



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Anwendungen 1 - WiSe 11/12

Armin Steudte

Complex Event Processing

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Armin Steudte

Complex Event Processing

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Gliederung	1
2	Complex Event Processing	3
2.1	Ereignisse	3
2.2	Definition	3
2.3	Basisarchitektur	5
2.4	Event Processing Networks	6
2.5	Aktuelle Forschungen	8
3	Zusammenfassung und Ausblick	10
3.1	Zusammenfassung	10
3.2	Ausblick	10

1 Einleitung

1.1 Motivation

Ereignisse spielen in vielen Bereichen der Interaktion eine wichtige Rolle. Insbesondere stellt das menschliche Handeln einen Bereich dar, in dem wir die Bedeutung von Ereignissen wahrnehmen. Beim Eintreten bestimmter Ereignisse, z.B. dem Klingeln eines Telefons oder dem Erhalt einer E-Mail, unterbrechen wir unsere gerade ausgeführte Tätigkeit und reagieren auf das wahrgenommene Ereignis. Im Falle des klingelnden Telefons könnte es sein, dass wir uns zum Telefon begeben und den Anruf entgegen nehmen oder uns dafür entscheiden das Telefon zu ignorieren. In beiden Fällen wird auf das Ereignis des Telefonklingelns reagiert.

Auch in Unternehmensabläufen (Geschäftsprozessen) spielen Ereignisse eine wichtige Rolle. Storniert ein Kunde eine Bestellung, muss darauf reagiert werden, indem ggf. bereits in Rechnung gestellte Beträge zurückgebucht werden oder die Versandabteilung angewiesen wird, den Versand der Waren nicht vorzunehmen.

Aus einer eher technischen Perspektive stellen die von Sensoren gelieferten Daten Ereignisse dar. Diese müssen durch Computersysteme in immer größerer Zahl gespeichert, analysiert und in Reaktionen umgesetzt werden.

Complex Event Processing (CEP) kann dabei helfen, die Komplexität von Abläufen und großen Datenaufkommen zu reduzieren, indem der Anwendungsbereich durch Ereignisse modelliert wird. CEP bietet Methoden und Werkzeuge an, um Ereignisse nahezu in Echtzeit zu erkennen und auf sie reagieren zu können.

Ziel dieser Ausarbeitung ist es, die Grundlagen und aktuellen Entwicklungen im Bereich des CEP vorzustellen. Sie soll eine Basis schaffen, um in zukünftigen Veranstaltungen¹ die Eignung von CEP zur dynamischen Laststeuerung (Demand Side Management) in Smart Grids untersuchen zu können.

1.2 Gliederung

Im Anschluss an diese Einleitung betrachtet Kapitel 2 die Technologie des CEP. Hierzu wird zuerst in Abschnitt 2.1 erläutert, was Ereignisse im Kontext des CEP sind, um im Anschluss das CEP zu erklären.

¹Gemeint sind die Veranstaltung im Master Studiengang Informatik an der HAW Hamburg ab dem zweiten Semester.

Danach wird im Abschnitt 2.3 die Basisarchitektur aus Sicht des Software Engineerings betrachtet. Es wird dabei auf die Softwarearchitekturprinzipien eingegangen, welche sich hinter der Basisarchitektur verbergen, und erklärt, welche Vorteile und Möglichkeiten sie eröffnen. In Abschnitt 2.4 wird dargestellt, wie die Funktionalität der Ereignisverarbeitung mittels Event Processing Networks beschrieben werden kann. Dabei wird besonderes Augenmerk auf die Event Processing Agents gelegt. Diese sind ein Bauteil des Event Processing Networks und können zur Erweiterung des Event Processing Networks um neue Funktionalität zu gewinnen genutzt werden. Anschließend beschreibt Abschnitt 2.5 eine Auswahl an aktuellen Forschungsthemen, die sich mit der Erweiterung des CEP beschäftigen.

Schlussendlich werden in Kapitel 3 die Ergebnisse dieser Ausarbeitung zusammengefasst und ein Ausblick auf mögliche Projekte gegeben.

2 Complex Event Processing

2.1 Ereignisse

Bei Ereignissen im Kontext des Complex Event Processing handelt es sich um Vorkommnisse die durch ein Computersystem beobachtet werden können. Diese Vorkommnisse können sowohl direkt innerhalb, also auch außerhalb des Systems beobachtet werden. Für die Beobachtung von Zustandsänderungen außerhalb des Systems werden Sensoren eingesetzt. (Mühl u. a., 2006; Etzion und Niblett, 2010). Zumeist handelt es sich bei den Vorkommnissen um die Änderung eines Zustandes. Komplexe Ereignisse (Complex Events) repräsentieren Ereignisse einer höheren Abstraktionsebene und bauen auf in Beziehung stehenden Ereignissen mit geringerem Abstraktionsgrad auf (vgl. Abbildung 2.1). Sie werden auch abstrakte oder abgeleitete Ereignisse genannt. Komplexe Ereignisse ermöglichen es, unterschiedliche Sichten auf eine Menge von Ereignissen zu definieren und stellen somit ein Mittel dar, um große Ereignismengen oder -ströme zu handhaben (Bruns und Dunkel, 2010; Etzion und Niblett, 2010).

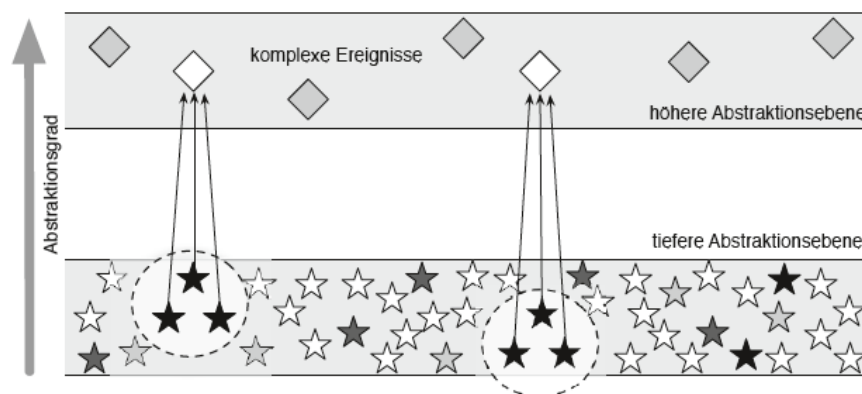


Abbildung 2.1: Complex Events (Bruns und Dunkel, 2010)

2.2 Definition

Der Begriff des Complex Event Processing (CEP) beschreibt Methoden und Werkzeuge zur parallelen und dynamischen Verarbeitung von Ereignissen. Sie ermöglichen es Serien von Ereignissen auf miteinander in Beziehung stehende Ereignisse zu untersuchen und so komplexe Ereignisse zu entdecken. Ereignisse können dabei auf unterschiedliche Arten miteinander in Beziehung stehen:

- Kausale Beziehungen
- Temporale Beziehungen
- Räumliche Beziehungen

Neben den eben genannten Beziehungsarten sind auch weitere denkbar. Die Beziehungen werden dabei als Muster von Ereignissen ausgedrückt, so dass man sagen kann, dass nach Ereignismustern in einem Strom von Ereignissen gesucht wird. CEP ermöglicht es, die Suche in nahezu Echtzeit (sehr schnell, garantiert aber keine festen zeitlichen Schranken) auszuführen (Luckham, 2001; Etzion und Niblett, 2010). Zusätzlich bietet CEP die Möglichkeit anzugeben, wie auf ein oder mehrere gefundene Ereignisse reagiert werden soll. Hierzu können z.B. Regeln definiert werden, die beschreiben, was das System in einem solchen Fall tun soll. Diese sogenannten *Production Rules* besitzen hierzu eine Bedingung und eine Liste von Aktionen, die beim Eintreten der Bedingung ausgeführt werden. Sie entsprechen also dem if-then-Konstrukt aktueller Programmiersprachen.

Durch den Einsatz von CEP können Computersysteme entwickelt werden, welche sehr große Mengen von Ereignissen, bis zu mehreren Millionen von Ereignissen pro Sekunde, verarbeiten können. Systeme entwickeln sich dabei von responsive (Benutzereingabe verarbeitenden) hin zu reactive (wahrnehmenden und reagierenden) Systemen¹ (Etzion und Niblett, 2010; Engel und Etzion, 2011). Die darauffolgende letzte Stufe stellt das Proactive Computing dar. Ein proaktives System erkennt, dass eine unerwünschte Situation in Zukunft eintreten könnte und ergreift daraufhin Maßnahmen um diese zu vermeiden. Abbildung 2.2 zeigt die Entwicklung vom Responsive hin zum Proactive Computing.

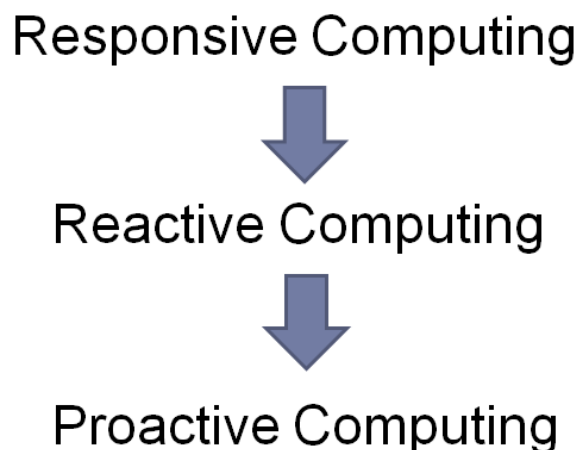


Abbildung 2.2: Evolutionsschritte im Complex Event Processing

¹Einen tiefergehenden Einblick vermittelt die Diskussion zwischen David Luckham und Opher Etzion Responsive, Reactive and Proactive computing, difference? (zuletzt abgerufen am: 31.01.2012)

2.3 Basisarchitektur

Die Basisarchitektur eines CEP besteht in der Regel aus den folgenden Komponenten und ist in Abbildung 2.3 dargestellt:

- Event Producers
- Event Consumers
- Event Processing Logic

Event Producer sind für das Erkennen von Ereignissen und das Erzeugen von Ereignisrepräsentationen verantwortlich. Die Event Processing Logic analysiert die erkannten Ereignisse, und abstrahiert sie zu komplexen Ereignissen. Event Consumer können von unterschiedlicher Natur sein. Sie können Aktoren sein, die auf die Ereignisse reagieren und eine Aktion in der physischen Welt bewirken. Sie können aber auch nur Ereignisse speichern oder anzeigen. Allen Event Consumern ist gemein, dass sie die Ereignisse von den Event Producern oder der Event Processing Logic aufnehmen und sie verarbeiten.

Die Komponenten werden über einen Mediator miteinander verbunden. Der Mediator ermöglicht die Kommunikation der Komponenten untereinander und arbeitet oft nach dem Publisher/Subscriber-Prinzip. Realisiert wird er durch eine Message Oriented Middleware (MOM), so dass Komponenten miteinander über den Austausch von Nachrichten kommunizieren. Durch den Einsatz einer MOM kennen sich die Komponenten untereinander nicht und teilen lediglich dem Mediator mit, dass sie bestimmte Nachrichten (z.B. von einer Komponente oder zu einem bestimmten Thema) erhalten möchten. Selber senden sie ihre Nachrichten auch direkt an den Mediator.

Auf diese Weise realisiert die Basisarchitektur in Verbindung mit einem Mediator die folgenden

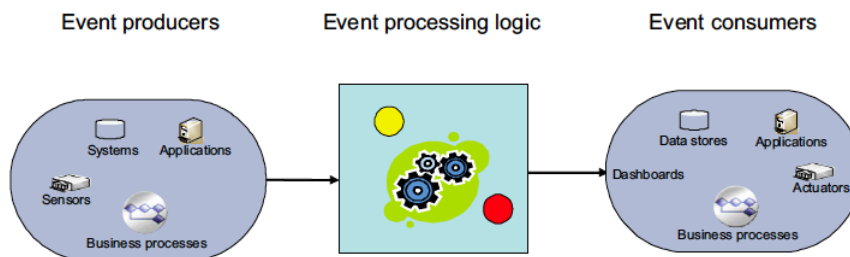


Abbildung 2.3: Basisarchitektur Complex Event Processing nach Bruns und Dunkel (2010)

wünschenswerten Softwarearchitekturprinzipien:

- Lose Kopplung
- Asynchrone Verarbeitung

Dank dieser Prinzipien erhält man ein flexibles und fehlertolerantes System. Letzteres bringt vor allem dem Vorteil mit sich, dass der Ausfall einzelner Komponenten nicht den Ausfall des Gesamtsystems nach sich zieht.

2.4 Event Processing Networks

In der gerade vorgestellten Basisarchitektur repräsentiert die Event Processing Logic die Verarbeitungslogik zur Erkennung von Ereignismustern und der Erzeugung von komplexen Ereignissen. Die Event Processing Logic ist dabei kein Monolith sondern besteht aus einzelnen Teilkomponenten. Diese Teilkomponenten sind zu einem Netzwerk bzw. Graphen zusammen geschlossen. Dieser wird Event Processing Network (EPN) genannt (Perrochon u. a., 1999; Luckham, 2001; Etzion und Niblett, 2010).

EPNs sind eine Methode zur Visualisierung und Modellierung der Funktionalität für die Ereignisverarbeitung innerhalb eines Systems und repräsentieren die gesamte Funktionalität des Complex Event Processing. Abbildung 2.4 zeigt ein Beispiel für ein EPN. Die verwendete Notation wurde in Etzion und Niblett (2010) vorgestellt.

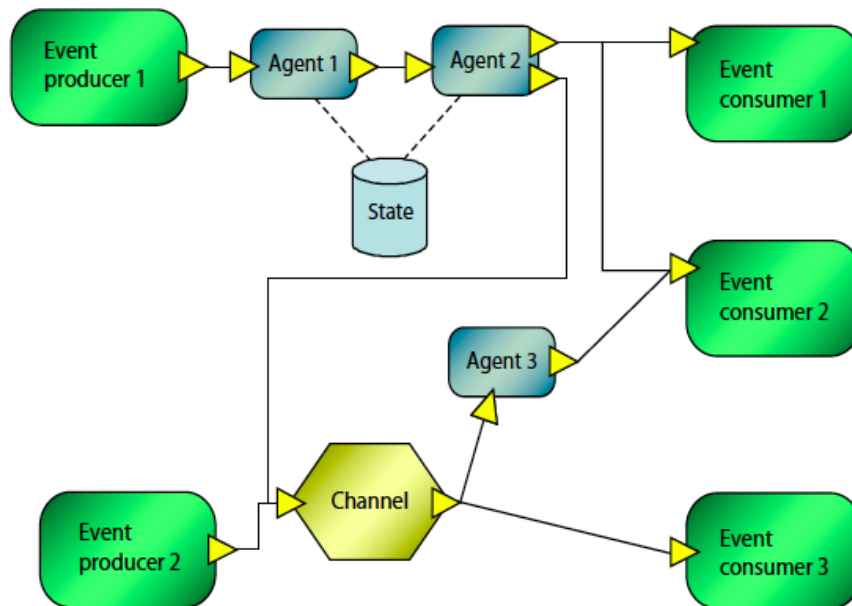


Abbildung 2.4: Beispiel Event Processing Network (Etzion und Niblett, 2010)

Wie auch Abbildung 2.4 entnommen werden kann, besteht ein Event Processing Netzwerk aus fünf verschiedenen Komponenten:

- Event Producer
- Event Consumer
- Kanäle
- Event Processing Agents
- Global State Elements

Event Producer und Event Consumer sind bereits aus der Basisarchitektur bekannt und repräsentieren diese Komponenten innerhalb des EPN.

Kanäle verbinden die verschiedenen Bestandteile des EPN und sind für die Weiterleitung von Ereignissen zwischen den Komponenten verantwortlich. Sie gibt es in zwei Ausführungen, als implizite und explizite Kanäle.

Implizite Kanäle werden durch die Kanten des Graphen symbolisiert, wohingegen Sechsecke explizite Kanäle darstellen. Explizite Kanäle unterscheiden sich von den impliziten durch ihren Funktionsumfang. Explizite Kanäle modellieren zusätzliche Funktionalität neben dem Weiterleiten von Ereignissen, z.B. in Abbildung 2.4 das Duplizieren von Ereignissen.

Event Processing Agents (EPA) besitzen unterschiedliche Aufgaben und es existieren daher unterschiedliche Typen von Agenten. Alle Agenten funktionieren nach dem gleichen Prinzip. Sie nehmen ein oder mehrere Ereignisse auf, analysieren und verarbeiten sie und erzeugen als Ausgabe ein oder mehrere neue Ereignisse. So gibt es Agenten die ankommende Ereignisse auf bestimmte Eigenschaften hin filtern oder die eigentliche Mustererkennung ausführen. Eine Hierarchie der gängigsten EPAs zeigt die Übersicht in Abbildung 2.5.

Wie der Übersicht in Abbildung 2.5 entnommen werden kann gibt es drei Hauptkategorien von EPAs: Filter, Transformation und Pattern Detect Agents. Filter Agents überprüfen Ereignisse auf vorher festgelegte Eigenschaften. Sie geben nur Ereignisse weiter, die diesen Kriterien entsprechen. Die Anderen werden verworfen.

Pattern Detect Agents sind für die Erkennung von Ereignismustern in Ereignisströmen und die Erzeugung der korrespondierenden komplexen Ereignisse verantwortlich. Sie gehören zu den zustandsbehafteten Agenten und nutzen ein Sliding Window zur Durchführung der Mustererkennung. Die Transformation Agents stellen die letzte Kategorie von EPAs dar. Ihre Aufgabe besteht in der Umwandlung von ankommenden Ereignissen in Ereignisse eines anderen Ereignistyps. Dieses kann durch das Übersetzen von einem Ereignistyp in einen Anderen, durch das Zusammenfassen mehrere Ereignisse, das Teilen eines Ereignisses und das Zusammenfassen mehrere Ereignisse unter der Berücksichtigung bestimmter Kriterien erfolgen. Diese Funktionsweisen werden durch die Unterkategorien Translate, Aggregate, Split und Compose repräsentiert. Dabei kann die Übersetzung auf zwei zusätzliche Weisen erfolgen, indem Ereignisse mit zusätzlichen Attributen angereichert werden (Enrich Agent) oder die Attribute eines Ereignisses werden auf zwei neue Ereignisse aufgeteilt (Project Agent). Für eine genauere Erläuterung der einzelnen Typen sei hier auf Etzion und Niblett (2010) verwiesen.

EPAs sind im Grunde zustandslos. Sie nehmen wie beschrieben jeweils nur Ereignisse entgegen verarbeiten sie und erzeugen ggf. neue Ereignisse. Global State Elements erweitern EPAs um Zustände (vgl. Abbildung 2.4). Ein Global State Element stellt im Allgemeinen einen Speicher für Ereignisse bereit. Das kann im einfachsten Fall eine Datenbank sein. Eine andere häufig genutzte Alternative

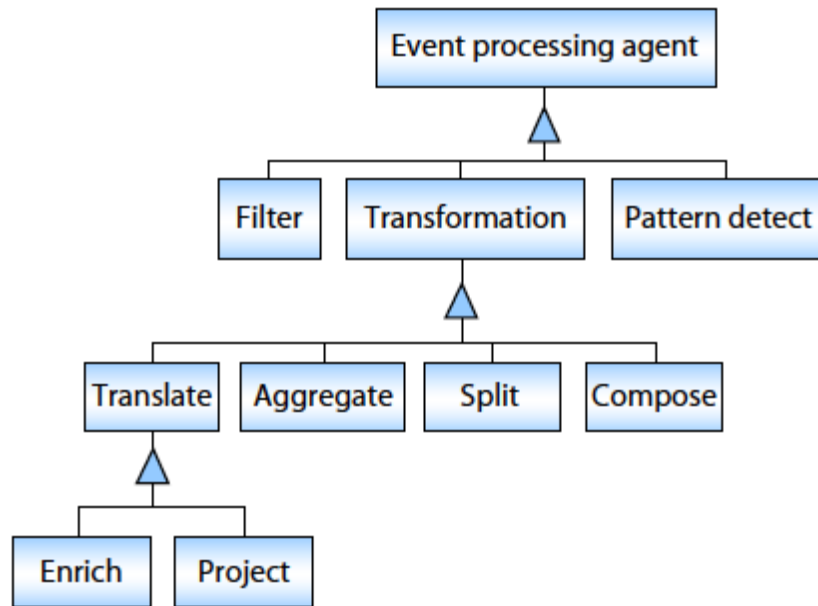


Abbildung 2.5: Event Processing Agent Hierarchie (Etzion und Niblett, 2010)

stellt ein Sliding Window über Ereignisse dar. Dieses ermöglicht es den Agenten, besonders den Pattern Detect Agents, kausale bzw. temporale Zusammenhänge zwischen Ereignissen zu erfassen.

2.5 Aktuelle Forschungen

Ein wichtiger Bereich in dem noch Forschungsbedarf besteht, ist die Entwicklung anerkannter Standards, da die Anzahl an Produkte, welche Funktionalität aus dem Bereich des CEP nutzen, immer mehr zunimmt. Die Problematik daran ist, dass aktuelle Produkte alle eigene Ereignisrepräsentationen, Abfragesprachen für Ereignismuster und Architekturen verwenden. Hier gilt es Standards zu schaffen, um einen Informationsaustausch und eine Interoperabilität zwischen Produkten unterschiedlicher Hersteller zu ermöglichen (Etzion und Niblett, 2010, Seite 326-327). Hierbei stellt die durch Cureen und Windley (2012) vorgestellte Evented API Specification einen ersten Schritt in diese Richtung dar.

Semantic Complex Event Processing (SCEP) versucht die Techniken des Semantic Web, wie Ontologien etc., zu nutzen, um das Wissen über den Fachbereich bei der Interpretation und der Reaktion auf Ereignisse ausnutzen zu können. Beziehungen zwischen Ereignissen lassen sich durch die genau Beschreibung des Fachbereichs in einer Ontologie einfacher identifizieren. Sie ermöglicht eine bessere Reaktion im Kontext des Fachbereichs (Teymourian u. a., 2009; Teymourian und Paschke, 2009; Zhou u. a., 2012).

Einen weiteren Forschungsschwerpunkt stellt Proactive Computing dar. Proaktive Computing stellt einen weiteren Forschungsschwerpunkt dar. In diesem Bereich möchte man ein Proactive Complex

Event Processing entwickeln und einen neuen Typ von Event Processing Agent kreieren. Dieser soll nicht nur in der Lage sein, auf Ereignisse zu reagieren, sondern soll auch unerwünschte Situationen voraussehen können. Hiermit soll der Agent in die Lage versetzt werden, aktiv eingreifen zu können, um das Eintreten der Situation zu vermeiden oder ihre Folgen zu mildern. Dazu analysiert der Agent die Ereignisströme mit Methoden der Predictive Analytics oder des Contengency Planing und leitet damit aus den Ereignissen Prognosen ab. Mit Hilfe der Prognosen lassen sich dann Maßnahmen und Pläne entwickeln (Etzion und Niblett, 2010; Engel und Etzion, 2011).

3 Zusammenfassung und Ausblick

3.1 Zusammenfassung

Complex Event Processing (CEP) erlaubt es neuartige, flexible und robuste Computersysteme zu entwickeln. Durch die Konzentration auf Ereignisse als Modellierungsansatz und eine Architektur, die wichtige Prinzipien der Softwareentwicklung unterstützt und realisiert, schafft CEP den Schritt vom Responsive hin zum Reactive Computing.

Aktuelle Forschungen im Bereich des CEP versuchen das Potential von CEP durch die Verbindung mit anderen Technologien wie Semantic Web und Business Intelligence voll auszuschöpfen. Es soll so der Schritt von reaktiven Computersystemen hin zu proaktiven (vorausschauenden) Systemen, und damit dem Proactive Computing, gegangen werden.

3.2 Ausblick

Mit den in dieser Ausarbeitung gewonnen Erkenntnissen soll in zukünftigen Veranstaltungen untersucht werden, inwieweit sich CEP zur Steuerung von Smart Grids einsetzen lässt. Angebot und Nachfrage müssen in Elektrizitätsnetzen immer ausgeglichen sein. Sowohl Überlast (Nachfrage übersteigt das Angebot) als auch Unterlast (Angebot übersteigt die Nachfrage) können zu Störungen in der Stromversorgung, bis hin zu großflächigen Stromausfällen, führen. Das Lastmanagement stellt eine große Herausforderung für Energieversorgungsunternehmen (EVU), besonders im Hinblick auf die Integration der erneuerbaren Energien, dar.

In Smart Grids erfolgt das Lastmanagement zukünftig zusätzlich durch intelligente Endgeräte und intelligente Stromzähler. Dadurch verringert sich der Aufwand für die EVUs. Demand Side Management (DSM), das gezielte Abschalten von Verbrauchern und Hinzuschalten von Stromerzeugern, soll in Smart Grids für das Gleichgewicht zwischen Stromangebot und -nachfrage sorgen.

In einem möglichen nächsten Schritt soll untersucht werden, ob sich mittels CEP ein verlässliches DSM realisieren lässt. Besonderes Augenmerk soll hierbei auf die, in Abschnitt 2.5 beschriebenen Themen, Proactive und Semantic Complex Event Processing gelegt werden. Sie besitzen das Potential Probleme beim Lastausgleich frühzeitig feststellen zu können (vgl. (Zhou u. a., 2012)). So könnten schnell Gegenmaßnahmen durch das Smart Grid ergriffen und die Sicherheit der Stromversorgung sichergestellt werden.

Die Untersuchungen sollen sich vorrangig auf DSM in Micro-Grids konzentrieren. Micro-Grids sind Untereinheiten eines Smart Grids, welche einen Insel-Modus besitzen, in dem sie autonom

vom Rest des Smart Grids agieren. Micro-Grids entsprechen von ihrer Größe meist einer dörflichen Gemeinde. Die Energieerzeugung erfolgt dezentral innerhalb des Micro-Grids (Hatzigiorgiou u. a., 2007; Llarra u. a., 2009; Farhangi, 2010; Xue-song u. a., 2011). Auf Grund der geringeren Größe und der Möglichkeit zur autonomen Steuerung, soll ein noch zu spezifizierendes Micro-Grid als Szenario für die Untersuchungen von DSM in Verbindung mit CEP dienen. Durch diesen Ansatz kann die Komplexität der zu untersuchenden Problemstellung verringert werden.

Literaturverzeichnis

- [Bruns und Dunkel 2010] BRUNS, Ralf ; DUNKEL, Jürgen: *Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse*. Berlin : Springer, 2010. – ISBN 978-3-642-02438-2
- [Cureen und Windley 2012] CUREEN, Sam ; WINDLEY, Phil: *Evented APIs*. 2012. – URL <http://www.eventedapi.org/>. – Datum des letzten Zugriffs: 20.02.2012
- [Engel und Etzion 2011] ENGEL, Yagil ; ETZION, Opher: *Towards proactive event-driven computing*. New York, NY, USA : ACM, 2011 (DEBS '11). – 125–136 S. – URL <http://doi.acm.org/10.1145/2002259.2002279>. – ISBN 978-1-4503-0423-8
- [Etzion und Niblett 2010] ETZION, Opher ; NIBLETT, Peter: *Event Processing in Action*. 1st. Greenwich, CT, USA : Manning Publications Co., 2010. – ISBN 1935182218, 9781935182214
- [Farhangi 2010] FARHANGI, H: The Path of the Smart Grid. In: *Ieee Power And Energy Magazine* 8 (2010), Nr. 1, S. 18–28. – URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5357331>
- [Hatziargyriou u. a. 2007] HATZIARGYRIOU, Nikos ; ASANO, Hiroshi ; IRAVANI, Reza ; MARNAY, Chris: Microgrids: An Overview of Ongoing Research, Development, and Demonstration Projects. In: *IEEE Power and Energy Magazine* 5 Issue:4 (2007), S. 78 – 94
- [Llaria u. a. 2009] LLARIA, Alvaro ; CUREA, Octavian ; JIMENEZ, Jaime ; CAMBLONG, Haritza: Survey on Microgrids: analysis of technical limitations to carry out new solutions. In: *13th European Conference on Power Electronics and Applications EPE '09*, 2009, S. 1 – 8
- [Luckham 2001] LUCKHAM, David C.: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2001. – ISBN 0201727897
- [Mühl u. a. 2006] MÜHL, Gero ; FIEGE, Ludger ; PIETZUCH, Peter: *Distributed Event-Based Systems*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006. – ISBN 3540326510
- [Perrochon u. a. 1999] PERROCHON, Louis ; MANN, Walter ; KASRIEL, Stephane ; LUCKHAM, David C.: Event Mining with Event Processing Networks. In: *IN PROCEEDINGS OF THE THIRD PACIFIC-ASIA CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*, Springer, 1999, S. 474–478

- [Teymourian und Paschke 2009] TEYMOURIAN, Kia ; PASCHKE, Adrian: Towards semantic event processing. In: *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA : ACM, 2009 (DEBS '09), S. 29:1–29:2. – URL <http://doi.acm.org/10.1145/1619258.1619296>. – ISBN 978-1-60558-665-6
- [Teymourian u. a. 2009] TEYMOURIAN, Kia ; STREIBEL, Olga ; PASCHKE, Adrian ; ALNEMR, Rehab ; MEINEL, Christoph: Towards semantic event-driven systems. In: *Proceedings of the 3rd international conference on New technologies, mobility and security*. Piscataway, NJ, USA : IEEE Press, 2009 (NTMS'09), S. 347–352. – URL <http://dl.acm.org/citation.cfm?id=1790343.1790410>. – ISBN 978-1-4244-4765-7
- [Xue-song u. a. 2011] XUE-SONG, Zhou ; LI-QIANG, Cui ; YOU-JIE, Ma: Research on Control of Micro Grid. In: *Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2011, S. 1129 – 1132
- [Zhou u. a. 2012] ZHOU, Qunzhi ; NATARAJAN, Sreedhar ; SIMMHAN, Yogesh ; PRASANNA, Viktor: Semantic Information Modeling for Emerging Applications in Smart Grid,. In: *9th IEEE International Conference on Information Technology: Next Generations*, 2012