endgeräte z. B. Tablets statt Klemmbretter. Deren Transportlisten werden durch ereignisorientierte Systeme aktualisiert.

Dynamische Warenlisten: Der Güterverkehr zwischen den Lagerplätzen ist vielfältig. Jedoch werden Waren immer in Sendungen gruppiert. Basierend auf der verbleibenden Haltbarkeit, können diese optimiert zusammengestellt werden. Die Geschwindigkeit der Transportwege unterscheidet sich. So können Waren mit einem geringen MHD einen schnelleren Transportweg bekommen. Sind Informationen über den Zustand einer Lieferung bekannt, während sie unterwegs ist, so können Verzögerungen kompensiert werden. Beispiel: Die Tomaten aus der Lieferung A erreichen nicht rechtzeitig eine Anschlusslieferung B. Durch dieses Ereignis lassen sich Transportlisten schnell anpassen um Lieferkapazitäten bestmöglich auszunutzen.

Historie: Im System wird eine Historie aller eingegangenen Waren protokolliert. Der Warenfluss ist sowohl für Unternehmer als auch Verbraucher transparent. Fehlerfälle können leichter identifiziert werden. Fehler können im Transportwesen entstehen: Eine Kiste Früchte wurde versehentlich zum Gemüse gestellt und bleibt vorerst unbemerkt, da die Kühlung annähernd ähnlich ist. Wird jedoch die vermeintliche Gemüsebox zu einer Sendung Früchte gestellt, schlägt das System Alarm. Die Kombination von Früchten in diesem Gemüsetransport ist ein Fehlerfall. Die Früchte sind nicht auf der Sendungsliste eingetragen. Dieser Fehlerfall kann schon beim Verlassen des Kühlraums erkannt werden. Der Abholer hat keine Berechtigung die Früchte aus dem Kühlraum zu entnehmen. Basierend auf den Anforderungen aus den dargestellten Anwendungsfällen (Use Cases) wurde die folgende Hard-/Software-Architektur prototypisch umgesetzt.

4 Technischer Aufbau

Der Aufbau in Abbildung 1 lässt sich, analog der in [BD10] vorgestellten Systeme, in drei Schichten unterteilen, wie in den folgenden Abschnitten dargestellt.

¹Elektronische Produktcode, international verwendete Identifikationsnummer für Produkte

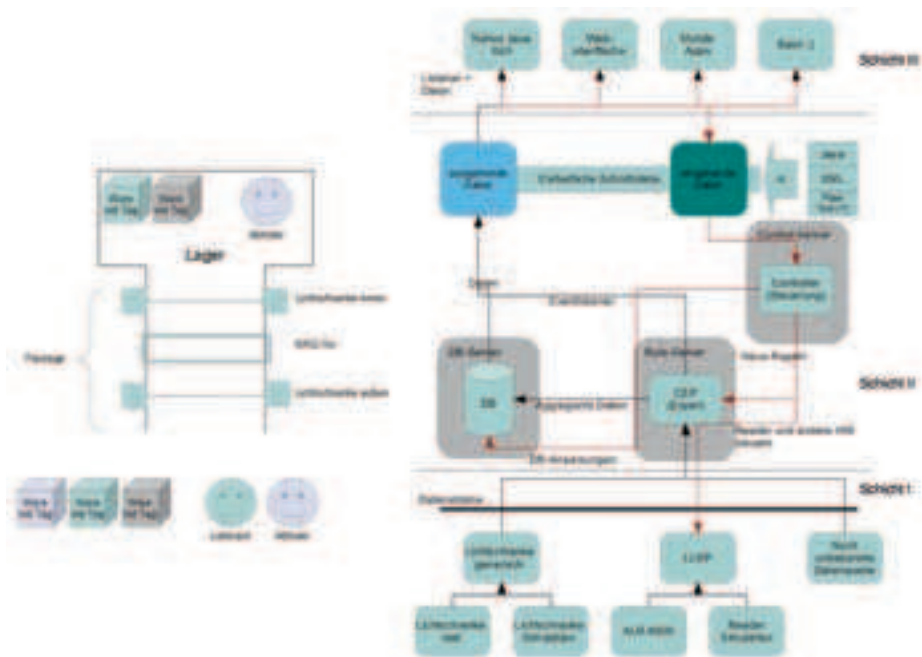


Abbildung 1: Softwarearchitektur der Ereignisverarbeitung

4.1 Hardwareschicht (Schicht I)

In dieser Schicht werden alle Sensoren verwaltet, die im Gesamtsystem Daten generieren. Jeder einzelne Sensor übermittelt seine Messungen, dies können von Temperaturmessungen bis hin zur Position eines Objekts alle denkbaren Daten sein, die für ein Objekt von Bedeutung sind. Ein Sensor ist nicht zwangsläufig ein klassischer Sensor für Temperatur, Luftdruck oder Feuchtigkeit, sondern kann auch ein RFID-Leser, Überwacher einer Datenbank, QR- oder Barcodescanner sein. Manuelle Datenerfassungen im logistischen Prozess werden ebenfalls als Sensoreingaben betrachtet. Die gesammelten Daten werden mit einem Messaging-Dienst an die Schicht II weitergegeben. Dieser Messaging-Dienst erlaubt es den Sensoren, vollständig verteilt zu sein. Weiterhin übernimmt er die Sicherung des Kommunikationsverkehrs und die Bewältigung der Datenmengen. Zum Einsatz kam das Messaging-System HornetQ² aus dem Open-Source-JBoss-Projekt. Dieses Messaging-System übernimmt die Nachrichtenzustellung zwischen allen Schichten und Komponenten der Softwarearchitektur. In der Hardwareschicht liegt das Hauptaugenmerk auf Verlässlichkeit und Durchsatz der Ereignisverarbeitung.

Konkret verwendet das System einen RFID-Multi-Antennen-Reader von Alien Technology (ALR-9900), welcher sogenanntes Bulk Reading, also das gleichzeitige Erfassen einer großen Menge RFID-Tags, wie sie üblicherweise bei der Erfassung logistischer Gebinde auf Paletten verwendet werden, unterstützt. Als weitere Informationsquelle im System (Datenfusion) wird eine Lichtschranke verwendet, durch welche die Bewegungsrichtung der Palette bei einer RFID-Tor-Durchfahrt erkannt werden kann (Ein- oder Auslagerung).

²<http://hornetq.org/>



Abbildung 2: Entwurf einer leichtverständlichen Ansicht und Rich-Client-Gui mit Java Swing

4.2 Verarbeitungsschicht (Schicht II)

Um aus der reinen Datenflut auf niedriger Ereignisebene Informationen zu extrahieren, werden eingehende Nachrichten der Sensoren an die integrierte CEP-Engine übertragen. Für die Verarbeitung der Ereignisse (z. B. Datenfusion und -aggregation) in der CEP-Engine wurden Regeln deklariert, die den verschiedenen logistischen Anwendungsfällen entsprechen. Die reinen Daten (Ereignisse auf niederster Ebene) sowie das Zutreffen/-Feuern von Regeln wird in der Datenbank persistiert. Über diesen Weg können – CEP-typisch – aus niederen einzelnen Ereignissen, höherwertige Informationen (komplexe Ereignisse nach Luckham [Luc07]) abgeleitet werden.

Zur Anbindung an die verwendeten In-Memory-Datenbank wurde der von JBoss entwickelte OR-Mapper Hibernate³ verwendet. Ein OR-Mapper, kurz für Object-Relational-Mapper, ermöglicht die Abbildung von Datenstrukturen auf POJOs⁴. Somit ist eine Datenbankinteraktion aus Entwicklersicht ein Erstellen und Bearbeiten von Objekten. Lasttests der Persistenzarchitektur ergaben, dass die Persistierung von 10 000 Datensätzen zu je fünf atomaren Spalten mittels dieses Ansatzes auf einem Standard-PC weniger als zehn Sekunden dauerte.

4.3 Anzeigeschicht (Schicht III)

Die aus Schicht II abgeleiteten Informationen werden auf lesbare Art bereitgestellt (siehe Abbildung 2). Die einfachste Möglichkeit ist das Anzeigen des aktuellen Zustandes eines Logistikprozesses oder eines Objekts (beispielsweise einer Palette mit frischen Früchten) z. B. bezüglich Ort, Ziel und Prozessabschnitt. Dem Benutzer kann eine Meldung zugestellt werden, die sein Eingreifen erfordert oder ihn auf eine mögliche Fehlfunktion hinweist. Mögliche Meldungen sind u. a. Pop-Up-Fenster an seinem Arbeitsplatzrechner, Nachrichten auf sein Smartphone oder mobiles Gerät bzw. Warnsignale z. B. das Auslösen einer Warnleuchte mit Sirene.

Konkret wurden verschiedene gleichberechtigten Benutzeroberflächen umgesetzt und mit-

³<http://www.hibernate.org>

⁴Plain Old Java Objects



Abbildung 3: Ansicht auf mobilen Endgeräten

tels Messaging-System an die Verarbeitungsschicht gekoppelt: Eine native Rich-Client-GUI auf Basis von Java Swing (siehe Abbildung 2), sowie eine webbasierte Oberfläche welche mittels *Responsive Design* das enthaltene Layout je nach Größe des Bildschirms optimal anpasst⁵. Somit ist diese Web-Oberfläche von Desktop-Arbeitsplätzen sowie auch von mobilen Endgeräten wie Tablets oder Smartphones verwendbar, wie es Abbildung 3 zeigt. Beide Benutzeroberflächen ermöglichen die Ausgabe der aktuellen Ereignissituation sowie die Konfiguration und das Management des Ereignisverarbeitungssystems. Dies bedeutet, dass mehrere Clients gleichzeitig in der Lage sind, sich auf den Datenstrommanagementserver zu verbinden und dessen Informationen grafisch darzustellen. Als Anwendungsfall sind verteilte und einfach gehaltenen Kontrollbildschirmansichten für Lagerarbeiter und – parallel dazu – eine Anzeige einer komplexen Oberfläche für den Vorarbeiter denkbar.

5 Anwendung und Erfahrungen

Der skizzierte Hardware/Software-Aufbau ermöglicht die Umsetzung des in Abschnitt 1 beschriebenen Szenarios Dynamisches MHD. Waren können in das angeschlossene IT-System ein- und ausgelagert werden und die hierbei erzeugten Ereignisströme können in Echtzeit analysiert, überwacht und gemanagt werden. Die erstellte Infrastruktur ist flexibel und leicht an weitere logistische Anwendungen anpassbar.

Um nun sicherzustellen, dass sich im Lager keine Waren mit kritischem MHD befinden, müssen alle Prozesse, die den Warenfluss betreffen, überwacht werden. Durch diese Überwachung werden Prozesse transparenter und können analysiert werden. Jeder Schritt, den eine Ware im Prozess durchläuft, muss protokolliert werden. Dadurch werden die Daten für die Analyse gewonnen.

Durch die Aggregation der Ereignisdaten des Warenflusses mit weiteren Daten werden zusätzliche Informationen gewonnen. Lädt beispielsweise ein LKW Waren ab, so ist im

⁵Die Weboberfläche arbeitete im Hintergrund mit Tapestry (<http://tapestry.apache.org/>). Als Hilfe zur Erstellung der Oberfläche wurde das von Twitter stammende Bootstrap-CSS-Framework (<http://twitter.github.com/bootstrap/>) verwendet.

Prozess die Information bekannt, dass die Waren im System existieren. Der LKW identifiziert sich beim System und somit kann verifiziert werden, dass die Waren durch den korrekten Dienstleister, an der passenden Verladestelle zur gewünschten Zeit unter Berücksichtigung des MHDs angeliefert wurden. Das Datenaufkommen lässt sich auf Grund der gut skalierenden Architektur nahezu beliebig erhöhen.

Produzierende oder weiterverarbeitende Betriebe fügen den Waren Informationen über ihre weiteren Verarbeitungsprozesse hinzu wie es beispielsweise bei der Just-In-Sequence-Produktion der Fall ist. Komplexere Produkte besitzen eine Vielzahl von Daten, bestimmt durch die Komponenten, aus denen sie zusammengesetzt sind. Diese zusätzlichen Informationen können ebenfalls auf beschreibbaren RFID-Tags untergebracht werden. Die in Echtzeit erfassten Primäreignisse werden vorverarbeitet, aggregiert und mit weiteren Daten angereichert. Es entstehen, online und ebenfalls in Echtzeit, komplexe Ereignisse die tiefgreifendere Rückschlüsse auf den Warenfluss erst ermöglichen.

Ein Ereignis wird aus dem Zusammenspiel verschiedener Informationen erzeugt. An einem Warenumschlagplatz trifft eine Lieferung mit verschiedenen Früchten ein. Alle Früchte wurden auf ihrem Transportweg einheitlich gekühlt. Dadurch ist die optimale Lagertemperatur, von beispielsweise Südfrüchten, nicht eingehalten worden. Ihr MHD ist somit reduziert. Diese Information löst eine Benachrichtigung an das Smartphone eines Mitarbeiters aus, dass die Früchte schneller in eine geeignete Kühlzone gebracht werden müssen. Im zukünftigen Warenfluss werden diese mit einer höheren Priorität versendet. Die kontinuierliche Echtzeit-Überwachung des MHD ermöglicht Rückschlüsse auf die Güte der Transportkette. Wurden Waren in der Vergangenheit falsch verladen oder unzureichend auf ihrem Transportweg gekühlt, kann dies automatisch als Ereignis aus dem Gesamtsystem hervorgehen. Jedem Akteur in diesem System werden andere Informationen bereitgestellt. Einen Lagerist betreffen kurzfristige Planänderungen. In der administrativen Ebene ist es wichtiger zu wissen, dass der Einzelhandel benachrichtigt werden muss, dass Südfrüchte mit verringertem MHD eintreffen.

Für die gegebenen Anwendungsfälle der Logistik hat sich eine Umsetzung auf Basis von CEP angeboten. Nicht zuletzt die voneinander unabhängigen Datenquellen RFID-Reader und Lichtschranke machten eine Datenfusion (Verknüpfung der Ereignisse) nötig. Auch der hohen Frequenz der Ereignisse musste mit einer entsprechenden Softwarearchitektur Rechnung getragen werden. Nach einer Evaluation verschiedener CEP-Engines (z. B. JBoss Drools und Esper) und der Gegenüberstellung mit den Anforderungen an eine solche Komponente, kam Esper zu Einsatz. Nicht zuletzt, da Esper einen einfacheren Einstieg sowie eine umfassende Dokumentation bietet.

Um die Einfachheit der Beschreibung von Überwachungsregeln mit Esper zu belegen, sollen im folgenden einige Regelfragmente exemplarisch für das System vorgestellt werden.

```
SELECT * FROM TagEvent
WHERE (id == '1234 5678' AND temp > 3.0 AND temp < 5.0)
OR (id == '2468 3579' AND temp > 11.0 AND temp < 13.0)
```

Obige Regel informiert darüber, ob die jeweiligen RFID-Tags ihren gültigen Temperaturbereich über- bzw. unterschreiten. Für die skizzierten Anwendungsfälle, war es zusätzlich notwendig, kausale Abfolgen von Ereignissen zu definieren. Dies ließ sich mit den ent-

sprechenden Ausdrucksmöglichkeiten der Esper-EPL gut umsetzen.

```
SELECT * FROM pattern [every SensorEvent(triggered, name='A') ->
  SensorEvent(NOT triggered, name='A') -> middleevent=TagEvent
  until SensorEvent(triggered, name='B') -> SensorEvent(NOT
  triggered, name='B')]
```

Erfasst werden alle RFID-Tags zwischen dem Ereignis A, dem Überschreiten der Kühltemperatur und dem abschließendem Ereignis B, dem Erreichen der optimalen Temperatur.

6 Fazit und Ausblick

Der Anwendungsfall Dynamisches MHD zeigt aktuelle Herausforderungen in lebensmittellogistischen Prozessen. Durch die vorgestellte Softwarearchitektur und den Einsatz von CEP steht eine leistungsfähige Möglichkeit zur Verarbeitung der anfallenden Daten zur Verfügung, die auch mit hohen Datenvolumen zurecht kommt, wie sie beispielsweise beim sogenannten Bulk Reading von ganzen Paletten RFID-ertüchtigter Waren die Regel sind. Die moderne multimodale Responsive-Design-Web-Oberfläche ermöglicht eine übersichtliche Situationsbewertung sowie ein transparentes Management des gesamten IT-Systems.

Zukünftige Arbeiten auf Basis der vorgestellten Hardware und Software umfassen eine Verteilung der Systemfunktionalitäten auf Softwareagenten des objektfunktionalen SCALA-Multiagentensystem SMAS⁶ sowie eine 3D-Visualisierung im parallel dazu entwickeltem 3D-SCADA-System für industrielle Anwendungen [Bor12]. Durch die bereits umgesetzten SMAS-Agenten-Plug-Ins für semantische Datenverarbeitung auf Basis von OWL-Ontologien, soll die Modellierung der logistischen Prozesse sowie die Überwachung dieser Prozesse auf Basis von Beschreibungslogik umgesetzt werden. Die Umsetzung des Daten- und Ereignismodells einer industriellen Fertigungsanlage stellen eine sinnvolle Ausgangsbasis für die beschreibungslogische Modellierung von Logistikprozessen und deren Ereignissen dar.

Danksagung Wir bedanken uns bei dem Team der Abteilung für Supply Chain Technologies des Fraunhofer-Instituts für Integrierte Schaltungen IIS, welches uns im Rahmen dieser Arbeit zugrundeliegenden Studienprojektes tatkräftig mit der notwendigen technischen Ausstattung unterstützt hat und uns auch über das Projekt hinaus bei der Erstellung dieser Arbeit beratend zur Seite stand.

Literatur

[BD10] Ralf Bruns und Jürgen Dunkel. *Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse*. Springer, Berlin, 1. Auflage, Mai 2010.

⁶<https://github.com/scala-multi-agent-system> sowie [Lie12]

- [Bor12] Benjamin Bortfeldt. *Verarbeitung und Visualisierung von Ereignissen für Industrieanlagen*. Masterarbeit, Hochschule für Angewandte Wissenschaften Augsburg, Augsburg, (noch nicht erschienen), 2012.
- [Bun12] Bundesministerium für Ernährung, Landwirtschaft und Verbraucherschutz. Das Mindesthaltbarkeitsdatum ist kein Verfallsdatum, 2012.
- [Fra12] Fraunhofer-Gesellschaft, Food Chain Management. Gassensorik für die Food Chain. http://www.fcm.fraunhofer.de/de/beispiele1/kompakte_messsystemefuerlebensmittelschnelltests.html, 2012.
- [GMRS07] F. Gandino, B. Montrucchio, M. Rebaudengo und E.R. Sanchez. Analysis of an RFID-based Information System for Tracking and Tracing in an Agri-Food chain. In *RFID Eurasia, 2007 1st Annual*, Seiten 1–6, sept. 2007.
- [GSBB08] Pablo Ezequiel Guerrero, Kai Sachs, Stephan Butterweck und Alejandro P. Buchmann. *Performance Evaluation of Embedded ECA Rule Engines: A Case Study*. 2008.
- [GSC⁺07] Pablo Ezequiel Guerrero, Kai Sachs, Mariano Cilia, Christof Bornhövd und Alejandro P. Buchmann. Pushing Business Data Processing Towards the Periphery. In Rada Chirkova, Asuman Dogac, M. Tamer Özsu und Timos K. Sellis, Hrsg., *ICDE*, Seiten 1485–1486. IEEE, 2007.
- [HCC06] Han-Pang Huang, Ching-Shun Chen und Tien-Ying Chen. Mobile Diagnosis based on RFID for Food Safety. In *Automation Science and Engineering, 2006. CASE '06. IEEE International Conference on*, Seiten 357–362, oct. 2006.
- [HHX09] Zhu Huaji, Wu Huarui und Sun Xiang. Research on the Ontology-Based Complex Event Processing Engine of RFID Technology for Agricultural Products. In *Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on*, Jgg. 1, Seiten 328–333, nov. 2009.
- [Ins12] Institut für Siedlungswasserbau, Wassergüte- und Abfallwirtschaft. Ermittlung der weggeworfenen Lebensmittelmengen und Vorschläge zur Verminderung der Wegwerfrate bei Lebensmitteln in Deutschland – Kurzfassung. http://www.bmelv.de/SharedDocs/Downloads/Ernaehrung/WvL/Studie_Lebensmittelabfaelle_Kurzfassung.html, 2012.
- [Jon06] P. Jones. Networked RFID for use in the Food Chain. In *Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on*, Seiten 1119–1124, sept. 2006.
- [LCTP10] S.I. Lao, K.L. Choy, Y.C. Tsim und T.C. Poon. A RFID-based decision support system for food receiving operations assignment. In *Supply Chain Management and Information Systems (SCMIS), 2010 8th International Conference on*, Seiten 1–7, oct. 2010.
- [Lie12] Rico Lieback. *Objekt-funktionale Plattform für Cyber-Physical Systems*. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Augsburg, Augsburg, 2012.
- [Luc07] David Luckham. *The power of events: an introduction to complex event processing in distributed enterprise systems*. Addison Wesley, 2007.
- [SMS05] Thorsten Schöler und Christian Müller-Schloer. First steps towards organic computing systems: monitoring an adaptive protocol stack with a fuzzy classifier system. In *Proceedings of the 2nd conference on Computing frontiers*, CF '05, Seiten 10–20, New York, NY, USA, 2005. ACM.
- [SSMW10] C. Seel, J. Schimmelpennig, D. Mayer und P. Walter. Conceptual modeling of complex events of the Internet of Things. In *eChallenges, 2010*, Seiten 1–8, oct. 2010.