ABSTRACT

The principles of agile working methods, short development cycles, customer proximity and customer understanding have arrived in data modeling. Changes to the data model should be possible at any point in the development cycle. A change in a data model directly affects data consumers. The development of a data model takes place independently of the application development. This is taken to extremes by a data lake architecture. The database developer on the one hand works with many changes to the data model. On the other hand, the application developer expects a stable interface to the same data. This creates tension because changes to the data model, especially at high frequency, cannot lead to garantee an stable interface. In order to bridge this tension, a schema evolution process is introduced.

Evolution is an continuous improvement process and passing advantageous properties to a child generation of a living being. Analogous to this, advantageous properties are passed to later versions of the same data model in the schema evolution process. Schema evolution processes accompany the schema evolution in such a way that schema changes are made orderly manner and known to the data consumers. Multitype schema operations are cross-entity schema operations. These are caused by copying or moving attributes. Data consumers are interested in multitype schema operations, since such schema changes represent a semantic change of an existing interface.

In this thesis a data-driven schema evolution process for a data-lake architecture using the NoSQL database Apache Hive has been prototypically developed in cooperation with Commerzbank AG. By continuous data analysis, schema changes from data modeling can be detected. Schema evolution is not a theoretical construct. Within one month up to 2.500 schema changes registered at Commerzbank. This high number motivates the need of an automated schema evolution process. Data analysis includes the steps of similarity calculation, compound calculation and schema operation extraction. Similarities are used to identify attribute pairs that are affected by an evolution process. A composite calculation establishes a row relationship between those attributes. Schema operations can then be extracted by analysing the schema history. Finally, multitype schema operations can be obtained from the schema history.

Three data models are used to evaluate correctness, completeness and performance. The TPC-H data model describes a generic data warehouse scenario. In addition, the data models *CORE* and *KK-Buchung* from the Commerzbank are considered. To evaluate the correctness of the procedures, the generated result is compared with the expected result according to the data model specification. Simulated multitype schema operations in the data models can be used to confirm the correctness. By a detailed status logging the complete processing can be ensured. A variation of the number of entities leads to a linear increase in runtime. Thus, the presented methods are suitable for a practical usage.

Keywords schema evolution, similarity measures, NoSQL databases

ZUSAMMENFASSUNG

Die Prinzipien der agilen Arbeitsweise, kurze Entwicklungszyklen, Kundennähe und Kundenverständnis sind in der Datenmodellierung angekommen. Änderungen am Datenmodell sollen zu jedem Zeitpunkt im Entwicklungszyklus möglich sein. Haupttreiber solcher Änderungen sind Anwendungen, die Daten konsumieren. Eine Veränderung eines Datenmodells beeinflusst Datenkonsumenten. Die Entwicklung eines Datenmodells findet unabhängig von der Anwendungsentwicklung statt. Dies wird durch eine Data-Lake-Architektur auf die Spitze getrieben. Der Datenbankentwickler auf der einen Seite arbeitet mit vielen Änderungen am Datenmodell. Auf der anderen Seite erwartet der Anwendungsentwickler eine stabile Schnittstelle zu denselben Daten. Das führt zu einem Spannungsverhältnis, weil Änderungen des Datenmodells, zumal in hoher Frequenz, nicht dazu führen können, dass die Schnittstelle ständig unerwartete Ergebnisse liefert. Um dieses Spannungsverhältnis zu überbrücken, wird ein Schema-Evolutions-Prozess eingeführt.

Evolution ist ein kontinuierlicher Verbesserungsprozess sowie die Weitergabe vorteilhafter Eigenschaften an eine Kindgeneration eines Lebewesens. Analog dazu werden in der Schema-Evolution vorteilhafte Eigenschaften an spätere Versionen desselben Datenmodells weitervererbt. Schema-Evolutions-Prozesse begleiten die Schema-Evolution dergestalt, dass Änderungen am Schema den Konsumenten der Daten in einem geordneten Verfahren bekannt gemacht werden.

Multitype-Schema-Operationen sind entitätsübergreifende Schema-Operationen. Solche werden durch ein Kopieren oder Verschieben von Attributen verursacht. Datenkonsumenten sind an Multitype-Schema-Operationen interessiert, da solche Schema-Veränderungen eine semantische Veränderung einer bestehenden Schnittstelle darstellen.

In Zusammenarbeit mit der Commerzbank AG ist ein datengetriebener Schema-Evolutions-Prozess für eine Data-Lake-Architektur unter der Verwendung der NoSQL-Datenbank Apache Hive prototypisch entwickelt worden. Durch eine kontinuierliche Datenanalyse können Schema-Veränderungen erkannt werden. Das Thema Schema-Evolution ist kein theoretisches Konstrukt. Innerhalb eines Monats werden bei der Commerzbank in etwa 2.500 Schema-Veränderungen registriert. Diese hohe Anzahl motiviert die Notwendigkeit nach einem automatisierten Schema-Evolutions-Prozess. Die Datenanalyse umfasst die folgenden Schritte: Ähnlichkeitsberechnung, Verbundberechnung und Schema-Operations-Extraktion. Mithilfe von Ähnlichkeiten werden Attribute identifiziert, die durch einen Evolutionsprozess betroffen sind. Durch eine Verbundberechnung wird zwischen diesen Attributen eine Zeilenbeziehung hergestellt. Abschließend können Multitype-Schema-Operationen aus der Schema-Historie gewonnen werden.

Anhand von drei Datenmodellen werden die Korrektheit, Vollständigkeit und die Performance des Schema-Evolutions-Prozesses evaluiert. Das TPC-H-Datenmodell beschreibt ein generisches Datawarehouse-Szenario. Zusätzlich werden die Datenmodelle *CORE* und *KK-Buchung* der Commerzbank betrachtet. Für die Evaluation der Korrektheit der Verfahren werden die Ergebnisse mit den Datenmodellspezifikationen verglichen. Anhand von simulierten Multitype-Schema-Operationen kann die Korrektheit der Verfahren bestätigt werden. Mit einer ausführlichen Statusprotokollierung kann eine vollständige Datenverarbeitung sichergestellt werden. Eine lineare Steigerung der Anzahl der Entitäten führt zu einer linearen Laufzeitsteigerung. Somit eignen sich die vorgestellten Verfahren für einen produktiven Einsatz.

Keywords Schema-Evolution, Ähnlichkeitsmaße, NoSQL-Datenbanken