

Konzeption und Implementierung eines raumbezogenen CEP Services mit der CEP (Complex Event Processing) Engine NEsper

Peter Kronsteiner¹, Katharina Meissner¹, Roland Grillmayer¹ und Manfred Mittlböck²

¹Fachhochschule Wiener Neustadt · peter.kronsteiner@fhwn.ac.at

²RSA iSPACE, Salzburg

Short paper

Kurzfassung

Diese Arbeit beschreibt die Konzeption & Implementierung eines raumbezogenen CEP Services (Complex Event Processing Services) mit der CEP Engine NEsper. Das Projekt war Teil des 5. Semesters im Bachelorstudiengang Informatik an der FH Wiener Neustadt und wurde im September 2014 von der außeruniversitären Forschungseinrichtung iSPACE (der Research Studios Austria Forschungsgesellschaft mbH) in Auftrag gegeben.

Ziel des Projektes war es, große Mengen von Geodaten und Geoinformationen aus verschiedenen Quellen zu sammeln und diese ereignisgesteuert, in naher Echtzeit verarbeiten und analysieren zu können. Zusätzliches Ziel war es, auf die ausgelösten Events entsprechend zu reagieren (z. B. Benachrichtigung eines Benutzers via eMail). Die im Rahmen des Projektes entwickelte Systemarchitektur basiert auf dem Open Source Projekt NEsper, welches unter der GNU General Public License (GPL) verfügbar ist. Für die ereignisgesteuerte Verarbeitung und Analyse von Geodaten galt es, als thematischen Schwerpunkt im Rahmen des Projektes eine erweiternde Geo-Komponente für NEsper zu entwickeln.

Die zu entwickelnde Geokomponente hat zum Ziel, regelbasiert auch auf räumliche Ereignisse reagieren zu können, wenn zum Beispiel ein Objekt eine bestimmte geographische Region betritt oder verlässt (= Geofence Regel). Die für das Regelwerk benötigten Geofences werden via OGC Web Feature Service als GML 3.x Polygonen zur Verfügung gestellt. Die Ereignisse des raumbezogenen CEP-Services werden durch NEsper Statements (Regeln) gesteuert. Diese Regeln und die definierten Geofences können dynamisch zur Anwendungslaufzeit konfiguriert und verändert werden.

Schlagwörter:

Complex Event Processing; CEP; NEsper; Geo-Events

1 Einleitung

In den letzten Jahren gewinnt der Begriff „Complex Event Processing“ (im Folgenden auch CEP) in der Welt der Informatik immer mehr an Bedeutung. Dies liegt einerseits an den zunehmend komplexeren Zusammenhängen von Ereignissen in der IT und andererseits auch an den Massen an Daten die tagtäglich produziert und ausgewertet werden. Um diese

komplexen Zusammenhänge auszuwerten, bedarf es effizienterer Technologien und Infrastrukturen. Eine dieser Technologien ist CEP.

In dieser Arbeit wird CEP und dessen Adaptierung für die Geoinformatik anhand des durchgeführten Projektes beschrieben. Im Folgenden wird zuerst auf den Begriff „Complex Event Processing“ im Allgemeinen eingegangen. Des Weiteren wird die entwickelte Geo-Komponente erläutert. Abschließend folgt ein kurzer Ausblick auf weitere Anwendungsbeispiele für die entwickelte Software.

2 Complex Event Processing

CEP ist ein ereignisorientiertes und auf Echtzeit basierender IT-Konzeptbaustein. In der IT auftretende Ereignisse können als Events in CEP verwendet werden. Dabei werden Events mit einer niedrigen Aussagekraft von CEP miteinander verknüpft und zu komplexen Events zusammengefasst, die eine höhere Aussagekraft besitzen. Damit ist es möglich, von auftretenden einzelnen Events auf größere, komplexe Zusammenhänge zu schließen. Die komplexen Events können dann automatisierte Prozesse anstoßen, Benutzer informieren oder alarmieren. Der Einsatz von CEP ist in verschiedenen IT-Architekturen realisierbar. CEP kann seine Informationen in Form von Events aus Verteilten Systemen, aus über Middleware verbundene Anwendungen, aber auch von einzelnen Systemen wie Feuermeldern oder RFID Chips bekommen (RIZVI & JEFFERY 2005).

Der Vorteil von CEP liegt in der Echtzeitverarbeitung von vielen Informationen. Anwendungen mit hohem Datendurchsatz können ein großes Volumen an Daten (zwischen 1.000 und 100.000 Nachrichten pro Sekunde) verarbeiten (ESPER TEAM AND ESPERTECH INC 2014).

Somit stellt CEP eine Methode dar, um große Mengen von Informationen aus mehreren Datenströmen zu sammeln, zu verarbeiten und anschließend zu analysieren. Die Informationen werden fortlaufend in Echtzeit verarbeitet und in Abhängigkeit von komplexen, aktuell auftretenden Ereignissen (Events) ausgewertet. Diese Events treten in einem Zeitintervall mehrfach, unbestimmt oft und unregelmäßig auf.

Zu den häufigsten Anwendungsgebieten von CEP-Lösungen gehören Finanzanwendungen (Risikomanagement, Erkennung von Betrug, algorithmisches Handeln), Netzwerk- und Anwendungsmonitoring (Angriffserkennung), Sensornetzanwendungen (RFID-Erkennung, Flugverkehr, Steuerung und Überwachung von Produktionsketten), Business process management and automation (GROHE, SCHLAMEUSS & SOMMER 2010).

Die oben angeführten Anwendungsgebiete verdeutlichen die breite und bereits etablierte Nutzung von CEP in den unterschiedlichsten Domänen. Im durchgeführten Projekt wurde als Grundbaustein die Open Source CEP Engine NESper genutzt. NESper basiert auf CLR (Common Language Runtime) und wurde für CEP und ESP (Event Stream Processing) Systeme entworfen. NESper verfügt über keine direkte Schnittstelle für raumzeitliche Ereignisse. Ziel dieses Projekts war es daher, eine Komponente zu entwickeln, die Geo-Funktionalitäten und CEP kombiniert.

3 Projekt-Komponenten

Die entwickelte Projektarchitektur besteht aus drei Komponenten (vgl. Abb. 1). Einer ASP.NET Webapplikation, welche das zentrale Frontend darstellt und mit der die Event-Regeln zur Laufzeit erstellt, editiert und gelöscht werden können. Die Regelergebnisse und die aktuelle Position des Objekts werden dynamisch auf der Hauptseite dargestellt. Zusätzlich werden der aktuelle Standort des Objekts (Marker) und die konfigurierten „Geofences“ mit einer auf Leaflet (www.leaflet.org) basierenden WebMap visualisiert. Als Backend dient eine Microsoft SQL-Datenbank in der die Regeln, Geofences und die Regelergebnisse verwaltet werden. Die dritte Komponente stellt den Server dar, der die gesamte Business Logik abbildet. Am Server werden die Geoprozesse ereignisgesteuert verarbeitet und analysiert. Die Daten werden über HTTP-Post Requests an den Server gesendet. In Abbildung 1 ist durch die Pfeile der Datenverkehr zwischen den einzelnen Komponenten zu erkennen.

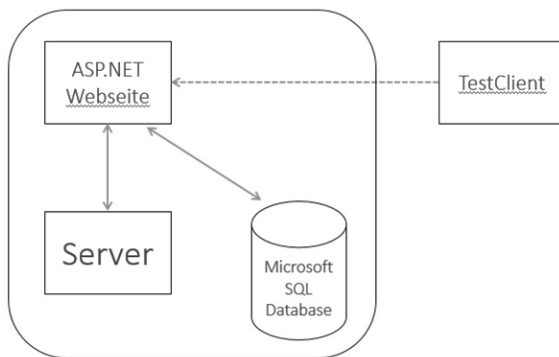


Abb. 1:
Projektkomponenten

3.1 Geofences

Wie bereits erwähnt, können dem Service Geofences zur Verfügung gestellt werden. Ein Geofence ist in diesem Fall ein mittels einem GML 3.x Polygon definierter räumlicher Bereich, der über ein Web Feature Service zur Verfügung gestellt wird. In Abbildung 2 ist als Beispiel der Geofence als räumliche Abgrenzung des 1. Bezirks in Wien dargestellt.

Sobald mindestens ein Geofence in der Datenbank vorhanden ist, wird durch CEP Engine mithilfe eines „Point-In-Polygon“-Tests (im Folgenden auch PIP) geprüft, ob sich ein beobachtetes Objekt innerhalb dieses räumlichen Bereiches befindet.

3.2 Webanwendung

Der Anwender hat die Möglichkeit, das Service rein über eine, in ASP.NET programmierte, Webseite zu steuern. Der Server und die Datenbank laufen im Hintergrund der Anwendung und sind für den Benutzer transparent.

Da das geplante Service vielseitig einsetzbar sein sollte, wurde die Geofence-Server Komponente als Klassenbibliothek entwickelt. Dadurch hat der Entwickler die Möglichkeit, alle Funktionen, z. B. PIP, in andere Projekte zu integrieren. Die „Geofencing“-Library ist das

Hauptelement des Projekts und bildet die Applikationslogik ab. Sie kann Daten empfangen und gibt diese an den „Nesper Event Processor“ weiter. Neben den funktionalen Baustein-konfigurationen werden auch die zusätzlichen Parameter wie Benutzereinstellungen (Event-Regeln & Geofences) am Server zwischengespeichert bzw. bestehende Konfigurationen bei Neustart der Applikation aus der Datenbank geladen. Der Benutzer kann die Event-Regeln und Geofences in der Webapplikation konfigurieren und anschließend werden diese in der Datenbank persistiert.

Einige Regelbeispiele sind:

- Regel 1: Inside Geofence – Objekt A betritt Geofence X.
- Regel 2: Outside Geofence – Objekt A verlässt Geofence X.
- Regel 3: Geschwindigkeitsüberschreitung in Geofence X – Objekt A überschreitet die Geschwindigkeitsbeschränkung von 100 km/h im Geofence X.

Aufgrund der definierten Regeln entscheidet die „Event Processor“ Komponente, die das Geofence Modul mit der Nesper CEP Engine verknüpft, welche definierten Regeln zur Anwendung kommen, führt diese aus und gibt die entsprechenden Benachrichtigungen aus. Diese Benachrichtigungen können auf unterschiedliche Weise an den User weitergegeben werden. Im implementierten Projekt werden die Ereignisse in der ASP.NET Webanwendung im Bereich „Nesper Event Log“ visualisiert (Abb. 2). Zusätzlich wird die Position des Objekts, welches die Benachrichtigung ausgelöst hat, auf einer WebMap dargestellt. Alle Informationen werden dabei dynamisch zur Laufzeit aktualisiert. Somit kann der Benutzer das aktuelle Objekt auf der WebMap „nachverfolgen“. Zusätzlich werden auch weitere Daten, zum Beispiel Geschwindigkeit, Position und Objekt ID, laufend mit aktualisiert.

Nesper Event Log:

```

11.02.2015 23:33:33: Inside Geofence: Wien 1, Bezirk
11.02.2015 23:33:36: Received json latitude: 48,2125945400808 / longitude: 16,3835854402155
11.02.2015 23:33:37: ===== NESPER EVENT: Speed too high: ID233 =====
11.02.2015 23:33:37: Position: 48,2125945400808 / 16,3835854402155 Speed: 125 kmph EngineSpeed: 1925 mps
11.02.2015 23:33:37: Outside Geofence: 'Wien 1, Bezirk'
11.02.2015 23:33:40: Received json latitude: 48,2054477056514 / longitude: 16,3645272150703
11.02.2015 23:33:40: ===== NESPER EVENT: Engine Speed too high (over 2200): ID231 =====
11.02.2015 23:33:40: Position: 48,2054477056514 / 16,3645272150703 Speed: 54 kmph EngineSpeed: 2213 mps
11.02.2015 23:33:40: Inside Geofence: Wien 1, Bezirk
11.02.2015 23:33:43: Received json latitude: 48,209063847777 / longitude: 16,3741702607387
11.02.2015 23:33:44: ===== NESPER EVENT: Engine Speed too high (over 2200): ID232 =====
11.02.2015 23:33:44: Position: 48,209063847777 / 16,3741702607387 Speed: 90 kmph EngineSpeed: 2670 mps
11.02.2015 23:33:44: Inside Geofence: Wien 1, Bezirk

```

Map Position:



Abb. 2: Webapplikation Hauptseite

Das Service verfügt auch über die Möglichkeit, alle Ereignisse per E-Mail oder Twitter an beliebige Personen zu senden. Alle ausgelösten Ereignisse werden zusätzlich in einer Datenbank gespeichert. Die Datenbank kann daher für weitere Auswertungen, insbesondere für Analysen über längere Zeiträume, verwendet werden.

3.3 Funktionsweise

Damit Ereignisse ausgelöst werden können, muss ein Client GeoJson-Daten an das Service senden. Somit muss der Client über eine Komponente verfügen, welches Signale bzw. Daten im GeoJson-Format verschicken kann. In den GeoJson-Daten werden Informationen, wie zum Beispiel die aktuelle Position, Geschwindigkeit und Motordrehzahl, übermittelt. Der Server konvertiert die Daten anschließend in ein analysierbares Format.

Im nächsten Schritt wird durch den implementierten „Point-In-Polygon Test“ geprüft, ob sich das Objekt (Client) innerhalb eines definierten räumlichen Bereiches (Geofence) befindet. Die PIP-Auswertung und alle weiteren konvertierten Informationen werden anschließend automatisch an die NESper CEP Engine weitergeleitet. Diese bestimmt anhand der konfigurierten Regeln, welche Werte auf eine Regel zutreffen und erstellt eine neue Nachricht im NESper Event Log. Zusätzlich können verschiedene Services (E-Mail, Twitter, GeoJSON an GIST (Github), EsriJSON an ArcGIS Server) aktiviert werden, um die Regelergebnisse zu versenden.

4 Zusammenfassung und Ausblick

Zusammenfassend lässt sich sagen, dass sich Open Source NESper sehr gut für die Analyse und Auswertung raumbezogener Massendaten eignet. Die empfangenen Daten werden in Echtzeit verarbeitet und die Regelergebnisse sind unmittelbar verfügbar. Im Zuge des Projekts wurde nur ein geringer Teil der Funktionalitäten des NESper Frameworks genutzt. Dieses kann auch für weit komplexere Fragestellungen genutzt werden.

4.1 Erweiterungen und Einsatzgebiete

Bei der Entwicklung des Projekts wurde beachtet, dass die zu entwickelnde Service Komponente gut konfigurierbar, erweiterbar und damit vielseitig einsetzbar ist. Somit kann das entwickelte Service für verschiedene Anwendungsfälle genutzt werden. Durch die Erweiterung der Geo-Events in Verbindung mit weiteren komplexen Ereignissen kann die Anwendung für folgende Beispielszenarien genutzt werden:

- *Tracking*: Als Beispiel für B2B im Transportwesen und bei Einsatzkräften als automatisierte Event-Unterstützung in near-real-time Lagebildsystemen.
- *Smart Building/Smart Home*: Hier könnten Events durch Sensormessungen ausgelöst werden. Beispielsweise, falls die Außentemperatur unter 15 °C fällt und die Innentemperatur unter 20 °C fällt, wird die Heizung automatisch aktiviert und der Nutzer bekommt eine Benachrichtigung, wenn ein Fenster nicht geschlossen sein sollte.
- *Sicherheitsmanagement*: Z. B. flexible near-real-time Bewertung und Simulation der aktuellen Hochwassersituation bezogen auf die Data Stream Auswertungen von Wet-

termessungen Pegelabflussmessung, Bodenfeuchtesättigung und Nowcasting (Kurzfristvorhersage).

- *Qualitätssicherung*: Z. B. mit der Messwertevaluierung benachbarter vergleichbarer Sensormesswerte
- *B2C Messdatenstromanalyse*: Im Bereich der Erweiterung von B2C Stream Analysis Lösungen/Apps. Der Nutzer erhält eine Benachrichtigung, wenn die Tankfüllung des PKW's eine vordefinierte Grenze unterschreitet und ein Service Tankstellen in der aktuellen Geofence-Umgebung aufzeigt, die günstige Treibstoffpreise anbietet.

4.2 Die Innovation

Die Innovation des Projektes stellt die Einbindung einer Geo-Regel-Komponente für den CEP-Bereich dar. Der Nutzer des Services kann alle Event-Regeln zur Laufzeit ändern, ohne die Anwendung neu starten zu müssen. Erst durch die, im Rahmen dieses Projektes entwickelte, Geokomponente wird es möglich, den Aspekt Raum in CEP-Anwendungen zu berücksichtigen. Das entwickelte Service verdeutlicht, wie nützlich CEP in Verbindung mit räumlichen Daten sein kann. Die Komponente eröffnet neue CEP-Anwendungsfelder. Der in diesem Artikel vorgestellte Ansatz kann dazu beitragen, dass in Zukunft durch die Berücksichtigung des Raumbezugs in existierenden CEP-Anwendungen qualitativ bessere Entscheidungen getroffen werden können.

Literatur

ESPER TEAM AND ESPERTECH INC. (2014), Esper Reference.

http://www.espertech.com/esper/release-5.1.0/esper-reference/pdf/esper_reference.pdf (30.01.2015).

ESPERTECH (2014), NEsper for .NET.

<http://esper.codehaus.org/about/nesper/nesper.html> (30.01.2015).

GROHE, S., SCHLAMEUSS, C. & SOMMER, R. (2010), Performance Vergleich von CEP-Engines. Kap. 1.1 „Anwendungsgebiete von CEP-Engines“. Fachstudie 115, Universität Stuttgart, Institut für Parallele und Verteilte Systeme.

http://elib.uni-stuttgart.de/opus/volltexte/2010/5526/pdf/FACH_0115.pdf (30.01.2015).

RIZVI, S., JEFFERY, S. R., KRISHNAMURTHY, S., FRANKLIN, M. J., BURKHART, N., EDAKUNNI, A. & LIANG, L. (2005), Events on the edge. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD' 05 (New York, NY, USA), ACM, 885-887.