

# ElasticData™-Technologie

Eine flexiblere Basis für die Datenverwaltung bei modernen webbasierten Anwendungen.

Peter Volynsky, Co-Founder & CTO, Comindware

Konstantin Bredyuk, Dir. of Product Management, Comindware

[www.comindware.com/de](http://www.comindware.com/de)

# ElasticData™-Technologie

## Das ist ElasticData™

Comindware® ElasticData™ ist die Datenverwaltungstechnologie der nächsten Generation, die modernen webbasierten Anwendungen eine hochflexible Datenverwaltung bietet. ElasticData™ basiert auf dem grafischen Datenmodell; ein Datenbanktyp, der geschaffen wurde, um die Einschränkungen und Starrheit der relationalen Datenmodelle zu überwinden. In den letzten Jahren entwickelte sich das grafische Datenmodell schnell von der Erwähnung in wissenschaftlichen Veröffentlichungen über Prototypen mit eingeschränktem Nutzen hin zu vollständig skalierbaren, hochbelastbaren Anwendungen. Twitter verwendet zum Beispiel eine grafische Datenbankimplementierung namens FlockDB. Neben der ElasticData™ von Comindware wird die Umsetzung vieler grafischer Datenbanken realisiert; zum Beispiel: AllegroGraph, Neo4j, HyperGraphDB oder InfiniteGraph. Da die modernen webbasierten Anwendungen mehr Flexibilität, bessere Skalierbarkeit und höhere Geschwindigkeiten erfordern als die traditionellen relationalen Datenbanken leisten können, steigen das Interesse und das Innovationstempo bei grafischen Datenbanken stetig.

Das bei ElasticData™ zugrundeliegende grafische Datenmodell verschafft Comindware-Produkten eine Funktionalität und Flexibilität, deren Entwicklung bei Anwendungen auf der Basis von relationalen Datenbanken schwierig oder extrem kostspielig wäre. Das grafische Datenmodell ist von Natur aus viel flexibler beim Definieren von Datenstrukturen und ermöglicht deren Änderungen „on the fly“. Begründet ist das in dem einzigartigen und deutlich flexibleren Ansatz beim Speichern und Verwalten der Daten.

ElasticData baut auf die Prinzipien des Semantischen Webs, bei dem das renommierte [W3C Consortium](#) federführend ist. Gemäß der Definition des W3C „bietet das Semantische Web einen allgemein zugänglichen Rahmen, der die gemeinsame Nutzung von Daten und den Zugriff darauf über die Grenzen von Anwendungen, Unternehmen und Communities hinweg ermöglicht.“ Durch das Einbeziehen der Prinzipien des Semantischen Webs entstehen Anwendungen, die einfacher zu verstehen und benutzerdefiniert anpassbar sind.

## Darum grafische Datenmodelle

Traditionelle Unternehmensanwendungen auf Basis von relationalen Datenmodellen funktionieren hervorragend, wenn alle Datenbeziehungen und -strukturen schon während ihrer Entwicklung definiert werden (Entwurfszeit). Kommt es aber zu Änderungen am Datenmodell oder an den Prozessen während diese bereits ausgeführt werden (Laufzeit), haben herkömmliche Anwendungen oft Probleme beim Speichern und Verwalten neuer Objekttypen, der Suche nach kürzlich erstellten Daten, beim Hinzufügen von Objektbeziehungen oder Ähnlichem. Diese Schwierigkeiten liegen in der Natur relationaler Datenmodelle: Das Datenmodell soll zunächst mit allen Beziehungen definiert werden und erst danach wird es eingesetzt.

Im Gegensatz dazu wurde ElasticData™ unter der Vorgabe entwickelt, dass Änderungen am Datenmodell und an Prozessen während des Betriebs erfolgen, um besser auf die Anforderungen moderner Unternehmen einzugehen, die sich ständig den veränderten Bedingungen anpassen müssen.

Ein weiterer Nachteil liegt darin, dass relationale Datenmodelle nur für das Aufbewahren von Daten ausgelegt sind, aber nicht die Geschäftsregeln für das Arbeiten mit diesen Daten enthalten. Alle Geschäftsregeln werden als Überbau der Datenbank in einer besonderen Anwendung umgesetzt. Bei nur einer Anwendung ist das ausreichend, gibt es aber zwei oder mehr werden auch an zwei oder mehr Stellen Geschäftsregeln definiert – höchstwahrscheinlich auf völlig verschiedene Art und Weise. Das grafische Datenmodell wurde hingegen dafür entwickelt, die Geschäftsregeln als einen natürlichen Teil der Datenbank zu speichern. Damit enthält nur sie allein die Geschäftsregeln für viele Anwendungen, die mit denselben Daten arbeiten. Die Verarbeitung der Daten unterliegt bei den verschiedenen Anwendungen also denselben implementierten Geschäftsregeln.

## So funktioniert es: Relationale und grafische Datenmodelle im Vergleich

Ein relationales Datenmodell speichert Daten in Form von verknüpften Tabellen. Jede Tabelle enthält in den Reihen eine Anzahl von Einträgen, deren gemeinsame Attribute von den Spalten dargestellt werden. Über spezielle Spalten, die Verknüpfungen zu anderen Tabellen enthalten, sind sie miteinander verbunden.

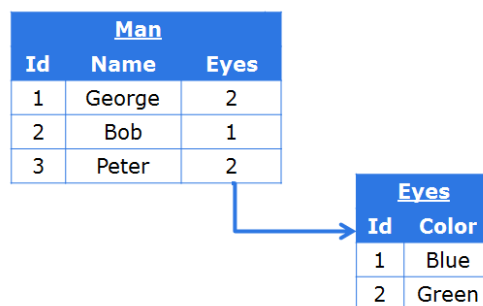


Abbildung 1. Beispiel für ein relationales Datenmodell

Das Darstellen von Daten in Tabellenform ist recht intuitiv und scheinbar einfach zu verwalten. Und tatsächlich ist das auch so, sofern es einen direkten Zugang zur Datenbank gibt und vielleicht fünf Benutzer mit ihr arbeiten. Aber hochentwickelte Unternehmensanwendungen setzen noch aufwändige Geschäftsregeln auf die Datenbank auf und so geht beim relationalen Datenmodell die natürliche Intuition im undurchdringlichen Dickicht verloren. Im Ergebnis werden RDBMS-basierte Anwendungen schnell starr und Veränderungen oder Erweiterungen schwierig. Das Abgleichen von Daten oder Hinzufügen neuer Funktionen bei der Anwendung erfordert viel Zeit und Mühe.

Durch das Verwenden eines grafischen Datenmodells speichert auch ElasticData™ die Daten in einer intuitiven Weise: in Form von Objekten und die Beziehungen zwischen ihnen. Wenn man in Objekten und deren Beziehungen denkt, ist es sogar noch intuitiver als eine Tabelle; z. B.: „Bob und George sind Freunde“ oder „Fenster sind Teil eines Gebäudes“ oder „Dieser bestimmte Monitor ist einer von allen Monitoren“. In einem grafischen Datenmodell bleiben Datenstrukturen intuitiv, weil die Geschäftsregeln

in derselben Datenbank gespeichert sind und als einfacher, deutscher Text gelesen werden können. Komplexe Datentransformationen zwischen Tabellen sind nicht erforderlich, weil fast alle Änderungen und Erweiterungen am Datenmodell durch das Hinzufügen neuer Fakten in einfacher Textform erfolgen.

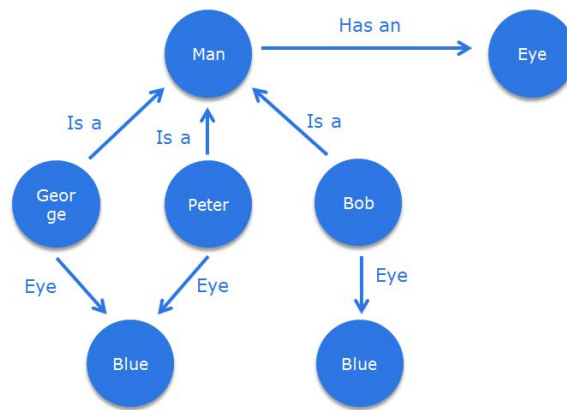


Abbildung 2. Beispiel für ein grafisches Datenmodell

So werden Fakten in einer grafischen Datenbank gespeichert:

*Charles Is Dog*

*Charles Age 11*

*Nikolas Own Charles*

*Charles Like Milk*

Abbildung 3. So werden Daten in einer grafischen Datenbank gespeichert

Im Vergleich zum N3-Standard sind die Geschäftsregeln bei ElasticData™ erweitert, um auch komplexe Regeln verarbeiten zu können. Zum Beispiel:

*if Priority of Request is High*

*then Due Date of Request is Today*

*else Due Date of Request is Tomorrow*

Abbildung 4. Komplexe Geschäftsregeln bei ElasticData™

Sie und alle anderen können diese Regeln jederzeit als einfachen, deutschen Text nachlesen; sie sind nicht in den Tiefen einer benutzerdefinierten Unternehmensanwendung hartcodiert, wie es bei RDBMS-basierten Anwendungen üblich ist.

Zudem reicht das Verwenden von objektrelationalen Mapping-Frameworks nicht aus, um die Herausforderungen aufgrund der vielfältigen Schwierigkeiten bekannt als „objektrelationales Impedance Mismatch“ zu meistern.

In relationalen Datenmodellen erfordern Veränderungen an der Struktur von gespeicherten Daten (den Objekten neue Felder hinzufügen oder neue Objekttypen erstellen) das Hinzufügen bzw. Löschen von Spalten oder sogar das Erstellen bzw. Löschen ganzer Tabellen. Damit wird das Hinzufügen neuer bzw. das Löschen vorhandener Objekte samt den zwischen ihnen bestehenden Beziehungen ein anspruchsvoller und fehleranfälliger Vorgang.

Sogar innerhalb desselben Unternehmens werden Datenstrukturen in verschiedenen Abteilungen für unterschiedliche Zwecke erstellt, so dass deren Zusammenführung aufgrund der Verschiedenartigkeit einen erheblichen Aufwand nach sich ziehen würde: inkompatible Schlüssel, ganz unterschiedliche Ansätze beim Modellieren ähnlicher Objekte usw.

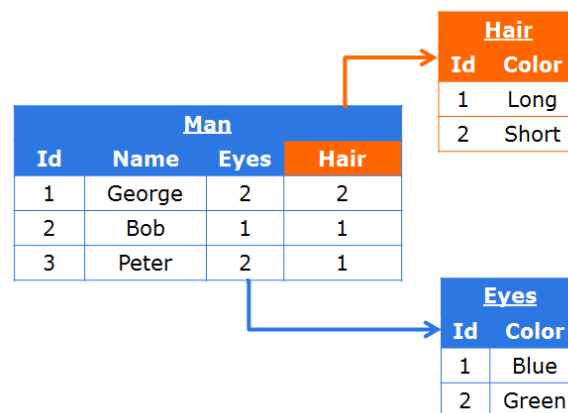


Abbildung 5. Datenstrukturen in einem relationalen Datenmodell aktualisieren

ElasticData arbeitet mit separaten Fakten zu den Daten und nicht mit vordefinierten Tabellen voller Daten. Um weitere Daten einzugeben oder Datenstrukturen zu ändern, werden einfach neue Fakten hinzugefügt – das ist unkompliziert und funktioniert sogar während der laufenden Prozesse. So können völlig neue Datentypen erstellt und mit bestehenden Daten verknüpft werden und zwar ohne damit andere Benutzer zu stören, die die Anwendung bereits verwenden.

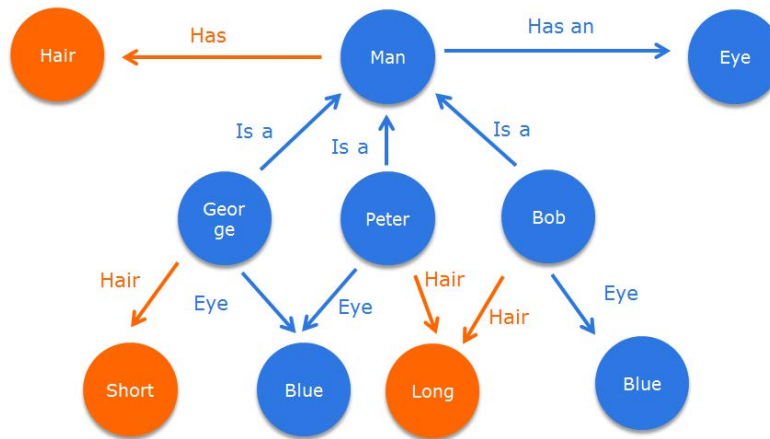


Abbildung 6. Datenstrukturen in einem grafischen Datenmodell aktualisieren

Das schafft deutlich mehr Flexibilität beim Definieren von Datenstrukturen und deren Änderungen „on the fly“. Und die bereits definierten Datenstrukturen können komfortabel als deutscher Text gelesen werden.

Auch bei der Suchfunktion ist das grafische Datenmodell vorteilhaft, weil es auf weitreichenden Wechselbeziehungen zwischen Objekten basiert. Das folgende Beispiel veranschaulicht die Unterschiede von RDBMS- und grafischen Datenbanken bei der Verarbeitung von Suchanfragen. Angenommen, wir haben eine Datenbank mit Benutzerinformationen. Im Datenmodell gibt es die folgenden geschäftlichen Objekte: Aufgaben, Benutzer, Abteilungen und Länder. Wir wollen nach dem Land suchen, in dem eine bestimmte Aufgabe erstellt wurde.

Bei einem RDBM-System sieht die Abfrage höchstwahrscheinlich so aus:

```
SELECT Country.Name from Task

INNER JOIN User ON Task.UserId = User.Id

INNER JOIN Department ON Department.Id = User.DepartmentId

INNER JOIN Country ON Country.Id = Department.CountryId

WHERE Task.Id = 9
```

Abbildung 7. Suche auf Basis der tiefen Querverbindungen in einem relationalen Datenmodell

Und das ist noch kein anspruchsvolles Beispiel. Das Erstellen von Abfragen in einem RDBM-System erfordert qualifiziertes Fachpersonal, die das Datenmodell verstehen und SQL-Abfragen verfassen können. Die meisten RDBMS-basierten Anwendungen versuchen durch das Begrenzen von Abfragefeldern und Suchkriterien diese Komplexität zu verschleiern, sowohl bei der Anwendung selbst als auch bei deren Benutzern.

Bei einer grafischen Datenbank wie ElasticData™ sehen Suchabfragen viel einfacher aus:

*Task9 → User → Department → Country*

Abbildung 8. Suche auf Basis der tiefen Querverbindungen in einem grafischen Datenmodell

Wie Sie sehen, besteht sie aus einer einzigen und leicht verständlichen Zeile, geschrieben von einer technisch nicht versierten Person. Dadurch werden den Benutzern die leistungsstarken Funktionen für das Abfragen und Suchen auf einfache Weise eröffnet.

## Leistungsstarke Comindware-Funktionen durch ElasticData™-Technologie

Wie in diesem Papier beschrieben, ist das von ElasticData™ verwendete grafische Datenmodell wesentlich flexibler und bietet modernen Unternehmensanwendungen wichtige Vorteile gegenüber der klassischen relationalen Datenbank. Den modernen webbasierten Anwendungen [Comindware Tracker](#) und [Comindware Task Management](#) verschafft ElasticData™ viele Chancen und ermöglicht Funktionen wie die folgenden:

- Workflow-Prozesse werden eingerichtet und anschließend schrittweise verändert, weil die aktualisierten Prozesse „on the fly“ bereitgestellt werden können, ohne aktive Prozesse unterbrechen zu müssen. Sie brauchen also nicht alles vollständig im Voraus festlegen. Starten Sie den Workflow-Prozess einfach mit minimaler Formalisierung und aktualisieren Sie ihn später bei Bedarf entsprechend; die Benutzer werden es noch nicht einmal bemerken. Bei RDBMS-basierten Anwendungen müssen Sie häufig ein erneutes Deployment ausführen und aktive Prozesse vorübergehend anhalten, um die aktualisierten Prozesse bereitzustellen.
- Datenmodelle können „on the fly“ geändert werden, ohne erneutes Deployment. Bei ElasticData™ fügen Sie neue Objekttypen und deren Beziehungen untereinander hinzu, ohne aktuell laufende Workflow-Prozesse anzuhalten. Bei RDBMS-basierten Anwendungen ist das Hinzufügen von neuen Objekttypen oft ein komplizierter und eingeschränkter Vorgang mit vielen Problemen wie zum Beispiel dem Erstellen neuer Spalten und Tabellen. Mit ElasticData™ fügen Sie beliebig neue Typen von Geschäftsobjekten, mit denen Sie in Zukunft arbeiten möchten, hinzu und verknüpfen Sie mit den anderen Objekten im System.
- Alle dem System hinzugefügten Daten sind sofort über die Suchfunktion zugänglich, sogar Referenzen zu anderen Objekten; eine erneute Indizierung (wie bei einem RDBMS) ist nicht erforderlich.
- Erstellen Sie auf einfache Weise Datenbankabfragen, die auf den tiefen Querverbindungen zwischen den Objekten basieren (zum Beispiel: Name des Vorgesetzten des Kontos des Produkts des Unternehmens), in ganz normaler Sprache.
- Komplexes Suchen im Verlauf ist möglich, weil Daten und Verlauf in derselben Datenbank gespeichert werden. Daher können aussagekräftige Analysen durchgeführt werden; ohne das Erstellen von OLAP-Cubes, wozu normalerweise qualifizierte und kostenintensive Spezialisten er-

forderlich sind. Mit anderen Worten, Dank ElasticData™ verfügen Sie ständig über ein sofort einsatzbereites OLAP, das nicht bei jeder Änderung am Datenmodell aktualisiert werden muss – wie das bei RDBM-Systemen der Fall ist.

- Separate ElasticData™-Datenbanken können auf einfache Art systemintern ohne Migrationsprozess zu einer einzigen Lösung vereint werden, da gemäß den Standards des Semantischen Webs alle Daten eindeutige Identifier haben. Sie müssen dafür also nicht die Daten von allen Servern exportieren, um sie auf einem gemeinsamen Server zu importieren oder andere Zeit verschwendende Schritte unternehmen. Wenn Sie nach dem Verwenden von separaten Comindware-Instanzen den Bedarf für eine gemeinsame Lösung erkennen, setzen Sie zwischen ihnen einfach die entsprechenden Verknüpfungen.
- Außerdem können Sie ebenso einfach die komplexe Integration von anderen externen Datenquellen konfigurieren, sogar solche mit völlig verschiedenen Datenstrukturen, die gar kein grafisches Datenmodell verwenden. Das Integrieren von Datenquellen mit anderen Datenstrukturen wird konfiguriert, indem der ElasticData™-Datenbank einfache „Fakten“ über die fremde Datenbank hinzugefügt werden; das ist wesentlich komfortabler als das Abgleichen verschiedener flacher Tabellen in RDBMS-basierten Anwendungen.