

# Prozessorientierte Überwachung in der Produktion

Daniela Fisseler<sup>1</sup>, René Reiners<sup>2</sup>

**Abstract:** Modelle im Industrie 4.0-Kontext sind ein wichtiges Hilfsmittel um Komplexität bei Produktionssystemen unter Kontrolle zu bekommen und sollen in der Betriebsphase verwendet werden um beispielsweise im Voraus mögliche Probleme zu erkennen. In dieser Arbeit stellen wir einen modellbasierten Ansatz zur Überwachung von Produktionsprozessen vor, bei dem Sensornetzwerke mit einem ausführbaren Modell der Produktion verknüpft werden. Ziel ist es, eine detaillierte Analyse auf Prozessschrittebene zu ermöglichen, etwa von Energieverbräuchen. Der vorgestellte Ansatz wird mit einem Beispiel in der Automobilproduktion verifiziert. Aus dem Modell zur Überwachung soll in einem nächsten Schritt ein Modell zur Simulation der Produktion erstellt werden.

**Keywords:** BPMN, Sensornetzwerk, Simulation, Geschäftsprozesse, Industrie 4.0

## 1 Einleitung

In Industrieunternehmen ist es häufig nicht möglich zu bestimmen, wie viele Ressourcen, in unserem Fall Energie, in einer einzelnen Station für einen einzelnen Artikel verbraucht wurden. Häufig sind die Sollwerte der Maschinenproduzenten für die einzelnen Maschinen bekannt sowie der Gesamtenergieverbrauch der Fabrik. Mit diesen Werten kann man aber nur den durchschnittlichen Energieverbrauch pro hergestelltem Artikel berechnen. Dabei ist eine prozessschritt- und stationsgenaue Datenerfassung und Auswertung aus verschiedenen Gesichtspunkten interessant. Eine genaue Datenerfassung durch Sensoren ermöglicht etwa eine vorausschauende Wartung, welche zu weniger und kürzeren Ausfällen in der Produktion führen kann. Außerdem werden mit diesen Daten neue Analysen ermöglicht: der Vergleich von verschiedenen Produkten oder verschiedene Produktionsstätten für gleiche Produkte. Die ermittelten Daten können dann zur Verbesserung der Ressourceneffizienz verwendet werden und dadurch für Firmen und Kunden monetäre und ökologische Vorteile bringen.

Industrie 4.0 beinhaltet die technische Integration von Cyber-Physischen-Systemen in die Produktion [Um13]. Sensornetzwerke und die Verarbeitung ihrer Daten spielen hierbei eine große Rolle. Außerdem sind Modelle in den Industrie 4.0-Konzepten ein wichtiges Instrument, um die Komplexität bei Produktionssystemen besser handhabbar zu machen. Es wird in der Regel zwischen Planungs- und Erklärungsmodellen

---

<sup>1</sup> Fraunhofer für angewandte Informationstechnik FIT, User-Centered Ubiquitous Computing, Schloss Birlinghoven, 53754 Sankt Augustin, daniela.fisseler@fit.fraunhofer.de

<sup>2</sup> Fraunhofer für angewandte Informationstechnik FIT, User-Centered Ubiquitous Computing, Schloss Birlinghoven, 53754 Sankt Augustin, rene.reiners@fit.fraunhofer.de

unterschieden. Erstere werden in der Planungsphase verwendet, um zu erklären, wie etwas umgesetzt werden soll. Erklärungsmodelle hingegen werden verwendet, um existierende Systeme zu erklären und Wissen über die Systeme zu erhalten, etwa durch Simulation, oder auch um Designentscheidungen zu verifizieren. Zukünftig sollen Erklärungsmodelle auch insbesondere in der Betriebsphase verwendet werden, um etwa „den regulären Betriebszustand zu überwachen, Verschleiß ohne Produktionsunterbrechung zu erkennen oder Komponentenausfälle und Störungen vorherzusagen“ [Um13].

Diese Industrie 4.0 Konzepte können genutzt werden, um eine prozessorientierte Überwachung umzusetzen. Diese Arbeit beschreibt einen Ansatz zur Modellierung von Produktionsprozessen und die erstellten Modelle mit Sensornetzwerken in Produktionsumgebungen zu kombinieren. Die Ausführung dieser Modelle erfolgt mit Hilfe einer Process Engine. Hierdurch wird eine prozessorientierte Überwachung der Produktion anhand von Sensorwerten und prozessrelevanten Ereignissen erreicht. So können Energieverbräuche zeitnah den Prozessschritten oder Subprozessen zugeordnet werden - eine nachgelagerte Zuordnung entfällt somit. Weiterhin können maschinelle Ausfälle vorausgesagt und frühzeitig Maßnahmen zur Behebung von Störungen eingeleitet werden. Durch die Verwendung einer graphischen Modellierungssprache werden die Prozessmodelle insgesamt leichter verständlich.

Ein vielversprechender Ansatz für die Modellierung ist Business Process Modelling and Notation (BPMN) 2.0., da es eine in der Industrie verbreitete, standardisierte, graphische Notation ist um Geschäftsmodelle darzustellen. Ein Ziel von BPMN ist die Bereitstellung einer Notation, die für Personen mit unterschiedlichen Rollen in einem Unternehmen einfach verständlich ist. Somit kann sie als Kommunikationsmittel eingesetzt werden, etwa zwischen technischen Entwicklern und Business Analysten. Der Standard definiert außerdem die XML Serialisierung von BPMN Modellen. Überdies hinaus gibt BPMN anderen XML-basierten Sprachen, wie WSBPEL, die für die Ausführung von Geschäftsprozessen konzipiert wurden, die Möglichkeit für eine graphische, geschäftsorientierte Visualisierung [Bu11].

Mit Hilfe dieses BPMN-Modells zur Überwachung wird ein Modell für die Simulation der Produktion erstellt. Dies findet Anwendung bei Tests oder Demonstrationen oder aber um Änderungen in der Produktion zu simulieren. Andere BPMN-Tools bieten zwar auch Simulationsmöglichkeiten für Prozesse, aber dort werden nur die Nachrichten simuliert, für die explizit Events definiert worden sind. Die Benutzung wäre aufwendig und unübersichtlich, da dann für jeden Sensor ein Event im Prozessmodell definiert werden muss, was in dem hier beschriebenen Ansatz nicht nötig ist. Eine Alternative wäre ein Simulator für Sensoren, der MQTT Nachrichten verschickt, jedoch müssten hierfür nochmals alle Sensoren neu modelliert bzw. konfiguriert werden.

Der restliche Aufbau des Artikels ist wie folgt: In Kapitel 2 werden verwandte Arbeiten vorgestellt; in Kapitel 3 wird erklärt, wie die Geschäftsprozessmodelle zur Überwachungen erstellt werden können inklusive einer Machbarkeitsstudie; Kapitel 4

erläutert beispielhaft, wie die Umwandlung in ein Simulationsmodell aussehen kann und Kapitel 5 beschreibt die nächsten forschungsrelevanten Schritte für den Ausbau des vorgestellten Ansatzes.

## 2 Verwandte Arbeiten

Bei der prozessorientierten Überwachung werden Sensornetzwerke mit der Ausführung von Geschäftsprozessen mit Hilfe von Ereignissen kombiniert. Es gibt Ansätze um Complex Event Processing und Prozess Management miteinander zu integrieren zu Event Driven Business Process Management. In [Am08] wird ein allgemeines Framework beschrieben, um dies zu erreichen. Die beiden unterschiedlichen Plattformen kommunizieren über Ereignisse, die entweder durch das IT-System ausgelöst werden oder die Prozess Engine. Um mögliche Probleme zu entdecken, die die korrekte Ausführung des Prozesses gefährden, werden Ereignis-Patterns definiert und die Ereignisse auf Übereinstimmung mit diesem geprüft. In [EH12] wird BPMN2.0 zum Modellieren der Hauptproduktionsprozesse verwendet. Eine Prozess Engine hilft Key Performance Indikatoren zu berechnen und komplexe Ereignisse zu erkennen und zu verarbeiten, so dass nur eine Plattform benötigt wird. In [Wa08] wird eine Architektur für ein einheitliches Ereignis Management bei komplexen Ereignissen vorgestellt.

In [Fr12] wird eine Erweiterung von BPMN vorgestellt um das Konzept von Business Activity Monitoring zu integrieren und Key Performance Indikatoren in BPMN darzustellen. In [Su13] wird eine Erweiterung von BPMN vorgestellt um eine Wireless Sensor Networks (WSN) in Prozessmodelle zu integrieren. Aus dem Modell soll dann Code generiert werden, um Daten aus dem WSN zu erfassen, zu aggregieren und Aktuatoren im WSN zu kontrollieren. In [Zo11] sollen Geschäftsprozessmodelle in der Produktion mit BPMN modelliert werden und BPMN wird hierfür erweitert um Materialflüsse, Maschinen Werkzeuge explizit modellieren zu können. BPMN kann in BPEL transformiert werden und in [Ko11] wird eine Klassifizierung und Evaluierung von verschiedenen BPEL Erweiterungen vorgenommen. In diesem Beitrag werden Alternativen zur Spracherweiterung untersucht, da diese potenziell die Komplexität erhöhen und somit sowohl das Modellieren als auch das Verständnis der Modelle erschweren.

Bei anderen Ansätzen geht es ebenfalls darum Sensordaten zu überwachen: In [WKN08] werden ausführbaren Workflows mit BPEL und Kontext-Information kombiniert, welche von Sensoren oder anderen Informationsquellen kommen, so dass direkt auf die vorhandene Situation reagiert werden kann und ggf. neue Prozesse etwa für die Bestellung von Ersatzteilen angestoßen werden können. In unserem Ansatz soll aber mit einer graphischen Sprache modelliert werden, weil diese auch für Nicht-Experten verständlich ist.

In [Am10] werden Prototypen für Demand-Side Energie Management beschrieben. In einem ersten Schritt wird analysiert, wieviel Energie benötigt wird, allerdings auf Ebene

eines einzelnen Produktstück oder auf Maschinenebene. Dafür müssen Sensoren und Smart Meter angeschlossen werden und die Verbräuche überwacht und analysiert werden. Danach soll dynamische Prozesskontrolle in einer Prozess Engine umgesetzt werden. In diesem Beitrag soll die Überwachung aber nicht auf Maschinen- oder Stückerbene erfolgen sondern auf Prozessschritt- oder Subprozessebene.

### **3 Prozessorientierte Überwachung**

#### **3.1 Ziele**

Prozessorientierte Überwachung bedeutet, dass die Werte, etwa Messwerte von Sensoren, die für die Überwachung relevant sind, bestimmten Prozessschritten und Prozessinstanzen zugeordnet werden und im Kontext des Prozesses betrachtet werden. Mit den entsprechenden Sensoren ermöglicht dies eine genaue Analyse der Stromverbräuche, z.B. für den Einbau einer Autotür bei der Automobilherstellung.

Prozessorientierte Überwachung in der Produktion kann mehrere Ziele verfolgen, zum Beispiel die Anwendungsfälle für Erklärungsmodelle in den Umsetzungsempfehlungen für den Industrie 4.0-Kontext [Um13]:

- Überwachung des regulären Betriebsstandes
- Erkennung von Verschleiß, etwa von Maschinenteilen, ohne den Produktionsprozess zu unterbrechen
- Vorhersage von Ausfällen oder Störungen von Komponenten

#### **3.2 Anwendungsfall**

Das Konzept der prozessorientierte Überwachung wurde anhand einer Automobilfertigungsstraße verifiziert. Abb. 1 zeigt einen Ausschnitt eines modellierten Schweißprozesses in der Automobilfertigung. Der Schweißprozess besteht aus 15 Stationen von denen 7 tatsächlich schweißen. In den anderen wird transportiert, ausgemessen oder gewartet. Es sollte eine Auswertung der Energieverbräuche pro Station erfolgen, welche hier als Subprozess modelliert wurden. Es wurden 220 Sensoren simuliert. Die Intermediate Catch Events warten auf ein Ereignis welches den Start einer einzelnen Station anzeigt. Dieses kann entweder durch Sensoren ausgelöst werden oder durch das Steuerungssystem. Das Ende der Bearbeitung in einer Station kann entweder implizit modelliert werden durch den Start der Bearbeitung in der nächsten Station oder, ähnlich wie beim Start, durch ein Ereignis ausgelöst durch einen Sensor oder das Steuerungssystem.

Um die Ereignisse zu empfangen, muss die Applikation sie vorher abonnieren. Dies

kann generisch geschehen, etwa durch die Benutzung von Wildcards.

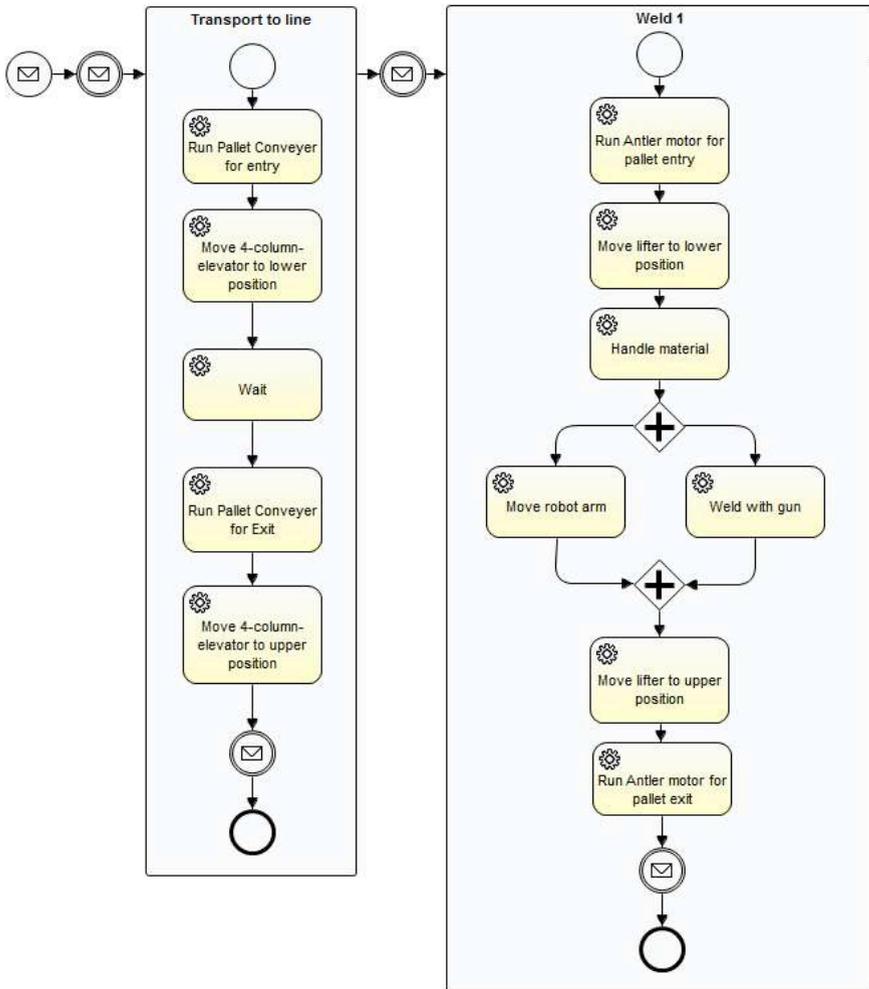


Abb. 1: Ausschnitt BPMN Diagramm Schweißen

Jede Automobilkarosse im Schweißprozess stößt eine neue Instanz an, die durch die Karosserieerkennung identifiziert werden kann. Bei Eintreffen eines Sensorereignis wird geprüft, zu welcher Instanz dieses Ereignis gehört und ob es zu dem gerade aktiven Subprozess passt. Ist dies der Fall, werden Informationen über die Instanz und den Subprozess in den Body des Ereignisses geschrieben und je nach Bedarf erneut über den Broker verschickt (mit abgeändertem Topic um die Ereignisse von den ursprünglichen zu unterscheiden) oder direkt in eine Datenbank gespeichert, siehe auch Abb. 2.

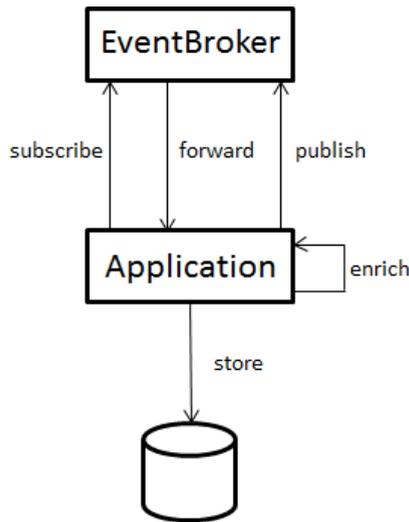


Abb. 2: Ereignisverarbeitung

### 3.3 Technische Umsetzung

Für die Modellierung und das Ausführen von Geschäftsmodellen wurde *activiti*<sup>3</sup> verwendet. *Activiti* ist ein Open-Source Workflow Management System, welches u.a. eine Process Engine und ein Plugin für Eclipse<sup>4</sup> beinhaltet, mit dem man Geschäftsprozesse modellieren kann. Das BPMN Modell selber kann sowohl in graphischer Form als auch als XML-Serialisierung bearbeitet werden.

Um Sensordaten mit Prozessschritten zu verknüpfen gibt es verschiedene Ansätze: man kann zum einen die Daten speichern und nachträglich den Prozessschritt und die Instanz anhand des Zeitstempels ermitteln oder man kann die Sensorevents zur Laufzeit mit den Prozessinformationen anreichern. In den hier beschriebenen Prototypen wird der erste Ansatz verwendet.

Ereignisse (Events) mit Sensordaten werden mittels eines Eventbrokers übertragen. Events bestehen aus Topics und einem Body. Jeder Sensor hat ein anderes Topic.

In einem ersten Prototyp wurden die Eventtopics manuell zugeordnet und zur Laufzeit aufgelöst. Schon bei mittleren Sensornetzwerken ist dies nicht mehr praktikabel und wird im nächsten Prototyp zumindest halb-automatisch geschehen. Die Zuordnung selber ist abhängig davon ob die Zuordnung auf Subprozess- oder Prozessschritzebene passieren soll. Die Zuordnung selber wird in der BPMN-Datei gespeichert. Um die

---

<sup>3</sup> <http://www.activiti.org>

<sup>4</sup> <https://eclipse.org/ide>

Komplexität nicht zu erhöhen, werden hier keine Erweiterungen an dem BPMN-Standard vorgenommen. Stattdessen werden die Mechanismen, die activiti bietet genutzt.

Im folgendem ist die XML-Serialisierung eines Prozessausschnitts dargestellt.

```
<process id="myProcess" name="My process"
isExecutable="true">
  <startEvent id="startevent1" name="Start">
  </startEvent>
  <endEvent id="endevent1" name="End"></endEvent>
  <serviceTask id="servicetask1" name="Service Task 1"
    activiti:class="de.beispiel.TaskHandler">
  </serviceTask>
  <sequenceFlow id="flow1" sourceRef="startevent1"
    targetRef="servicetask1">
  </sequenceFlow>
  <intermediateCatchEvent Id="msgintermcatch1"
    name="MessageCatchEvent">
    <messageEventDefinition messageRef="msg1Ref">
    </messageEventDefinition>
  </intermediateCatchEvent>
  <sequenceFlow id="flow4" sourceRef="servicetask1"
    targetRef="msgintermcatch1">
  </sequenceFlow>
  <serviceTask id="servicetask2" name="Service Task 2"
    activiti:class="de.beispiel.TaskHandler">
  </serviceTask>
  <sequenceFlow id="flow5" sourceRef="msgintermcatch1"
    targetRef="servicetask2">
  </sequenceFlow>
  <sequenceFlow id="flow6" sourceRef="servicetask2"
    targetRef="endevent1">
  </sequenceFlow>
</process>
```

Für die Zuordnung der Topics bei serviceTasks wird das von Activiti implementierte „Field Injection“ verwendet. Es gibt zwei Arten der Injektion: für fixe String Werte und für Ausdrücke. Die implementierende Klasse muss diese Felder dann definieren und setter und getter-Methode hinzufügen. Die Zuordnung der Topics wird als JSON-Objekt in dem Modellierungswerkzeug als Field definiert oder direkt in die XML-Serialisierung eingebettet, was dann wie folgt aussieht:

```
<serviceTask id="servicetask1" name="Service Task 1"
  activiti:class="de.beispiel.TaskHandler">
  <extensionElements>
```

```
<activiti:field name="testList">
  <activiti:expression>
  {
  "sensors": [
  {"topic": "RAW/MSRMT/L1/ST2/R5/S1"},
  {"topic": "RAW/MSRMT/L1/ST2/R6/S1"},
  {"topic": "RAW/MSRMT/L1/ST2/R7/S1"},
  {"topic": "RAW/MSRMT/L1/ST2/R8/S1"}]
  }
  </activiti:expression>
</activiti:field>
</extensionElements>
</serviceTask>
```

In diesem Fall werden dem ServiceTask1 vier verschiedene Topics und damit vier verschiedene Sensoren zugeordnet.

Für Message Catch Events wird ähnlich verfahren, nur dass die Injektion bei einem Listener erfolgt.

## 4 Simulation

Es gibt verschiedene Anwendungsfälle für eine Simulation des Prozesses: um die Implementierung zu testen oder das Prozessmodell zu prüfen, für Demonstrationszwecke oder auch um Änderungen im System zu simulieren und die Auswirkungen zu analysieren.

Um den Prozess zu simulieren wird neben dem Prozessmodell eine Simulation der Sensoren bzw. deren Nachrichten benötigt. Es gibt BPMN-Modellierungswerkzeuge, die die Simulationen für die erstellten Modelle ausführen können, allerdings würden in unserem Fall nur Nachrichten simuliert für die ein Catch-Event modelliert wurde. Die Nachrichten mit den zu überwachenden Sensoren würden nicht simuliert. Eine andere Möglichkeit wäre eine Simulation der Sensoren und der Versand über einen MQTT-Broker. Aber hier müssten alle Sensoren neu modelliert oder konfiguriert werden, welches bei einer mittleren Anzahl von Sensoren aufwendig ist.

Um den Aufwand möglichst gering zu halten, soll das Produktionsmodell, welches zur Überwachung verwendet wird auch für die Simulation verwendet werden. Dafür sind natürlich einige Änderungen nötig, da bei der Ausführung des Modells ja Nachrichten verschickt und nicht empfangen werden sollen. Außerdem sind zusätzliche Informationen notwendig, wie etwa die Bearbeitungsdauer in einer Station oder eines einzelnen Roboters.

Folgende Änderungen sollen vorgenommen werden:



**Message Start Event:** In activiti gibt es 4 verschiedene Arten von Start Events (die BPMN Spezifikation hingegen nennt 10 verschiedenen Events wenn man zwischen unterbrechenden und nicht-unterbrechenden Events unterscheidet [Bu11]): None Start Event, Message Start Event, Timer Start Event und Error Start Event. Abhängig von dem zu simulierenden Prozess, können verschiedene Ansätze nützlich sein. Wenn eine Produktionslinie simuliert werden soll, bei der regelmäßig eine neue Prozessinstanz gestartet werden soll, ist das Timer Event sinnvoll, ansonsten sollte a none start event verwendet werden. Um dies zu entscheiden, müssen Informationen über die Intervalllänge im Modell enthalten sein. Diese müssen manuell eingetragen werden. Nach dem StartEvent muss eine Nachricht verschickt werden, der im Modell zur Überwachung eine neue Instanz aktiviert. Dafür wird ein neuer ServiceTask generiert, der die Nachricht mit dem gemappten Topic verschickt.



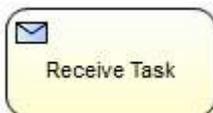
**MessageIntermediateCatchEvent:** dieses sollte ersetzt werden durch ein IntermediateThrowEvent, welches allerdings bisher nicht von Activi unterstützt wird, genauso wenig wie SendTasks für Nachrichten die an einen Message Broker versendet werden sollen. Deswegen sollten die IntermediateCatchEvents durch ServiceTasks ersetzt werden. Für ServiceTasks in activiti müssen Implementierungsklassen hinterlegt werden. In diesen Klassen kann das Versenden von Nachrichten zum MessageBroker implementiert werden.



**IntermediateSignalCatchEvent:** dies müsste genauer untersucht werden. Wahrscheinlich keine Ersetzung nötig, da Signale häufig im gleichen Prozess verschickt und empfangen werden (wenn von außerhalb verschickt werden soll, werden eher Nachrichten verwendet). Es ist aber möglich Signale sowohl programmatisch als auch zwischen Prozessen zu verschicken.



**EventGateway:** Normalerweise weist dieses Gateway darauf hin, dass es verschiedene Event-Möglichkeiten gibt und somit verschiedene Gateways folgen. Wie genau damit umgegangen werden soll, muss genauer untersucht werden.



**ReceiveTask:** sollte logisch gesehen durch einen SendTask

abgelöst werden. Laut einem activiti Entwickler<sup>5</sup> gibt es diesen bereits in einem experimentellen Stadium, ist aber noch nicht dokumentiert. Diesen SendTask kann man entweder wie einen MailTask oder wie einen WebserviceTask verwenden, bringt also keine neue Funktionalität. Insbesondere kann man keine Nachrichten an MessageBroker versenden, so dass auch dieser Task durch einen ServiceTask ersetzt werden muss.



ServiceTask: hier muss der Task nicht ersetzt werden, sondern nur die Implementierung. Wenn man davon ausgeht, dass ein Prozess mit einem Sensornetzwerk überwacht/ simuliert werden soll, welches etwa Energieverbräuche misst und dass der ServiceTask die Einheit repräsentiert, welches die Nachrichten verschickt, dann sollte die Simulationsimplementierung genau diese Nachrichten versenden.

Listener welche im Modell für die Überwachung definiert werden, sollten generell ersetzt werden (z.B. bei IntermediateCatchEvents) oder komplett entfernt (etwa bei ServiceTasks).

Das Modell braucht außerdem zusätzliche Informationen (zusätzlich zu den Informationen, wie häufig ein Prozess gestartet wird, falls dies regelmäßig und automatisch geschehen soll): MessageBroker URL und die Arbeitszeit der einzelnen Maschinen oder Maschinenteile.

## 5 Zusammenfassung und Ausblick

In diesen Artikel wurde ein Ansatz zur prozessorientierte Überwachung vorgestellt. Die Machbarkeit wurde anhand einer Produktionsstraße verifiziert. 220 Sensoren wurden simuliert und Prozessschritten zugeordnet. Mit Hilfe der Daten konnte eine genaue Analyse der Energieverbräuche auf Maschinen- und Prozessebene durchgeführt werden.

In den nächsten Schritten soll die Implementierung ausgebaut werden: der nächste Prototyp für die prozessorientierte Überwachung soll eine vereinfachte Zuordnung zwischen Prozessschritt und Ereignis Topic ermöglichen. Die Nützlichkeit der vorgestellten Kombination aus Prozess- und Simulationsmodell soll in einer weiteren Studie gezeigt werden.

Der Prototyp für die prozessorientierte Überwachung sollte auch noch auf Skalierbarkeit untersucht werden. Bei den bisherigen Tests hatte der Prozess maximal 130 Sensoren, die gleichzeitig Events verschickt haben bei 15 Prozessinstanzen. Das System war so konfiguriert, dass die Sensoren etwa einmal in der Sekunde ein Update bezüglich des

---

<sup>5</sup> <http://forums.activiti.org/content/send-task-not-recognized-designer>, 2015/01/05

Stromverbrauchs verschickt haben.

Es ist außerdem beabsichtigt, den Prototypen in einer realen Fabrikhalle zu installieren um dann evaluieren zu können, wie gut sich der Ansatz für die Überwachung des Produktionsprozesses und die vorausschauende Wartung eignet.

### **Danksagung**

Diese Arbeit wurde von dem über das BMWI –geförderte Leitprojekt E3-Produktion<sup>6</sup> der Fraunhofer Gesellschaft unterstützt.

### **Literaturverzeichnis**

- [Am10] Ameling, Michael, et. al: "Demand-side Energy Management Systems for Manufacturing." In GI Jahrestagung (1), S. 467-472, 2010
- [Am08] Ammon, Rainer von; et.al "Event-driven business process management and its practical application taking the example of DHL", In: 1st International workshop on Complex Event Processing for the Future Internet, Vienna Austria, S. 1-13, 2008
- [Bu11] Business Process Model and Notation (BPMN) Version 2.0, 2011.
- [EH12] Estruch, Antonio, Heredia Álvaro, José Antonio: "Event-driven manufacturing process management approach" In Business Process Management, Springer Berlin Heidelberg, S. 120-133, 2012.
- [Fr12] Friedenstab, J.; et al.: "Extending BPMN for business activity monitoring." In 2012 45th Hawaii International Conference on System Science (HICSS), S.4158-4167, 2012.
- [Ko11] Kopp, Oliver, et al.: "A classification of BPEL extensions." Journal of Systems Integration Vol 2, No 4 (2011), S.3-28. , 2011
- [Su13] Sungur, C.T.; Spiess, P.; Oertel, N.; Kopp, O.: "Extending BPMN for Wireless Sensor Networks," in 2013 IEEE 15th Conference on Business Informatics (CBI), pp.109-116, 2013
- [Um13] Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0, 2013.
- [Wa08] Walzer, Karen, et al.: "Event-driven manufacturing: Unified management of primitive and complex events for manufacturing monitoring and control." in IEEE International Workshop on Factory Communication Systems, S. 383-391, 2008.
- [WKN08] Wieland, Matthias; Kaczmarczyk, Peter, Nicklas, Daniela: "Context integration for smart workflows." In Pervasive Computing and Communications, PerCom 2008. Sixth Annual IEEE International Conference on, S. 239-242, 2008.
- [Zo11] Zor, S.; Leymann, F.; Schumm, D.: "A proposal of BPMN extensions for the manufacturing domain," in CIRP Conference on Manufacturing Systems, 2011.

---

<sup>6</sup> <https://www.fraunhofer.de/de/forschungsfelder/fraunhofer-leitprojekte/e3-produktion.html>