Echtzeitüberwachung und Langzeitanalyse mittels eingebetteter Systeme

Tino Noack
TU Cottbus
Institut für Informatik, Informations- und Medientechnik
Lehrstuhl Datenbank- und Informationssysteme
Tino.Noack@tu-cottbus.de

Kurzfassung

Der vorliegende Beitrag skizziert ein interdisziplinäres Forschungsvorhaben im Rahmen einer Doktorarbeit. Einer der Forschungsbeiträge ist die Kombination von Echtzeitüberwachung und Langzeitanalyse. Diese Kombination basiert auf existierenden Ansätzen und umfasst Event-Condition-Action-Regeln (ECA-Regeln), Data-Mining-Technologien sowie Complex Event Processing (CEP). Im vorliegenden Beitrag werden zunächst drei grundlegende Annahmen und fünf Überwachungsanforderungen erarbeitet. Darauf aufbauend wird die Forschungsfrage detailliert betrachtet. Die Grundlage für die vorgestellte Idee bildet ein mathematisches Modell (der Zustandsraum), welches das Wissen über das zu überwachende System repräsentiert. Mit Hilfe dieses Zustandsraums werden durch die Anwendung von Data-Mining-Technologien ECA-Regeln erzeugt und an eine CEP-Anwendung, die sich auf einem eingebetteten System befindet, übertragen. Dieser Teilschritt bezieht sich auf die Langzeitanalyse. Die CEP-Anwendung wertet anschließend die übertragenen ECA-Regeln auf einem kontinuierlichen Strom von Sensordaten aus und erzeugt Aktionen. Dieser Teilschritt bezieht sich auf die Echtzeitüberwachung. Weiterhin wird eine Prozesskette vorgestellt, die zyklisch durchlaufen wird und zur Kombination von Echtzeitüberwachung und Langzeitanalyse dient. Hier wird ein dynamischer und flexibler Überwachungsansatz vorgestellt.

Schlüsselwörter

Überwachung, Echtzeit, Langzeit, Eingebettete Systeme, Datenströme, Data Mining, Complex Event Processing

1. EINLEITUNG

Viele Produkte, in denen sich eingebettete Systeme verbergen, sind sicherheitskritisch und unterliegen Echtzeitanforderungen wie z.B. Kraft-, Schienen-, Luft- oder Raumfahrzeuge. Eingebettete Systeme werden oft für Regelungs-, Kontroll- und Überwachungsfunktionalitäten eingesetzt. Die Überwachung technischer Systeme ist ein sehr weit verbreitetes Forschungsfeld und bezieht sich auf viele heterogene Anwendungsdomänen. Häufig werden Überwachungssysteme für spezielle Anwendungen entworfen, entwickelt und implementiert. Dies führt zu erhöhten Entwicklungskosten und gleichzeitig zur Abnahme der Flexibilität bzw. der Wiederverwendbarkeit. Bedeutende Anwendungen sind z.B. die Überwachung von Raumfahrzeugen [21], [22] oder die Überwachung von Schienenfahrzeugen [16]. Die Überwachung von Raumfahrzeugen ist besonders herausfordernd, da komplette Systemtests in der vorgesehenen Systemumwelt (dem Weltraum) und kontinuierliche Wartung unpraktisch bzw. unmöglich sind.

Aufgrund der steigenden Komplexität heutiger Produkte werden verbesserte Überwachungsansätze benötigt, die heutige und zukünftige Anforderungen berücksichtigen. Im vorliegenden Beitrag steht die Überwachung des zu überwachenden Systems, welches im Weiteren als Produkt bezeichnet wird, im Vordergrund. Das Produkt besteht aus einer Menge von Systemkomponenten. Nur aufgrund des Zusammenspiels der einzelnen Systemkomponenten untereinander genügt das Produkt einer vorher definierten Funktion bzw. Aufgabe. Zusätzlich wirken externe Einflüsse aus der umgebenden Produktumwelt auf das Produkt (vgl. [15], [17]). Somit bezieht sich die Überwachung des Produkts je nach Anwendungsdomäne und je nach Überwachungsziel zusätzlich auf externe Einflüsse und auf die korrekte Arbeitsweise der beteiligten Systemkomponenten. Eine strikte Trennung der Überwachung externer Einflüsse, des Produkts selbst und der einzelnen Systemkomponenten, aus denen das Produkt besteht, kann nicht immer vollzogen werden.

Der vorliegende Beitrag skizziert ein interdisziplinäres Forschungsvorhaben im Rahmen einer Doktorarbeit. Einer der Forschungsbeiträge ist die Kombination von existierenden, gut bekannten und bereits praktisch angewendeten Ansätzen, die für die Kombination von Echtzeitüberwachung und Langzeitanalyse eingesetzt werden können. Anhand des Einsatzes von existierenden Ansätzen sind Einsparungen im Bereich der Entwicklungskosten möglich. Das Forschungsvorhaben umfasst die Erstellung von Event-Condition-Action-Regeln (ECA-Regeln) [10], Data-Mining-Technologien [27] sowie Complex Event Processing (CEP) [12]. Hier wird ein dynamischer und flexibler Überwachungsansatz vorgestellt, der auf den drei folgenden Annahmen basiert:

1. Anwendungsübergreifend werden ähnliche Methodiken und Algorithmen für die Überwachung technischer Systeme eingesetzt.

23^{rd} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 31.05.2011 - 03.06.2011, Obergurgl, Austria.

2. Das Auftreten von Fehlern im laufenden Betrieb lässt sich nicht ausschließen. Daher muss durch die Änderung des Systemverhaltens so schnell wie notwendig eine angemessene Aktion ausgelöst werden.

3. Teile des gesamten Überwachungsprozesses sind semimanuell. Informationssysteme werden nur zur Unterstützung des Überwachungsprozesses angewendet.

Der Rest des vorliegenden Beitrags ist wie folgt organisiert. Kapitel 2 beschreibt ein Anwendungsbeispiel. In Kapitel 3 wird der Begriff eines eingebetteten Systems definiert, so wie es für die Forschungsarbeit verwendet wird. Kapitel 4 fasst Anforderungen an die Überwachung zusammen und aufbauend darauf wird in Kapitel 5 die Forschungsfrage detailliert betrachtet. Kapitel 6 beschreibt das Systemmodell, welches dem vorgeschlagenen Überwachungsansatz zu Grunde liegt. In Kapitel 7 wird der vorgeschlagene Überwachungsansatz detailliert beschrieben. Kapitel 8 fasst existierende Lösungen zusammen und schließlich wird in Kapitel 9 eine Zusammenfassung gegeben.

2. ANWENDUNGSBEISPIEL: ZUGUNGLÜCK VOM ICE 884 IN ESCHEDE

Das Zugunglück vom ICE 884 in Eschede ist ein sehr praxisnahes Anwendungsbeispiel. Die Hauptursache des katastrophalen Zugunglücks war der Bruch eines gummigefederten Radreifens. Dieser Bruch war die Folge von langfristigen Verschleißerscheinungen (z.B. Verringerung der Radreifendicke und Korrosion). Bereits einige Monate vor dem Unglück wurden während der Wartung anomale Messwerte an dem besagten Radreifen festgestellt. Die detaillierte Bruchflächenanalyse stellte heraus, dass die langfristigen Verschleißerscheinungen zu einem Riss in dem Radreifen, lange vor dem Unglück, führten. Der Bruch des Radreifens führte zur Entgleisung des Zuges ([24], [13]). Der beschriebene Anwendungsfall deutet auf langfristige und auf kurzfristige Einflussfaktoren hin. Verschleißerscheinungen sind langfristige Einflussfaktoren und der Bruch des Radreifes bzw. die Zugentgleisung sind kurzfristige Einflussfaktoren.

Durch die Langzeitanalyse können langfristige Verschleißerscheinungen erkannt, analysiert und bewertet werden. Der Bruch des Radreifens und die nachfolgende Entgleisung des Drehgestells haben zu einer plötzlichen und signifikanten Veränderung des Systemverhaltens geführt (bspw. Schlingerbewegung des entgleisten Drehgestells). Anhand der Anwendung der Echtzeitüberwachung mittels eines eingebetteten Systems kann diese plötzliche Veränderung des Systemverhaltens erkannt und in einem angemessenen Zeitraum eine Aktion (z.B. Notbremsung) durchgeführt werden.

3. EINGEBETTETES SYSTEM

Abbildung 1 skizziert die abstrakte Architektur eines eingebetteten Systems, wie sie hier Einsatz findet. Eingebettete Systeme sind in ein umgebendes Produkt eingebettet. Das Produkt ist in eine Produktumgebung eingebettet. Eingebettete Systeme enthalten elektronische Baugruppen (Hardware), die die Systemkomponenten repräsentieren. Zusätzlich sind diese elektronischen Baugruppen mit Software ausgestattet. Eingebettete Systeme unterliegen eingeschränkten Systemressourcen wie z.B. Prozessorleistung, Strom- und Speicherverbrauch. Das eingebettete System steht mittels

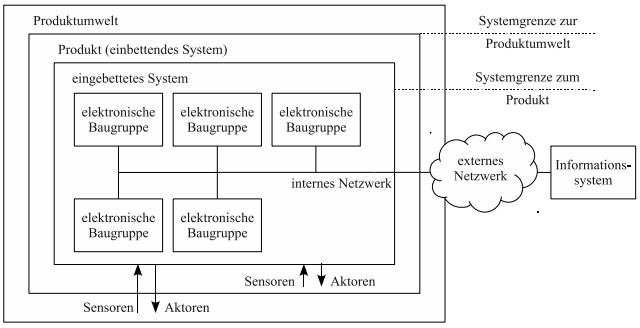

Abb. 1: Eingebettetes System

Sensoren und Aktoren mit dem Produkt und der Produktumgebung in Interaktion. Die elektronischen Baugruppen können mittels eines internen Netzwerkes miteinander verbunden sein. Zusätzlich kann eine temporäre Verbindung zu einem externen Informationssystem vorhanden sein. Weitere Informationen zu eingebetteten Systemen finden sich u.a. in [28], [20] und [23].

4. ÜBERWACHUNGSANFORDERUNGEN

Entsprechend des vorgestellten Anwendungsbeispiels und in Anbetracht der abstrakten Architektur eines eingebetteten Systems werden hier folgende fünf Überwachungsanforderungen erarbeitet: Zeit, Lokalität, Wissen, Systemressourcen und Schärfe. Abbildung 2 fasst die genannten Anforderungen zusammen.

Zeit: Diese Anforderung bezieht sich auf die zeitliche und kontinuierliche Veränderung der Bauteile.

- *kurzfristig*: Es können plötzliche Änderungen der Bauteile (z.B. Bruch des Radreifens) auftreten. Es ist notwendig, diese in Echtzeit zu erkennen.
- *langfristig*: Zur Erkennung langfristiger Einflussfaktoren und Veränderungen (z.B. Verschleiß und Alterung) sind Langzeitanalysen notwendig.

Lokalität: Diese Anforderung bezieht sich auf Wechselwirkungen der Einflussfaktoren und die räumliche Lokalität der Überwachung.

- *lokal*: Fehler, die sich z.B. auf wenige Bauteile beziehen, müssen durch eine lokale Überwachung erkannt werden.
- *global*: Aufgrund der steigenden Komplexität von Produkten und eingebetteter Systeme korrelieren die Einflussfaktoren zunehmend. Somit entstehen komplexe Zusammenhänge zwischen den Bauteilen, die durch eine globale Analyse erfasst und erkannt werden müssen.

Wissen: Diese Anforderung bezieht sich auf das vorhandene Wissen über das eingebettete System, das Produkt und die Produktumwelt.

- *bekannt*: Es ist notwendig das bekannte Wissen über das eingebettete System, das Produkt und die Produktumwelt möglichst umfassend und zielorientiert für die Überwachung einzusetzen.
- *unbekannt*: Aufgrund unbekannter bzw. unvorhersehbarer Umstände ist ein dynamischer und flexibler Überwachungsprozess notwendig.

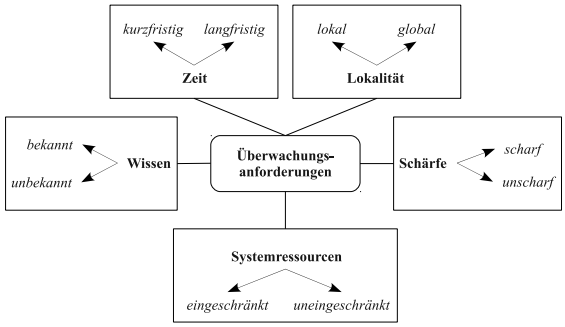

Abb. 2: Überwachungsanforderungen

Systemressourcen: Diese Anforderung bezieht sich auf die vorhandenen Ressourcen, die für die Überwachung zur Verfügung stehen.

- *uneingeschränkt*: Die Überwachung von Systemen benötigt äußerst viele Systemressourcen. Somit ist eine Kombination von interner und externer Überwachung (hybrides Überwachungssystem [26]) notwendig, um ausreichend Ressourcen für die Überwachung zur Verfügung zu stellen.
- *eingeschränkt*: Aufgrund der eingeschränkten Systemressourcen eingebetteter Systeme ist es notwendig, diese angemessen und zielführend für die Überwachung einzusetzen.

Schärfe: Diese Anforderung bezieht sich auf die Auswertung von Bedingungen (vgl. [25], [4]).

- *scharf*: Systemzustände müssen exakt und zuverlässig durch eine exakte binäre Auswertung von Bedingungen (Boolesches Modell) erkannt werden.
- *unscharf*: Diese scharfe Grenze zwischen Systemzuständen ist nicht immer gegeben. Um dies zu berücksichtigen, wird die exakte binäre Auswertung mittels Zugehörigkeitsgrade zwischen 0 und 1 verallgemeinert. Der Wert 1 wird als volle Zugehörigkeit und der Wert 0 als nicht zugehörig interpretiert.

5. FORSCHUNGSFRAGE

Es gibt eine Lücke zwischen Echtzeitüberwachung und Langzeitanalyse von Ereignissen, die die Zuverlässigkeit von Produkten beeinträchtigen. Dies ist die Motivation für unsere Forschung an der Kombination von Echtzeitüberwachung und Langzeitanalyse von Ereignissen. In dem ersten Schritt werden hier alle Überwachungsanforderungen außer der Schärfe betrachtet. Abbildung 3 fasst die Forschungsfrage grafisch zusammen.

Langzeitanalyse benötigt meist sehr viele Systemressourcen. Zusätzlich sind Data-Mining-Technologien semi-manuell und müssen durch Fachpersonal betreut und gepflegt werden. Aus diesem Grund ist eine Offlineverarbeitung auf einem externen Informationssystem mit nahezu uneingeschränkten Systemressourcen notwendig. Data-Mining-Technologien werden hier für das Erlernen von Klassifikatoren eingesetzt, die anschließend durch ECA-Regeln repräsentiert werden. Die persistent gespeicherten Daten umfassen alle gesammelten Attribute und geben eine globale Sicht über das gesamte zu überwachende Produkt. Dabei ist die Anzahl der Attribute je nach Anwendungsdomäne und Überwachungsziel unterschiedlich. Diese Daten können zur Identifikation von relevanten Wechselwirkungen Verwendung finden. Data-Mining-Technologien werden eingesetzt, um das Wissen über das Produkt mit der Zeit zu erhöhen.

In Bezug zum genannten Anwendungsbeispiel werden die Daten mittels eines externen Informationssystems gesammelt. Diese persistent gespeicherten Daten werden eingesetzt, um einen Klassifikator zu erlernen, der zwischen bekanntem und unbekanntem Verhalten des Zuges unterscheiden kann. Dies wird in [8] als Anomalieerkennung bezeichnet. Weiterhin können diese gespeicherten Daten zur Erkennung gradueller Änderungen der Systemkomponenten in Bezug zur Zeit genutzt werden. Somit können langfristige Einflussfaktoren wie z.B. Verschleiß erkannt werden.

Echtzeitüberwachung wird auf dem eingebetteten System durchgeführt. Dieses unterliegt eingeschränkten Systemressourcen. Die Echtzeitüberwachung wird automatisch, online und ohne Benutzerinteraktion durchgeführt. Plötzliche Änderungen des Systemverhaltens müssen so schnell wie notwendig erkannt werden. Anschließend ist eine angemessene Aktion notwendig. Die gelernten Klassifikatoren bzw. ECA-Regeln werden hier zum eingebetteten System übertragen und anschließend zur Erkennung von Änderungen des Systemverhaltens bzw. zur Anomalieerkennung eingesetzt. CEP ist hier ein ausgewähltes Werkzeug, um die ECA-Regeln auf den kontinuierlichen Datenströmen anzuwenden. ECA-Regeln repräsentieren das Wissen über das Produkt. Verhalten, welches nicht zu diesen Regeln passt, kann als unbekannt bzw. anomal gekennzeichnet werden. Dies ist ein lokaler Aspekt, da nur eine Teilmenge der vorhandenen Attribute für die Definition eines speziellen Verhaltens mittels ECA-Regeln Verwendung findet. Wie auch bei der Langzeitanalyse ist die Anzahl der Attribute je nach Anwendungsdomäne und Überwachungsziel unterschiedlich, aber geringer als für die Anwendung der Langzeitanalyse.

In Bezug zum genannten Anwendungsbeispiel stellen das Brechen des Radreifens und die anschließende Entgleisung signifikante und plötzliche Änderungen der Fahreigenschaften des Zuges dar. Nachfolgend wird in Bezug zum Anwendungsbeispiel der ECA-Ansatz kurz erläutert. Ein Ereignis (Event) ist hier das Verhalten des Zuges zu einer bestimmten Zeit. Die Bedingung (Condition) bezieht sich auf die gelernten Klassifikatoren bzw. die Regeln, die ermittelt wurden, um das Verhalten des Zuges zu einem bestimmten Zeitpunkt zu klassifizieren. Eine Aktion (Action) kann bspw. die Verringerung der Geschwindigkeit des Zuges oder das Auslösen der Notbremse sein, um materielle Schäden und menschliche Opfer zu vermeiden.

6. SYSTEMMODELL

Ein wesentlicher Punkt ist das Verständnis der Eingangsdaten. Sensoren erzeugen kontinuierliche Daten. Diese kontinuierlichen Sensordaten werden hier als Datenströme interpretiert. Ein Datenstrom besteht aus einer Sequenz von Datenelementen. Häufig ist diese Sequenz sehr lang. Ein System, welches Datenströme verarbeitet, hat keine A-Priori-Kontrolle über die Reihenfolge der eintreffenden Datenelemente. Die erneute Übertragung von verlorenen Datenelementen ist nicht möglich. Weitere Informationen über Datenströme finden sich u.a. in [1], [7], [3] und [14].

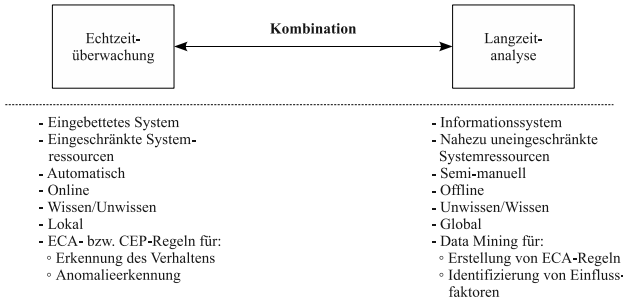

Abb. 3: Kombination von Echtzeitüberwachung und Langzeitanalyse

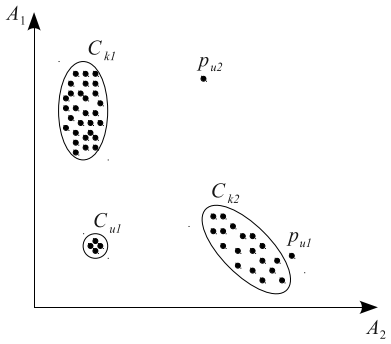

Abb. 4: Zustandsraum [8]

Die Menge von Eigenschaften, die das Zielsystem beschreiben, wird hier als eine Menge von Attributen $A_1, ..., A_n$ interpretiert. Diese Attribute können u.a. nominal, ordinal oder metrisch sein. Attributwerte sind Funktionen der Zeit, so dass Werte von A_i einer Funktion $a_i : T \rightarrow \mathbb{R}$ entsprechen. Dabei ist T die Zeit und $\mathbb{R}$ die Menge der reellen Zahlen. Somit ist ein Zustand in Bezug zur Zeit ein Zustandsvektor

$$\vec{a}(t) = \begin{pmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_n(t) \end{pmatrix}.$$

Der Raum, der durch die Attribute aufgespannt wird, heißt Zustandsraum. Die Anzahl der Attribute definiert die Anzahl der Dimensionen des Zustandsraums. Eine Menge von Zustandsvektoren im Zustandsraum, die ähnliche Arten von Zuständen repräsentieren, können geometrisch interpretiert werden. Diese geometrische Interpretation wird im Rahmen von Data-Mining-Technologien als Cluster bezeichnet ([27], [8], [5], [2]).

Abbildung 4 veranschaulicht den Zustandsraum in einem Zeitfenster unter Berücksichtigung der zwei Attribute A_1 und A_2. Zur besseren Übersichtlichkeit sind die Zustandsvektoren als Punkte dargestellt. Sei S die Menge aller möglichen Systemzustände bzw. der gesamte Zustandsraum, C_k die Menge der bekannten Cluster und C_u die Menge aller unbekannten Cluster, so dass $C_k \bigcup C_u = S$ und $C_k \bigcap C_u = \varnothing$. Somit sind bekannte Cluster komplementär zu unbekannten Clustern. In Abbildung 4 repräsentieren die Cluster C_{k1} und C_{k2} Mengen von bekannten Systemzuständen. Der Cluster C_{u1} sowie die Punkte p_{u1} und p_{u2} stehen exemplarisch für unbekannte Systemzustände. In [8] werden diese unbekannten Systemzustände als Anomalien bezeichnet. Das Ziel des gelernten Systemmodells ist die Klassifizierung eines Zustandsvektors zu einem Zeitpunkt t zu einem bekannten Cluster. Kann dieser Zustandsvektor keinem bekannten Cluster zugeordnet werden, so ist dieser Zustandsvektor unbekannt und wird als eine Anomalie gekennzeichnet. Somit repräsentieren die ECA-Regeln den Klassifikator, der mittels der Data-Mining-Technologien erlernt wurde.

7. KOMBINATION VON ECHTZEITÜBERWACHUNG UND LANGZEITANALYSE

Wie bereits beschrieben, liegt die Kombination von Echtzeitüberwachung und Langzeitanalyse im Fokus des Interesses. Ziel ist es, ein Modell bzw. einen Zustandsraum zu erlernen, der das Wissen über das Produkt repräsentiert. Zunächst wird in diesem Kapitel eine Prozesskette für die Überwachung beschrieben. Anschließend wird diese Prozesskette in eine abstrakte Überwachungsarchitektur überführt.

Die Prozesskette ist in Abbildung 5 grafisch verdeutlicht. Sie ist in zwei Teile gegliedert. Der obere Teil repräsentiert die Echtzeitüberwachung auf dem eingebetteten System. Der untere Teil repräsentiert die Langzeitanalyse auf einem externen Informationssystem. Zur besseren Übersichtlichkeit ist die untere Teilkette in umgekehrter Reihenfolge dargestellt.

Im ersten Schritt startet die Prozesskette mit Ereignissen bzw. Zustandsvektoren. Die Vorverarbeitung ist der zweite Schritt. Dieser kann u.a. zur Filterung, zur Selektion oder für Fensterfunktionen, zur Verringerung des Verarbeitungsaufwands, Verwendung finden. Die Ausführung der Regeln ist der dritte Schritt. In diesem dritten Schritt werden die im Voraus definierten Regeln auf dem Datenstrom angewendet. Der vierte Schritt umfasst das Senden von Nachrichten zu den Aktoren. Der fünfte Schritt wird für die temporäre Speicherung verwendet. Das schließt Datenaggregation zur Minimierung des Speicherbedarfs sowie angemessene Speicherstrategien wie z.B. Ringpuffer oder eingebettete Datenbanken mit ein. Der letzte Schritt der obersten Teilkette bezieht sich auf das Senden der Daten vom eingebetteten System zum stationären System. Aufgrund der temporären Verbindung mittels des externen Netzwerkes können die Daten nur von Zeit zu Zeit an das externe Informationssystem gesendet werden. Die genannten Schritte sind automatisch. Es ist notwendig, dass jeder Teilschritt austauschbar und konfigurierbar (z.B. Plug-in-System) ist, um einen dynamischen und flexiblen Überwachungsansatz bereitzustellen. Somit ist es möglich, das CEP-System auf die vorhandene Hardware und den beabsichtigten Überwachungszweck zuzuschneiden.

Der erste Schritt der Langzeitanalyse betrifft das Laden der empfangenen Daten vom eingebetteten System in ein persistentes Datenverzeichnis wie z.B. ein Data Warehouse (DWH). Der zweite Schritt umfasst die Erstellung der Regeln mittels Data-Mining-Technologien. Dazu gehört die Integration der empfangenen Daten in den Zustandsraum. Dabei steigt das Wissen über das Produkt durch die Integration von neuen und noch unbekannten Zustandsvektoren in den Zustandsraum. Aktuell werden hier folgende Algorithmen zur Klassifikation bzw. überwachtes Lernen eingeschlossen: Regelinduktion, Support Vector Machine und k-nächste Nachbarn. In vielen Fällen müssen die genannten Algorithmen ebenfalls kombiniert werden, um einen ange-

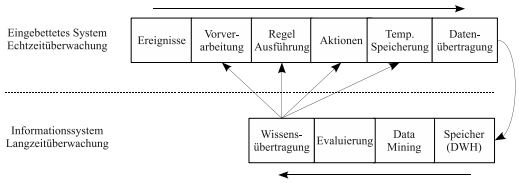

Abb. 5: Prozesskette

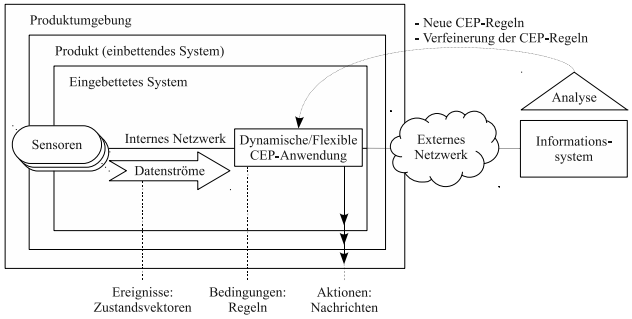

Abb. 6: Überwachungsarchitektur

messenen Klassifikator bereitzustellen ([8], [27]). Der dritte Schritt dient zur Evaluierung der neu ermittelten Regeln und zum Testen mit vorhandenen Regeln, um evtl. Seiteneffekte auszuschließen. Der letzte Schritt der unteren Teilkette betrifft die Übertragung des so ermittelten Wissens zum eingebetteten System. Dies schließt die Anpassung und die Rekonfiguration des bestehenden Überwachungssystems auf Basis des neuen Wissens mit ein. Die genannten Schritte sind semi-manuell und werden durch Fachpersonal betreut.

Die komplette Prozesskette wird zyklisch durchlaufen. So kann mit der Zeit das Wissen über das zu überwachende Produkt gesteigert werden.

Die vorgeschlagene Überwachungsarchitektur ist in Abbildung 6 grafisch verdeutlicht. Sie basiert auf der Prozesskette, die bereits beschrieben wurde. Sensoren erzeugen kontinuierlich Datenströme, die über das interne Netzwerk übertragen werden. Es ist notwendig, diese Ereignisse bzw. Zustandsvektoren kontinuierlich unter Berücksichtigung von Echtzeitbedingungen zu verarbeiten. Das CEP-System muss entsprechend der festgelegten Regeln Aktionen auslösen. Weiterhin wird der Datenstrom aggregiert und temporär gespeichert, bevor er zum externen Informationssystem übermittelt wird. Das externe Informationssystem wird für die Langzeitanalyse und zur Ermittlung neuer bzw. zur Verfeinerung bestehender Regeln eingesetzt. Anschließend ist die Evaluierung der Regeln und die Übertragung zum eingebetteten System notwendig.

Für das beschriebene Forschungsvorhaben können folgende zwei Herausforderungen identifiziert werden.

1. Übersetzung der erlernten Klassifikatoren in verfügbare Anfragesprachen bzw. Funktionen.

2. Erstellung einer dynamischen und flexiblen CEP-Anwendung, die stetig an neue Anforderungen anpassbar ist. Weiterhin muss unter Berücksichtigung der eingeschränkten Systemressourcen und Echtzeitanforderungen ein kontinuierlicher Strom von Zustandsvektoren zuverlässig klassifiziert werden können.

8. EXISTIERENDE LÖSUNGEN

Zur Analyse von Datenströmen werden Datenstrom-Management-Systeme (DSMS), z.B. STREAM [1] oder Aurora [6], eingesetzt. Aurora enthält ein Pfeil-Box-Architektur-Modell, welches einem Plug-in-System ähnlich ist. Ein Überblick über DSMS wird u.a. in [14] gegeben. CEP-Systeme wie CAYUGA [9] oder ESPER [11] werden für das Anwenden von Regeln auf Datenströme mittels Anfragesprachen verwendet. Ein Überblick über CEP-Systeme wird in [12] gegeben. Die genannten DSMS und CEP-Systeme sind nicht für Überwachung mittels Data-Mining-Technologien konzipiert.

NanoMon [29] ist eine sehr spezielle Überwachungssoftware für Sensornetzwerke. MobiMine [19] ist ein mobiles Data-Mining-System für den Aktienhandel. Beide Überwachssysteme unterstützen die genannten Überwachungsanforderungen nicht. Weiterhin enthalten NanoMon und MobiMine keine Anfragesprache.

VEDAS [18] ist ein Datenstrom-Mining-System, welches einigen der hier erarbeiteten Überwachungsforderungen entspricht. Die Erkennung von ungewöhnlichem Fahrerverhalten ist eines der Hauptaugenmerke von VEDAS. Wie auch hier kommen bei VEDAS Data-Mining-Technologien zum Einsatz. Der Unterschied liegt in der Verwendung von unüberwachtem Lernen für Datenstrom-Mining. Weiterhin gibt es keine strikte Trennung zwischen Echtzeitüberwachung und Langzeitanalyse sowie zwischen automatischen und semiautomatischen Funktionen. Dieses Argument kann durch die interaktive Verbindung vom externen Informationssystem zum eingebetteten System untermauert werden. Weiterhin wird bei VEDAS die Evaluierung vernachlässigt. Zusätzlich wird die Überwachungsanforderung Lokalität nicht berücksichtigt. In VEDAS ist das eingebettete System so konfiguriert, dass alle Attribute für die Überwachung Verwendung finden. Dies kann unter Umständen zu sehr hohem Rechenaufwand führen.

9. ZUSAMMENFASSUNG

Es besteht ein Bedarf an neuen Lösungen für die Überwachung von Systemen, die heutige und zukünftige Anforderungen in Betracht ziehen. Der vorliegende Beitrag skizziert ein interdisziplinäres Forschungsvorhaben im Rahmen einer Doktorarbeit. Einer der Forschungsbeiträge ist die Kombination von Echtzeitüberwachung und Langzeitanalyse mittels eingebetteter Systeme, ECA-Regeln, Data-Mining-Technologien und CEP. Drei Annahmen bilden die Basis für den beschriebenen Überwachungsansatz. Weiterhin wurden hier fünf Überwachungsanforderungen erarbeitet. Die Analyse bestehender Lösungen zeigt, dass die dargestellten Überwachungsanforderungen nur unzureichend in Betracht gezogen werden. Aufbauend darauf wurde hier ein dynamischer und flexibler Überwachungsansatz vorgestellt. Der hier vorgestellte Überwachungsansatz basiert auf einem mathematischen Modell, welches als Zustandsraum bezeichnet wird. Dieser Zustandsraum repräsentiert das Wissen über das Produkt, welches im laufenden Betrieb überwacht wird. Weiterhin wurde eine Prozesskette erläutert. Diese Prozesskette wird zyklisch durchlaufen und somit das Wissen über das Produkt mit der Zeit gesteigert. Der Zustandsraum wird mit der Hilfe von Data-Mining-Technologien in ECA-Regeln übersetzt und an eine CEP-Anwendung, die sich auf einem eingebetteten System befindet, übertragen. Durch die CEP-

Anwendung werden die ECA-Regeln verwendet, um die kontinuierlich eintreffenden Zustandsvektoren als bekannt oder unbekannt zu klassifizieren.

10. LITERATUR

[1] Babcock, B. ; Babu, S. ; Datar, M. ; Motwani, R. ; Widom, J. : Models and Issues in Data Stream Systems. In: *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM, 2002, S. 1–16

[2] Bellmann, R. : *Adaptive Control Processes*. Princeton University Press, 1961

[3] Bifet, A. ; Kirkby, R. : Data Stream Mining - A Practical Approach / Centre for Open Software Innovation (COSI) - Waikato University. Version: 2009. `http://moa.cs.waikato.ac.nz/wp-content/uploads/2010/05/StreamMining.pdf`. – Forschungsbericht

[4] Borgelt, C. ; Klawonn, F. ; Kruse, R. ; Nauck, D. : *Neuro-Fuzzy-Systeme: Von den Grundlagen künstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen*. Vieweg, 2003

[5] Boslaugh, S. ; Watters, P. A.: *Statistics in a Nutshell*. O'Reilly, 2008

[6] Carney, D. ; Çetintemel, U. ; Cherniack, M. ; Convey, C. ; Lee, S. ; Seidman, G. ; Stonebraker, M. ; Tatbul, N. ; Zdonik, S. : Monitoring Streams: A New Class of Data Management Applications. In: *VLDB '02: Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB Endowment, 2002, S. 215–226

[7] Chakravarthy, S. ; Jiang, Q. : *Stream Data Processing: A Quality of Service Perspective*. Springer, 2009

[8] Chandola, V. ; Banerjee, A. ; Kumar, V. : Anomaly detection: A survey. In: *ACM Comput. Surv.* 41 (2009), S. 15:1–15:58

[9] Demers, A. J. ; Gehrke, J. ; Panda, B. ; Riedewald, M. ; Sharma, V. ; White, W. M.: Cayuga: A General Purpose Event Monitoring System. In: *CIDR*, 2007, S. 412–422

[10] Dittrich, K. R. ; Gatziu, S. ; Geppert, A. : The Active Database Management System Manifesto: A Rulebase of ADBMS Features. In: *SIGMOD Rec.* 25 (1996), Nr. 3, S. 40–49

[11] EsperTech: *Esper*. `http://www.espertech.com/products/esper.php`. Version: 2011. – Online: 30.03.2011

[12] Etzion, O. ; Niblett, P. : *Event Processing in Action*. Manning Publications Co., 2010

[13] Fischer, G. ; Grubisic, V. : Praxisrelevante Bewertung des Radbruchs vom ICE 884 in Eschede. In: *Materialwissenschaft und Werkstofftechnik* 38 (2007), Nr. 10, S. 789–801

[14] Golab, L. ; Özsu, M. T.: *Data Stream Management*. Morgan & Claypool Publishers, 2010

[15] Gordon, G. : *Systemsimulation*. Oldenbourg, 1972

[16] Guo, Y. : *Algorithmen zur On-Board-Diagnose von Fahrwerksschäden an Schienenfahrzeugen*, TU Berlin, Diss., 2005. `http://opus.kobv.de/tuberlin/volltexte/2005/1120/`

[17] Imboden, D. M. ; Koch, S. : *Systemanalyse*. Springer, 2003

[18] Kargupta, H. ; Bhargava, R. ; Liu, K. ; Powers, M. ; Blair, P. ; Bushra, S. ; Dull, J. ; Sarkar, K. ; Klein, M. ; Vasa, M. ; Handy, D. : VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring. In: *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004

[19] Kargupta, H. ; Park, B.-H. ; Pittie, S. ; Liu, L. ; Kushraj, D. ; Sarkar, K. : MobiMine: Monitoring the Stock Market from a PDA. In: *SIGKDD Explor. Newsl.* 3 (2002), S. 37–46

[20] Marwedel, P. : *Eingebettete Systeme*. Springer-Verlag, 2007

[21] Noack, E. ; Belau, W. ; Wohlgemuth, R. ; Müller, R. ; Palumberi, S. ; Parodi, P. ; Burzagli, F. : Efficiency of the Columbus Failure Management System. In: *AIAA 40th International Conference on Environmental Systems*, 2010

[22] Noack, E. ; Noack, T. ; Patel, V. ; Schmitt, I. ; Richters, M. ; Stamminger, J. ; Sievi, S. : Failure Management for Cost-Effective and Efficient Spacecraft Operation. In: *Proceedings of the 2011 NASA/ESA Conference on Adaptive Hardware and Systems*, IEEE Computer Society, 2011 (AHS '11). – To appear

[23] Peckol, J. K.: *Embedded Systems: A Contemporary Design Tool*. John Wiley & Sons, 2007

[24] Richard, H. ; Fulland, M. ; Sander, M. ; Kullmer, G. : Fracture in a rubber-sprung railway wheel. In: *Engineering Failure Analysis* 12 (2005), Nr. 6, S. 986 – 999

[25] Schmitt, I. : QQL: A DB&IR Query Language. In: *The VLDB Journal* 17 (2008), S. 39–56

[26] Tsai, J. J. P. ; Yang, S. J. H.: *Monitoring and Debugging of Distributed Real-Time Systems*. IEEE Computer Society Press, 1995

[27] Witten, I. H. ; Frank, E. ; Hall, M. A.: *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011

[28] Wolf, F. : *Behavioral Intervals in Embedded Software: Timing and Power Analysis of Embedded Real-Time Software Processes*. Kluwer Academic Publishers, 2002

[29] Yu, M. ; Kim, H. ; Mah, P. : NanoMon: An Adaptable Sensor Network Monitoring Software. In: *IEEE International Symposium on Consumer Electronics (ISCE)*, 2007