

Odysseus

Thomas Vogelgesang

04. 06. 2010

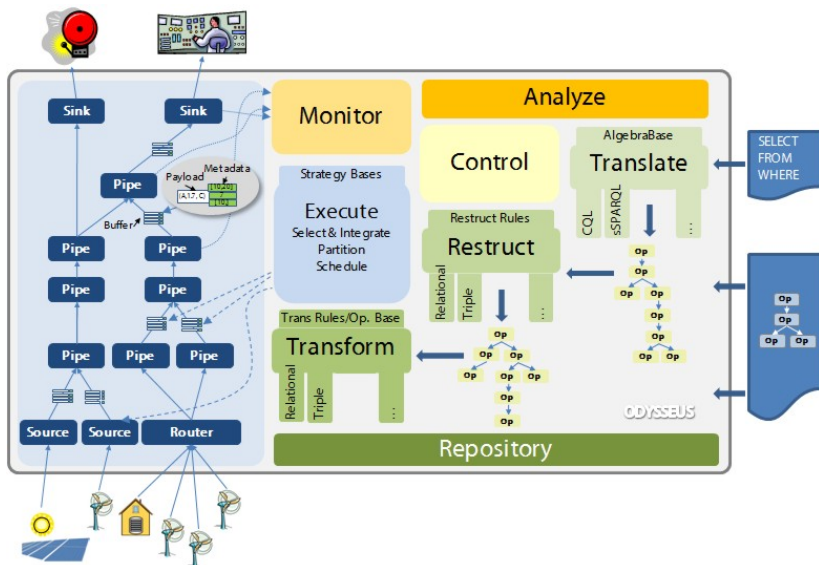
Übersicht

- ① Was ist Odysseus?
- ② Architektur von Odysseus
 - Übersetzung
 - Restrukturierung
 - Transformation
 - Physische Algebra-Basis
 - Ausführungsumgebung
 - Monitoring
- ③ OSGi als Modularisierungsframework
- ④ Verwendung von Odysseus
- ⑤ Zusammenfassung

Was ist Odysseus?

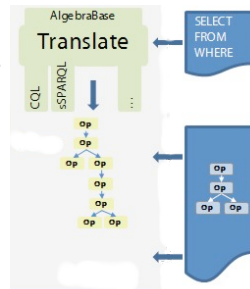
- Generisches Framework zur Erstellung von Datenstrommanagementsystemen (DSMS)
- Wird zur Zeit an der Universität Oldenburg entwickelt
- Deckt den gesamten DSMS-Workflow ab
- Grundlegende Eigenschaften
 - Leicht zu erweitern und anzupassen
 - Modularer Aufbau
 - Datenmodellunabhängig
 - Push-basierte Verarbeitung

Architektur von Odysseus



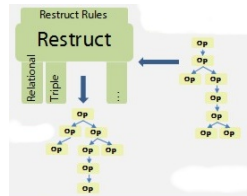
Übersetterskomponente

- Übersetzt Anfragen in logische Anfragepläne
- Anbindung verschiedener Parser und Anfragesprachen möglich
- Anfragepläne stellen Metadaten der Anfrage in Baumstruktur dar
- Logische Algebra-Basis kapselt erforderliche Verwaltungsstrukturen
- Operatoren-Hierarchie
 - Abstrakte Operatoren
 - Allgemeine (konkrete) Operatoren
 - Datenmodellspezifische Operatoren



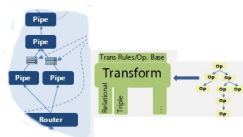
Restrukturisierungskomponente

- Optimierung übersetzter logischer Anfragepläne erforderlich
- Strukturiert den Anfrageplan um
- Verarbeitung erfolgt anhand einer erweiterbaren Regelmenge
- Verwendet Drools als Regelengine



Transformationskomponente

- Übersetzt logischen in physischen Anfrageplan
- Mehrere physische zu einem logischen Operator möglich
- Ergebnis kann mehrere Anfragepläne umfassen
 - Auswahl anhand von Kostenmodellen
- Regelbasierte Transformation
 - Regelengine: Drools

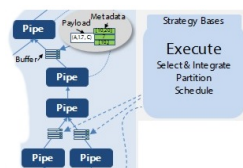


Physische Algebra-Basis

- Definiert abstrakte physische Operatoren
- Datenmodellunabhängiger Aufbau von physischen Anfrageplänen
 - Verknüpfung physischer Operatoren durch Publish-Subscribe-Mechanismus
- Konkrete physische Operatoren enthalten tatsächliche Datenverarbeitung
- Kapselt push-basierte Variante des Open-Next-Close-Protokolls
 - Erweiterungen können sich auf eigentliche Datenverarbeitung konzentrieren
- Metadaten können an Datenstromelemente angeheftet werden

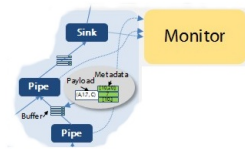
Ausführungsumgebung

- Verarbeitung erfolgt push-basiert
- Ausführung direkt verbundener Operatoren in einem Schritt
- Scheduler steuert Quelloperatoren und Puffer
- Parametrisierbar durch
 - Schedulingstrategien (ScS)
 - Pufferplatzierungsstrategien (PPS)



Monitoring

- Ermöglicht Überwachung physischer Anfragepläne zur Laufzeit
- Registrierung von EventListenern im Anfrageplan
 - Publish-Subscribe-Mechanismus
- Event-Verarbeitung durch Monitoring-Scheduler gesteuert
 - Parametrisierbar durch ScS



OSGi Service Platform 1/2

- Dynamisches Modulsystem in Java
- Spezifiziert von der OSGi Alliance
- Basiskomponente: OSGi Framework
- Module (Bundles)
 - Fachlich oder technisch zusammenhängende Klassen und Ressourcen
 - Können eigenständig im Framework (de)installiert werden
 - Für andere Bundles unsichtbar
 - Expliziter Import/Export von Paketen

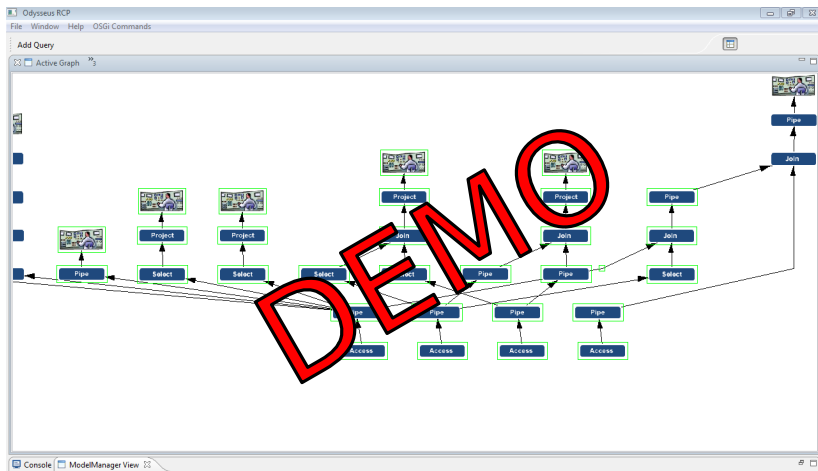
OSGi Service Platform 2/2

- Dienste (Services)
 - Unter Interface-Namen im Framework bekannt gemachtes Java-Objekt
 - Dynamische Anmeldung/Abmeldung bei Service Registry
 - Müssen keine Rücksicht auf abhängige Bundles nehmen
- Management Agent
 - Dynamische Verwaltung von Bundles zur Laufzeit
 - Ermöglicht ohne Anhalten/Neustarten des Frameworks
 - Installation/Deinstallation
 - Starten/Stoppen
 - Aktualisieren
 - von Bundles zur Laufzeit
 - Einfachste Umsetzung: Textbasierte Konsole

OSGi als Basis von Odysseus

- Teilkomponenten von Odysseus in Bundles gekapselt
- Abhängigkeiten zwischen Bundles über Services realisiert
 - Sehr geringe Kopplung zwischen Komponenten
 - Zur Laufzeit mehrere Implementierungen von Parser, Transformation, ... gleichzeitig
- Verwendung von Fragment Bundles
 - Erweitern Classpath eines anderen Bundles (Host Bundle)
- Verwendetes Framework: Eclipse Equinox

Verwendung von Odysseus



Zusammenfassung

- Odysseus ist ein generisches DSMS-Framework
- Modular aufgebaut (OSGi)
- An vielen Stellen leicht erweiterbar
 - Logische / physische Algebra-Basis
 - Transformationsregeln
 - Restrukturierungsregeln
 - Parser
 - ScS / PPS
- Ideale Plattform für unterschiedlichste Szenarien durch Unabhängigkeit vom Datenmodell

Ende

Vielen Dank für Ihre Aufmerksamkeit.

Literatur

- André Bolles, Marco Grawunder, Jonas Jacobi, Daniela Nicklas und Hans-Jürgen Appelrath. *Odysseus: Ein Framework für maßgeschneiderte Datenstrommanagementsysteme*. Informatik 2009 - Im Fokus das Leben, 2009.
- Michael Cammert, Christoph Heinz, Jürgen Krämer, Bernhard Seeger, Sonny Vaupel und Udo Wolske. *Flexible Multi-Threaded Scheduling for Continuous Queries over Data Streams*. Data Engineering Workshops, 22nd International Conference on, 0, 2007.
- Bugra Gedik, Henrique Andrade, Kun-Lung Wu, Philip S. Yu und Myungcheol Doo. Bugra Gedik, Henrique Andrade, Kun-Lung Wu, Philip S. Yu und Myungcheol Doo. *SPADE: the system's declarative stream processing engine*. In SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, Seiten 1123–1134, New York, NY, USA, 2008. ACM.
- Erich Gamma, Richard Helm, Ralph E. Johnson und John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series, 2008.
- Marco Grawunder. *How to create new query operators in odysseus*.
- Goetz Graefe. *Query Evaluation Techniques for Large Databases*. ACM Computing Surveys, 25(2), 6 1993.
- Peter J. Haas und Joseph M. Hellerstein. *Ripple joins for online aggregation*. In SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, Seiten 287–298, New York, NY, USA, 1999. ACM.
- Jürgen Krämer und Bernhard Seeger. *Semantics and implementation of continuous sliding window queries over data streams*. ACM Trans. Database Syst., 34(1):1–49, 2009.
- Gerd Wütherich, Nils Hartmann, Bernd Kolb und Matthias Lübken. *Die OSGi Service Platform*. dpunkt.verlag, 2008.