

Die Umwandlung von Daten in intelligente Aktivitäten

mit Visual Studio 2010 & SQL Server 2008 R2



Immer mehr leicht zugängliche Daten sorgen dafür, dass geschäftliche Entscheider den Wert von zeitnah verfügbaren Daten zu schätzen wissen. Eine weitreichende Datenverfügbarkeit bedeutet, dass die richtigen Personen diese Daten zum passenden Zeitpunkt in aussagekräftige Informationen umwandeln können und so relevante Erkenntnisse erlangen. In diesem Whitepaper erfahren Sie, wie SQL Server 2008 R2 und Visual Studio 2010 Ihnen Lösungen ermöglichen, mit denen Sie Ihre Daten in Erkenntnisse umwandeln und diese dann den richtigen Personen unter Einhaltung der Datenzugriffsrichtlinien des Unternehmens verfügbar machen. Kurz gesagt: Die passenden Personen mit den passenden Daten zum passenden Zeitpunkt.

Heutzutage kommt es auf Daten an

Nehmen Sie eine der großen aktuellen Webanwendungen. Ihr Erfolg basiert höchstwahrscheinlich nicht ausschließlich auf ihrer Benutzeroberfläche. Natürlich ist eine gute Basis-Benutzeroberfläche wichtig. Sie muss jedoch mit hervorragenden Daten gekoppelt werden. Sie muss außerdem über viele unterschiedliche Benutzeroberflächen erweitert werden, die über APIs auf diese Daten zugreifen. Stellen Sie sich vor, Sie können Ihr bevorzugtes soziales Netzwerk nicht über Ihr Mobiltelefon oder Ihr Notebook aufrufen. Oder stellen Sie sich vor, es gäbe keine Elemente aus diesem sozialen Netzwerk, die in andere Webseiten integriert werden können. Webanwendungen und soziale Netzwerke arbeiten nicht mit einer einzigen Benutzeroberfläche. Sie werden erst durch die Verfügbarkeit der Daten in verschiedenen aufgabenspezifischen Anwendungen wirklich wertvoll.

Dieser Wechsel von der anwendungsbezogenen Sichtweise hin zur datenbezogenen Sichtweise passiert auch in der Unternehmenswelt. Daten, die auf eine einzige Anwendung beschränkt sind, sind nicht mehr akzeptabel. Solche Daten sind für die Erkenntnisse, die in einem dynamischen und wettbewerbsorientierten Unternehmen erforderlich sind, einfach zu eingeschränkt. Unternehmen fordern heute stattdessen offene und nutzbare Daten-Feeds, die leicht in neue Tools integriert werden können und so für Wettbewerbsvorteile sorgen.

Je mehr Daten einfach nutzbar werden, desto stärker wird auch geschäftlichen Entscheidern klar, welchen Wert zum richtigen Zeitpunkt und am richtigen Ort verfügbare Daten haben. Eine weitreichende Datenverfügbarkeit bedeutet, dass die richtigen Personen die Daten im richtigen Moment in aussagekräftige Informationen umwandeln können und so für geschäftliche Erkenntnisse sorgen. Um dieses Versprechen erfüllen zu können, müssen BI-Funktionalitäten immer tiefer in das Unternehmen integriert werden. Dies steht im krassen Gegensatz zu der noch vor ein paar Jahren vorherrschenden Ansicht, nach der "Business-Intelligence" und "Datenanalysen" als wichtige und geheimnisvolle Technologie für Führungskräfte eingestuft wurden. Mit diesen Technologien wurden wichtige Muster ermittelt und kritische Entscheidungen getroffen. Diese kritischen Entscheidungen werden jedoch auf allen Ebenen eines Unternehmens getroffen. Wenn nur die obersten Führungskräfte Zugriff auf solche Funktionalitäten haben, dann fehlt Entscheidern auf unteren Hierarchieebenen die Möglichkeit, die zur Realisierung der Anforderungen der Kunden, zur Kostensenkung und für den geschäftlichen Erfolg erforderlichen Erkenntnisse abzurufen. Mit SQL Server 2008 R2 und Visual Studio 2010 ziehen die richtigen Personen unabhängig von ihrer Hierarchieebene wichtige Erkenntnisse aus den verfügbaren Daten – und zwar im Einklang mit den Datenzugriffsregeln des Unternehmens.

Daten > Anwendungen > Information > Aktion

Information-Worker treffen jeden Tag Entscheidungen, die Auswirkungen auf den Erfolg des Unternehmens haben. Mit einem schnellen Zugriff auf Daten können schneller bessere Entscheidungen getroffen werden. So kann das gesamte Unternehmen schneller auf Änderungen im Marktumfeld reagieren.

Es reicht jedoch nicht aus, einen Zugriff auf die Daten zu gewährleisten. Die Information-Worker müssen in der Lage sein, die Daten aus verschiedenen Blickwinkeln zu betrachten. Sie benötigen die Flexibilität, sich einen Überblick zu verschaffen, bestimmte Bereiche detailliert zu überprüfen, Muster zu erkennen und Beziehungen zwischen bestimmten Effekten und ihren Ursachen zu ermitteln. Im praktischen Einsatz kann dies bedeuten, dass verarbeitete Daten in einer oder mehreren Anwendungen genutzt werden. Für detaillierte Einblicke werden dann die Rohdaten genutzt. Die Information-Worker benötigen einen schnellen Zugriff auf die erforderlichen Daten über vertraute Benutzeroberflächen – egal, ob über eine Weboberfläche, ein Office-Produkt oder eine spezielle Client-Anwendung.

Der gesamte Prozess beginnt mit Daten. Diese Daten sollten in leicht einsetzbaren Formaten bereitgestellt werden. Sie sollten schnell zugreifbar sein, und sie sollten vor Fehlern und vor Verlust geschützt werden. Um diese Anforderungen zu realisieren, ist ein leistungsstarker Datenspeicher wie SQL Server 2008 R2 erforderlich.

Anwendungen sollten über einheitliche Ansichten einen einfachen Zugriff auf die Daten ermöglichen. Sie sollten als Grundlage für die Nutzung der Daten dienen und werden meistens durch professionelle Entwickler in Zusammenarbeit mit Datenbankexperten erstellt. Die schnelle Erstellung von qualitativ hochwertigen Anwendungen erfordert leistungsstarke Entwicklungstools wie Visual Studio 2010, die gemeinsam mit SQL Server 2008 R2 arbeiten.

“Wir verfügen über sehr erfahrene Experten. Diese sollten bei der Erstellung von Berichten nicht mehr vollkommen von der IT abhängig sein. Wir müssen diesen Experten die Möglichkeit geben, bei der Erstellung von Analysen und beim Reporting selbstständig aktiv zu werden.”

Stephen Walker, Enterprise Database Architect, Chevron

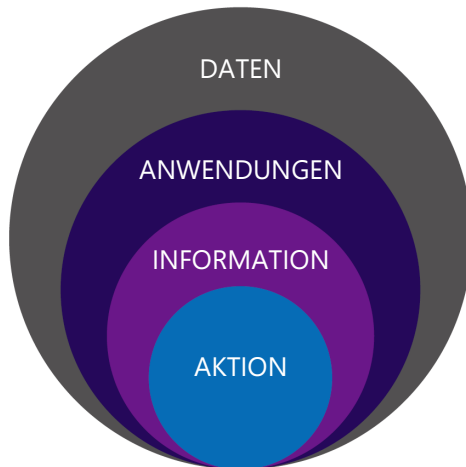


Abbildung 1 – Von den Daten bis zur Aktion

Natürlich können vorab erstellte Anwendungen und Berichte wertvolle Informationen bereitstellen. Knowledge-Worker, die mit den Rohdaten arbeiten, können jedoch oft wichtige zusätzliche Erkenntnisse gewinnen. Unternehmen stellen ihre Daten heute viel mehr Mitarbeitern als früher bereit. Sie bieten ihren Experten die Tools, die diese für ihre Analysen benötigen. Es gibt zwei Voraussetzungen, die für einen Zugriff auf die benötigten Daten durch die Knowledge-Worker erfüllt sein müssen: einfach zugreifbare Daten in nutzbaren Formaten und leistungsstarke und benutzerfreundliche Analysetools.

Sobald den Mitarbeitern eines Unternehmens die passenden Informationen auf Basis zeitnaher und nutzbarer Daten zur Verfügung stehen, können diese sicher agieren – was zu tiefgreifenden Ergebnissen führen kann.

Die Datenbasis

Die Erstellung von Lösungen, die an Daten ausgerichtet sind, ist schwierig. Dies liegt zum Großteil daran, dass Daten in vielen Fällen in Datenbanken gespeichert werden und dort nur für wenige zentrale Datenbankexperten und Entwickler zur Verfügung stehen. Sehen wir uns daher zuerst einmal die Datenbankverwaltung durch die Datenbankexperten an.

Erweiterte Datenbankverwaltung

Die Verwaltung der Dateninfrastruktur kann bei an Daten ausgerichteten Anwendungen eine schwierige Aufgabe darstellen. Diese Anwendungen können erhebliche Datenmengen enthalten, die oftmals auch noch in unterschiedlichen Datenbanken und möglicherweise sogar auf verschiedenen Computern gespeichert werden. Diese Ressourcen müssen alle auf koordinierte Art verwaltet werden.

Um dieses Problem zu lösen, führt SQL Server 2008 R2 den SQL Server Utility Explorer ein (SQL Server-Hilfsprogramm-Explorer). Dieses Tool dient zur zentralisierten Visualisierung und Verwaltung mehrerer Datenbankserver, Umgebungen und Ressourcen als verwaltete Gruppe. Es sorgt außerdem für die Zusammenarbeit zwischen Entwicklung und IT-Abteilung. Hierzu stellt es Dashboards und Ansichten zum Status der Datenbanken und Anwendungen bereit. Die regelmäßige Nutzung des Utility Explorers sorgt dafür, dass Engpässe im Rahmen der Ressourcennutzung und Konsolidierungsmöglichkeiten aufgedeckt werden. Dies sorgt für eine bessere Performance, eine einfachere Verwaltung und weniger Ausfallzeiten.

Bessere Datenintegrität: Master Data Services

Einer der schwierigsten Aspekte bei der Bereitstellung einer autoritativen Datenquelle ist es, die Daten aktuell und konsistent zu halten. Daten veralten mit der Zeit. Es können außerdem ungültige Duplikate auftreten, und es kann durch menschliche Fehler zu Schäden am Datenbestand kommen. Sobald sich

Datenformate ändern, können des Weiteren Inkonsistenzen auftreten. In vielen Fällen ist jedoch ein einziger autoritativer Datenstream für den Einsatz im gesamten Unternehmen erforderlich.

Master Data Services (MDS) ist ein Feature, das mit SQL Server 2008 R2 eingeführt wurde. Es stellt sicher, dass kritische Daten jederzeit verwaltet sind. Mit MDS können Administratoren proaktiv Regeln zur Datenqualität durchsetzen. Sie können außerdem Workflows rund um Datenänderungen definieren und eine einzelne autoritative Datenquelle für verschiedene Anwendungen erstellen. Zusammen mit anderen Features zur Bereitstellung von Daten in Standardformaten kann MDS sicherstellen, dass standardisierte Daten für die richtigen Einsatzzwecke bereitstehen.

Bessere Datenintegrität: Versionskontrolle für das Datenbankschema

Werden Datenbankschemas ohne eine Versionskontrolle gepflegt, so weichen Sie ziemlich schnell von Änderungen am Anwendungscode ab. Dies gilt vor allem dann, wenn der Anwendungscode in Branches aufgeteilt ist (zum Beispiel um verschiedene Versionen des gleichen Codes nutzen zu können). Werden in diesem Fall Änderungen direkt an einer Datenbank vorgenommen (egal, ob im Rahmen eines Tests oder in der Produktivumgebung), so sind diese nicht mehr der entsprechenden Version des Anwendungscode zugeordnet. Ein Rollback der Anwendung, das Debuggen einer älteren Version oder das Nachvollziehen von Änderungen am Schema wird so zunehmend komplexer.

Mit Visual Studio 2010 können Sie ein Reverse-Engineering Ihrer bestehenden Datenbanken durchführen und das Schema in ein Datenbankprojekt mit Versionskontrolle übertragen. Von dort aus können Sie auf sichere Art parallel neben den anderen Entwicklern in Ihrem Team arbeiten. Die Datenbank und der Anwendungscode können als Einheit mit Versionskontrolle entworfen, programmiert, getestet und bereitgestellt werden. Anwendung und Datenbank bleiben hierbei synchron. Sie haben so außerdem die Möglichkeit, Ihr Schema „offline“ zu bearbeiten und dann alle Ihre Änderungen über eine einzige Transaktion bereitzustellen.

“Früher habe ich mit hunderten von Skriptdateien gearbeitet. Hierbei gab es grundsätzlich keine Versionskontrolle. Heute steht uns eine sichere und verwaltete Möglichkeit zur Speicherung des Quellcodes für unser gesamtes Reporting-Data-Warehouse (inkl. aller Datenbankschemas, Integrationspakete, Report-Definitionen und Bereitstellungsskripte und -Utilities) zur Verfügung. Team Foundation Server bietet uns die Sicherheit, exakt zu wissen, was wir bei einem Upgrade bereitstellen, und kontrollierte und verwaltete Änderungen durchführen zu können.“

Jeff Lynch, IT Manager, Gulf Coast Seal

Bessere Datenintegrität: Verwaltung von Änderungen am Datenbankschema

Eine weitere Schlüsselkomponente für die Datenintegrität ist die kontrollierte Verwaltung von Änderungen am Datenbankschema. Ein Problem, mit dem sich Entwickler und Datenbankexperten häufig konfrontiert sehen, ist die Aufgabe, zwei Datenbanken zu vergleichen – nicht nur in Bezug auf das Schema, sondern auch hinsichtlich der Daten. Wenn beispielsweise Version 2 einer Anwendung in die Produktivumgebung eingeführt wird, dann ist es sehr wahrscheinlich, dass sich das Schema geändert hat. Gleiches gilt für die Daten selbst (beispielsweise Vertriebsbereiche oder Produktkategorien). Visual Studio 2010 stellt Ihnen Tools für den Vergleich von Schema und Daten in zwei Datenbanken zur Verfügung. Sie können mit diesen Tools die Skripte generieren, die zur Anpassung des Schemas und der Daten der Zieldatenbank an die der Quelldatenbank benötigt werden.

Diese Skripte können sogar in geplante oder On-Demand-Builds integriert werden. So kann das Datenbankschema in Entwicklungs- und Testumgebungen automatisch aktualisiert werden. Mit einer nächtlichen Bereitstellung von Datenbankänderungen kann das Datenbankteam darauf vertrauen, dass das Schema konsistent ist und die Bereitstellung des Schemas wie geplant funktioniert.

Erweiterte Bereitstellungsoptionen

Heutzutage sehen sich Unternehmen mit verschiedensten Bereitstellungsoptionen konfrontiert: On-Premise, IaaS (Infrastructure as a Service) oder PaaS (Platform as a Service). Sie können ihre Datenbanken also intern bereitstellen, auf einer Maschine, die mit Azure als VHD gehostet wird, oder über SQL Azure. Die On-Premise-Bereitstellung und IaaS bieten eine vollständige Kontrolle über die Administration von SQL Server 2008 R2. SQL Azure gibt dem Unternehmen hingegen die Möglichkeit, die Administrationsaufgaben abzugeben und sich auf die Datenbankerstellung zu konzentrieren.

Beim On-Premise-Hosting kann SQL Server 2008 R2 in vielen verschiedenen Editionen genutzt werden (zum Beispiel Datacenter und Parallel Data Warehouse). So ist eine hohe Skalierbarkeit und Leistung möglich. Beim Hosten per PaaS können Sie Ihre Anwendung direkt über SQL Azure bereitstellen. Praktisch bedeutet das, dass Sie sie entweder On-Premise oder in der Cloud hosten können. Mit ein paar Konfigurationsänderungen kann eine On-Premise-Datenbanklösung nach SQL Azure migriert oder bereitgestellt werden. Und natürlich ist der entgegengesetzte Weg ebenso einfach möglich.

Eine Schlüsselkomponente zur vereinfachten Bereitstellung, Verwaltung und Aktualisierung von Datenbankanwendungen (sowohl On-Premise als auch in SQL Azure) ist die Einführung von DACs (Data Tier Application) in SQL Server 2008 R2. Eine DAC definiert das Database Engine-Schema und die Objektinstanzen (beispielsweise Tabellen, Ansichten und Logins), die für die Anwendung erforderlich sind. Dieser neue Mechanismus sorgt dafür, dass eine einzelne Datei mit Versionskontrolle an den DBA übergeben werden kann. Diese kann der DBA dann entweder zur On-Premise-Bereitstellung im Rechenzentrum oder zur Bereitstellung in der Cloud nutzen. Visual Studio 2010 und SQL Server 2008 R2 bieten bei der Arbeit mit DACs eine hohe Flexibilität. DACs können beispielsweise für bestehende Datenbankanwendungen erstellt werden. Sie können außerdem zur Wartung, Veröffentlichung und Bereitstellung neuer und nachfolgender Versionen genutzt werden. Wenn eine Nachfolgeversion einer Anwendung auf Datenebene in ein neues DAC-Paket implementiert wird, dann kann dasselbe Paket zur Bereitstellung einer neuen Instanz des DAC oder zur Aktualisierung einer bestehenden Instanz auf die neue Version genutzt werden. Ein Assistent vergleicht das Schema einer bestehenden DAC-Instanz und das neue DAC-Paket. Er führt dynamisch die Aktionen aus, die zur Umwandlung der bestehenden DAC-Instanz in die neue Version der DAC erforderlich sind. Sie können Visual Studio 2010 sogar dazu nutzen, ein einzelnes Bereitstellungspaket zur Bereitstellung über SQL Server 2008 R2 oder SQL Azure zu erstellen.

Die Schnittmenge zwischen Daten und Anwendungen: So werden intelligente Aktionen möglich

Zur Umwandlung von Daten in intelligente Aktionen ist eine umfassende Plattform erforderlich. Diese muss die Datenbankverwaltung und die Tools integrieren. Nur so können die Entwickler Anwendungen erstellen, die den Benutzern die benötigten Informationen zum passenden Zeitpunkt zur Verfügung stellen.

Integrieren der Datenbankentwicklung in das Application Lifecycle Management

Eines der wichtigsten Features von Visual Studio 2010 stellen die ALM-Tools (Application Lifecycle Management) dar. Mit ihnen können Teams ihre Arbeit integrieren und nachverfolgen. Diese anpassbare ALM-Unterstützung wird über Visual Studio 2010 Team Foundation Server realisiert. Team Foundation Server ermöglicht eine Kontrolle des Quellcodes, das Zuweisen und Nachverfolgen von Aufgaben, eine Build-Automatisierung, die Nachverfolgung von Anforderungen und Fehlern, die Systemverifizierung und ein Reporting.

Visual Studio 2010 bietet einige der Vorteile, die Anwendungsentwickler bei der tagtäglichen Datenbankadministration benötigen. Automatisierte Builds zum regelmäßigen Kompilieren, Integrieren, Bereitstellen und Testen des Codes können beispielsweise auch die automatisierte Datenbankbereitstellung abdecken. Mit den erweiterten Build- und Bereitstellungsfeatures für

Datenbanken aus Team Foundation Server 2010 können sowohl das Datenbankschema als auch Datenänderungen jede Nacht gemeinsam mit dem Anwendungscode erstellt, bereitgestellt und getestet werden.

Visual Studio 2010 stellt außerdem bestimmte Tools für Datenbankentwickler bereit, die diese bei der Integration in die gesamte Entwicklung unterstützen. Die Datenbankentwickler können ihre Arbeit so am Rest des Teams ausrichten. DBAs und andere IT-Experten haben früher oftmals ohne „Netz und doppelten Boden“ gearbeitet. Änderungen an Datenbanken und Systemen wirkten sich direkt aus. Dieser häufig verwendete Ansatz ist zwar leicht zu realisieren, sorgt aber für drastisch steigende Risiken. Die Auswirkungen eines Fehlers können ziemlich weitreichend sein. Visual Studio 2010 integriert die DBAs und IT-Experten mit dem Rest des Entwicklungsteams und mit dem Unternehmen. So wird der bisher risikoreiche Ansatz geordnet. Durch die Nachverfolgung von Datenbankänderungen über Vorgaben und geschäftliche Anforderungen kann das Datenbankteam ermitteln, wann und warum bestimmte Änderungen an einem System durchgeführt wurden. Sie haben so die Möglichkeit, Ihre Änderungen mit den Änderungen der Anwendungsentwickler in Beziehung zu setzen. Eine solche Teamintegration ist ein wesentlicher Faktor zur Vermeidung von Überraschungen in der Produktivumgebung, unzufriedenen Kunden und ungeplantem Arbeitsaufwand.

Die Nachverfolgbarkeit, die sich durch die Aufzeichnung von Änderungen an den Anforderungen ergibt, hat weitreichende Auswirkungen auf das gesamte Entwicklungsteam. Nicht nur die Anwendungs- und Datenbankentwickler nutzen Team Foundation Server, sondern auch die Tester und die geschäftlichen Analysten. Somit steht ihnen ein integrierter Überblick der Qualität der gesamten Anwendung zur Verfügung. Mit dutzenden von standardmäßig verfügbaren Berichten können Sie den Fortschritt, die Qualität und die Risiken anzeigen. In Abbildung 2 sehen Sie einen dieser Berichte. Er zeigt den Arbeitsaufwand, den Testfortschritt und die Fehler in Bezug auf die einzelnen Anforderungen auf. Die ALM-Tools von Visual Studio 2010 machen Ihre Entwicklung sicherer, konsistenter und schneller. Trotzdem sind Sie auch weiterhin in der Lage, Änderungen schnell umsetzen. Sie können die Tools mit traditionellen oder agilen Prozessen und natürlich mit einer beliebigen Kombination aus beidem nutzen. Die Collaboration-Möglichkeiten in Teamprojekten auf Basis von Team Foundation Server 2010 können Ihr Team darin unterstützen, komplexe Änderungen durchzuführen, und sicher, kontrolliert und zuverlässig mit großen IT-Systemen und hoher Komplexität umzugehen.

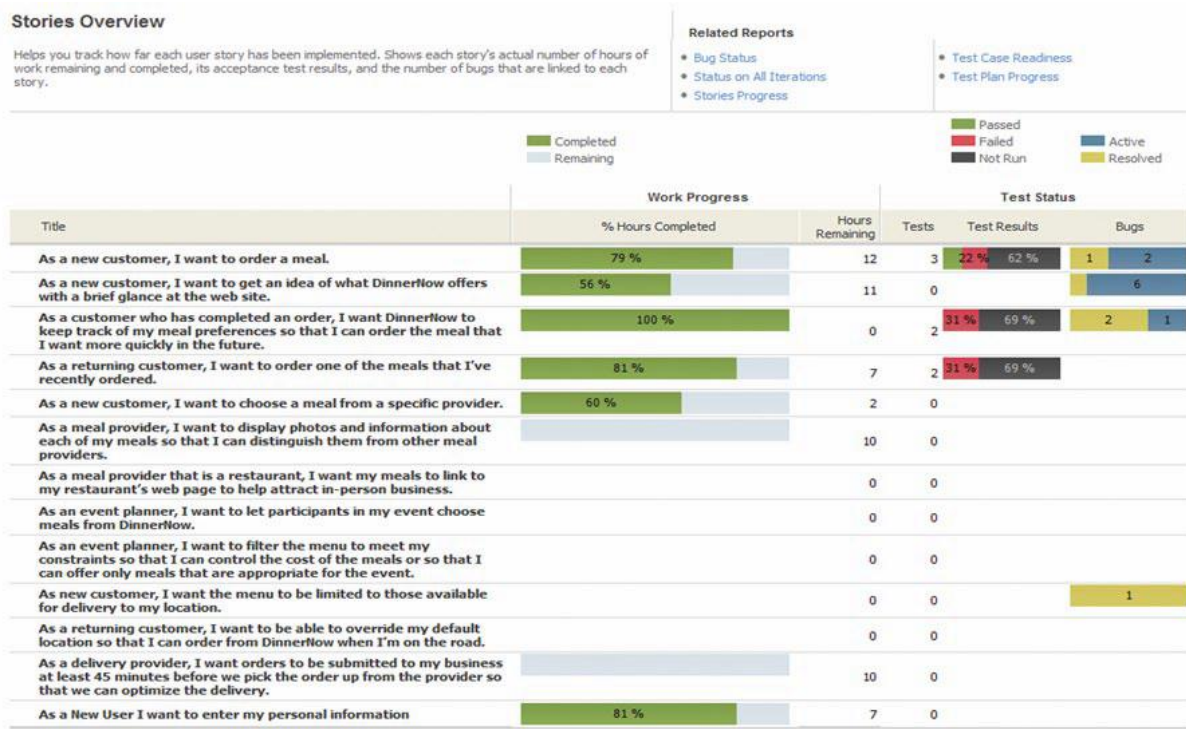


Abbildung 2 – Umfassende Nachverfolgbarkeit

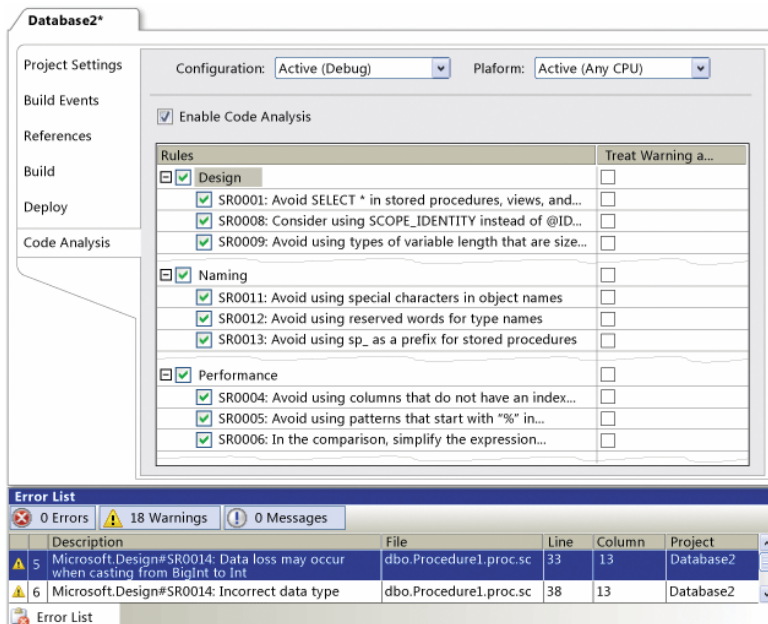
Sobald das Datenbankschema durch eine Versionskontrolle gesichert ist, können Sie damit beginnen, einige der erweiterten Features zur Datenbankentwicklung, beispielsweise das automatisierte Refactoring, Datenbank-Unit-Tests, statische Analysen und Tier-Interaction-Profiling, einzusetzen. Mit diesen Features haben Datenbankentwickler die Möglichkeit, ihre bestehenden und neuen Datenbanken schnell zu bearbeiten, zu testen und zu validieren. Bisher waren diese Aktivitäten oft sehr fehleranfällig und zeitaufwendig. Ohne effektive Tools ist das Refactoring eines Datenbankobjektes ein gefährvoller Prozess, der ungewollte Auswirkungen haben kann. Mit den Tools aus Visual Studio 2010 können Sie Ihre Datenbanken nun erheblich sicherer testen und bearbeiten.

Das Refactoring gibt Ihnen die Möglichkeit, ein Datenbankobjekt umzubenennen oder es in ein anderes Schema zu verschieben, eine automatische vollständige Qualifizierung aller Datenbankobjekte durchzuführen oder alle Datenbank- oder Serverreferenzen umzubenennen. Sie können sogar automatisch alle Wildcards in SELECT-Statements erweitern.

Ihnen steht des Weiteren ein leistungsstarkes Tool zur Generierung von Daten für den Einsatz in Tests zur Verfügung. Dies ist besonders dann von entscheidender Bedeutung, wenn PII-Daten (Personally Identifiable Information – personenbezogene Daten) in der Produktivdatenbank gespeichert werden. Das Datengenerierungstool gibt Ihnen die Möglichkeit, Ihre Testdaten auf Basis der Produktivdaten zu erstellen und alle PII-Daten in nutzbare Testdaten umzuwandeln. Die Entwickler und Tester können so realistische Daten einsetzen und gleichzeitig wird für die Benutzerdaten in der Produktivdatenbank der erforderliche Datenschutz gewährleistet.

Bei den Datenbank-Unit-Tests stehen Tools zur Erstellung automatisierter Unit-Tests zum Testen von gespeicherten Prozeduren und benutzerdefinierten Funktionen bereit. Diese Unit-Tests können dann nachts oder bei Bedarf ausgeführt werden. Sie stellen sicher, dass sich Schemas oder Daten nicht negativ auf die Datenbank auswirken.

Mit statischen Analysen kann Transact-SQL auf Design-, Benennungs- und Leistungsprobleme überprüft werden. So können Sie Probleme proaktiv erkennen und Ihre Datenbank wartungsfreundlicher und leistungsstark halten.



“SQL Server 2008 R2 stellt für uns einen echten Paradigmenwechsel dar. Wir können unseren Kunden standortbasierte Funktionalitäten bereitstellen, mit denen sie das „wo“ und das „was“ in ihren Anwendungen verknüpfen können.”

Peter Hammond, President, CyberSavvy

Abbildung 3 – Statische Analysen für Datenbanken

Tier-Interaction-Profiling zeigt Ihnen, wie oft eine Datenbankabfrage oder eine gespeicherte Prozedur durch Ihre Anwendung ausgeführt wurde. Sie haben die Möglichkeit, den von ADO.NET für die entsprechende Abfrage benötigten Zeitraum zu ermitteln. Wenn Sie eine Webanwendung testen, dann werden die entsprechenden Daten noch weiter aufgeschlüsselt. Sie können die Anzahl der Aufrufe einer Seite sehen, die Verarbeitungszeit für die Seite und die Abfragen, die von der entsprechenden Seite

aufgerufen wurden. Es ist sogar möglich, den Text der Abfragen anzeigen zu lassen. Dies kann Sie zum Beispiel beim Refactoring Ihres Programmcodes unterstützen (beispielsweise, indem Sie die Anzahl der Datenbankaufrufe durch Ihre Anwendung verringern oder Bereiche für mögliche Leistungsverbesserungen ermitteln).

Verbessertes Transact-SQL

In vielen Fällen wird eine Anwendungs- oder Validierungslogik direkt in einer gespeicherten Prozedur programmiert. Auf diese Weise können die geschäftlichen Logiken sehr eng an den Datenspeicher gebunden werden – was für eine sehr hohe Leistung sorgt. Die direkte Arbeit mit Transact-SQL kann für Mitarbeiter, die mit der entsprechenden Syntax nicht so vertraut sind, jedoch recht kompliziert und aufgrund der fehlenden komplexen Datentypen auch sehr frustrierend sein.

SQL Server 2008 R2 führt einige Erweiterungen ein, die Ihnen die Arbeit mit Transact-SQL vereinfachen. Eine vollständig integrierte Volltextsuche, Sparse-Columns (Spaltensätze) und gefilterte Indizes sorgen für die schnellere Suche in Daten. Des Weiteren wurde Transact-SQL um umfangreiche Datentypen erweitert, mit denen Sie nun nahezu jede Art von Daten handhaben können (inklusive präziser Datums- und Zeitinformationen, XML-Daten und sogar externe Dokumente und Dateien).

Der wohl wichtigste neue Datentyp sind die Geoinformationen. Mit ihnen können geografische Daten in einem einzigen Feld gespeichert und mit den Verbesserungen in Transact-SQL auch abgefragt werden. Mit Geoinformationen können Sie Standortdaten nun direkt festhalten und Abfragen erstellen, die Datensätze mit „geografischer Nähe“ zurückgeben. Beispielsweise „Alle Kunden im Umkreis von 100 Kilometer um Kunde A“.

Zusätzlich unterstützt Sie IntelliSense bei der Erstellung von Transact-SQL-Logiken mit Visual Studio 2010. Die Entwicklung neuer gespeicherter Prozeduren, benutzerdefinierter Funktionen und Abfragen kann so vor allem bei weniger erfahrenen Datenbankentwicklern drastisch beschleunigt werden.

Bessere Unterstützung für Object-Relational Mapping

Der Brückenschlag zwischen relationalen Datenbankschemas und dem objektorientierten Design von Anwendungen ist schon immer recht schwierig gewesen. Der direkte Zugriff auf eine Datenbank ist oft recht fragil. Auch bei kleinen Änderungen an der Datenbankstruktur wird eine Neukompilierung erforderlich. Die Erstellung einer Zugriffsschicht für die Datenbank oder auch nur der Einsatz einer schon vorhandenen Zugriffsschicht ist auch weiterhin ziemlich arbeitsaufwendig. Viele der entsprechenden Lösungen sind außerdem mit automatisierten Testtools nur schlecht zu testen.

Visual Studio 2010 und .NET 4.0 stellen Ihnen das Entity Framework bereit. Ein erweitertes ORM-Tool (Object-Relational Mapping) vereinfacht die Interaktion zwischen Programmcode und Datenbank erheblich. Das Schöne am Entity Framework ist, dass es eine einfache Drag-and-Drop-Datenmodellierung unterstützt und gleichzeitig einige sehr leistungsfähige Features bietet: zum Beispiel Lazy-Loading, n-Tier-Support, testfähige Interfaces und eine vollständige LINQ-Unterstützung (Language Integrated Query).

Das Entity Framework unterstützt außerdem unterschiedliche Programmierstile. Wenn Entwickler erst Datenmodelle erstellen, dann können sie die Vorteile der

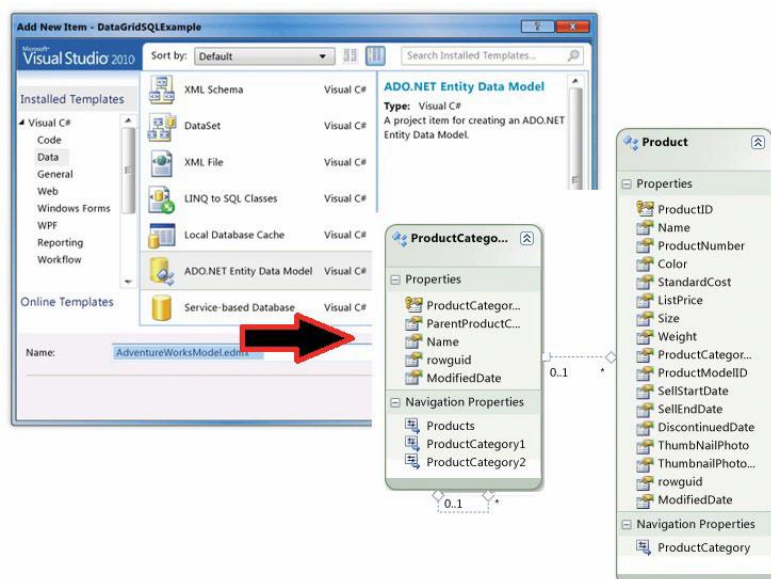


Abbildung 4 – Drag-and-Drop mit dem Entity Framework

grafischen Tools nutzen und mit diesen .NET-Klassen aus bestehenden Datenbankschemas generieren. Die generierten Klassen umfassen unter anderem bereits das Laden und eine LINQ-Unterstützung. Natürlich können sie auch erweitert werden. Wenn Entwickler zuerst eine objektorientierte Struktur entwerfen, dann gibt ihnen das Entity Framework die Möglichkeit, ein Datenbankschema über .NET-Klassen zu generieren. Die Softwareentwickler können so Datenbankschemas bereitstellen, ohne Transact-SQL-Code schreiben zu müssen.

Automatische Generierung einer anpassbaren und erweiterbaren Benutzeroberfläche

Eine der einfachsten Möglichkeiten, eine Benutzeroberfläche mit einer Datenbank zu verbinden, ist das direkte Drag-and-Drop aus einer Datenbankabfrage oder gespeicherten Prozedur in den Designbereich. Diese Möglichkeit wird für alle wichtigen Benutzeroberflächen unterstützt: zum Beispiel Web, Silverlight, WebForms und WPF. Für die Entwickler, die lieber vollständig datenzentrierte Websites entwickeln (heute eine Grundanforderungen in fast allen Unternehmen), steht eine noch einfachere Lösung parat: ASP.NET Dynamic Data.

Visual Studio 2010 und ASP.NET Dynamic Data sind eine Lösung für stark datenzentrierte Anwendungen, die Datenbankinformationen schnell und einfach bereitstellen müssen. Durch ein umfangreiches Basisframework und die Wiederverwendung bestehender ASP.NET-Datensteuerelemente ist eine sehr schnelle Anwendungsentwicklung möglich. Durch die einfache Registrierung eines LINQ-to-Entity Framework-Datenmodells in einem Projekt (in wenigen Minuten erledigt) können Sie eine voll funktionsfähige Website erstellen, die alle CRUD-Operationen (Create, Read, Update, Delete) mit einem Button-Klick unterstützt. Die so entstandene Site und die Transformationsregeln für die Daten sind vollständig anpassbar. Sie können zum Beispiel problemlos eine numerische Personalausweisnummer in ein Format mit Bindestrichen umwandeln.

Viele Entwickler haben bereits Erfahrungen mit anderen Tools zur Generierung von Websites gemacht und stehen automatisierten Lösungen daher kritisch gegenüber. Sie sind der Meinung, dass die enge Koppelung zwischen Benutzeroberfläche und Daten dafür sorgt, dass die Wartung einer solchen Website schwierig wird. ASP.NET Dynamic Data bindet die Benutzeroberfläche jedoch nicht direkt an die Datenbank. Stattdessen kann es mit dem sauber erstellten Entity Framework-Modell arbeiten. Daher befindet sich zwischen Anwendung und Daten eine zusätzliche Abstraktionsschicht, die mit extrem wenig Aufwand eine überraschend hohe Leistung und Anpassbarkeit bietet. ASP.NET Dynamic Data ist tatsächlich sehr flexibel, unterstützt eine anpassbare Benutzeroberfläche, bietet eine anpassbare Validierung der Datenfelder und sorgt für die anpassbare Anzeige von Datentabellen.

Standarddatenformat zur einfacheren Datennutzung: So werden intelligente Aktionen möglich

Berichte sind hervorragend dazu geeignet, Informationen grafisch darzustellen. Manchmal ist es jedoch erforderlich, auf die Rohdaten zuzugreifen. Self-Service-Szenarien sind daher für ein flexibleres Unternehmen von grundlegender Bedeutung.

Um Daten aus SQL Server 2008 R2 in nutzbaren Formaten darzustellen, gibt es unzählige Technologien. Die wohl bekannteste dieser Technologien ist OData (Open Data). OData ist ein offener Standard zur Bereitstellung von Daten über HTTP. Mit OData können Benutzer und Anwendungen auf Daten zugreifen, die über einen OData-Endpunkt veröffentlicht werden. So haben sie die Möglichkeit, Daten über REST (ein Format zur Interaktion mit Daten über URLs) abzufragen. Eine der einfachsten Lösungen zur Bereitstellung von Daten ist Reporting Services. Berichte, die über SQL Server Reporting Services erstellt werden, können optional auch die zugrundeliegenden Daten als OData-Stream bereitstellen. Die Entwickler und Benutzer können so die Daten hinter einem Bericht in ihren eigenen Anwendungen, Berichten und Ad-hoc-Analysen nutzen.

OData wird bereits von vielen Anwendungen und Entwicklungsframeworks unterstützt. PowerPivot für Excel 2010 kann beispielsweise mit OData arbeiten. Die entsprechenden Daten können mit anderen Datenquellen zusammengeführt werden und ad-hoc von den Endbenutzern bearbeitet werden. Die

meisten Webbrowser können des Weiteren direkt auf OData-Daten zugreifen. Entwicklern stehen außerdem Bibliotheken für Silverlight 4, PHP, Windows Phone 7 und viele andere Plattformen zur Verfügung.

Wie Sie im nächsten Abschnitt sehen werden, hat die Möglichkeit der Bereitstellung von Daten über OData auf Unternehmen, die wichtige Daten auch in unteren Entscheidungsebenen bereitstellen, weitreichende Auswirkungen.

Mehr Einblick – Report Builder 3.0, Excel, PowerPivot und Sharepoint

Die Darstellung von Daten in Berichten ist eine leistungsstarke Möglichkeit, wenn es darum geht, Daten besser zu verstehen. Zum direkten Bearbeiten von Daten „On the fly“ durch Experten ist sie aber nicht geeignet. Ein solcher Experte ist jedoch mit den richtigen Daten und den richtigen Tools dazu in der Lage, Erkenntnisse zu gewinnen, was mit sorgfältig entworfenen und vorbereiteten Berichten nicht möglich ist.

Früher war die Bereitstellung eines Zugriffs auf Rohdaten für die Endbenutzer gleichbedeutend damit, ihnen direkte Zugriffsrechte auf den Datenspeicher zu gewähren. Es war außerdem erforderlich, ihnen die entsprechenden IT-Ressourcen zur Umwandlung und Darstellung der Daten an die Hand zu geben.

Heute ist die Bearbeitung solche Daten durch Endbenutzer hingegen Realität. Stephen Walker ist Enterprise Database Architect bei Chevron Data Management Solutions und meint zu diesem Thema: „Uns stehen fachkundige und erfahrene Experten zur Verfügung. Wir möchten nicht mehr, dass sie in Bezug auf die Erstellung und Verwaltung von Berichten vollkommen von der IT abhängig sind. Wir müssen diesen Experten die Möglichkeit geben, selbstständig Analysen und Berichte zu erstellen und zu bearbeiten.“¹

Microsoft Excel, PowerPivot, Report Server 3.0 und SharePoint stellen den Endbenutzern gemeinsam leistungsstarke Möglichkeiten zur Nutzung, Analyse und Darstellung von Daten zur Verfügung.

Leistungsstarke ad-hoc-Datenanalysen

Microsoft Excel war schon immer ein hervorragendes Tool zur Datenanalyse. Die meisten Endbenutzer, die sich mit Daten befassen, kennen sich mit Pivot-Tabellen zur schnellen Datenanalyse aus. Mit Pivot-Diagrammen können sie die entsprechenden Ergebnisse außerdem schnell darstellen. Ihnen stehen umfangreiche Optionen zur Datentransformation bereit. In der neuesten Version hat Microsoft diesem bereits hervorragenden Tool nun noch wichtige Möglichkeiten hinzugefügt.

Eine der wichtigsten Erweiterungen von Excel ist PowerPivot. PowerPivot gibt den Endbenutzern die Möglichkeit, auch die größten Datensätze aus nahezu jeder Datenquelle zu laden, massive Datenmengen in wenigen Sekunden zu verarbeiten (inklusive der Nutzung von Mehrkernprozessoren) und eine Vielzahl von neuen Analysefunktionalitäten wie beispielsweise Data Analysis Expressions einzusetzen. Und das alles über die vertraute Benutzeroberfläche von Excel.

Eine mögliche Datenquelle für PowerPivot können Daten im OData-Format sein. Endbenutzer können somit einen bestehenden Bericht aus Reporting Services nehmen und die zugrundeliegenden Daten in Excel verarbeiten. Diese Daten können dann mit Daten aus anderen Quellen (beispielsweise einem online CRM-Tool) zusammengeführt und gemeinsam analysiert werden. Durch die Zusammenführung



Abbildung 5 – PowerPivot in Excel

¹ http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=4000007043

mehrerer Datenstreams können die Endbenutzer leistungsstarke Lösungen schaffen, mit denen Daten auf ganz neue und einzigartige Art angezeigt werden.

Excel und PowerPivot stellen beide hervorragende Tools zur Datenvisualisierung bereit. Mit diesen Tools können Sie Dashboards, Diagramme und andere Darstellungen erstellen. Sie realisieren so mehr Einblick und neue Erkenntnisse, die für andere leichter verständlich sind.

Einfache Erstellung von Berichten durch die Endbenutzer

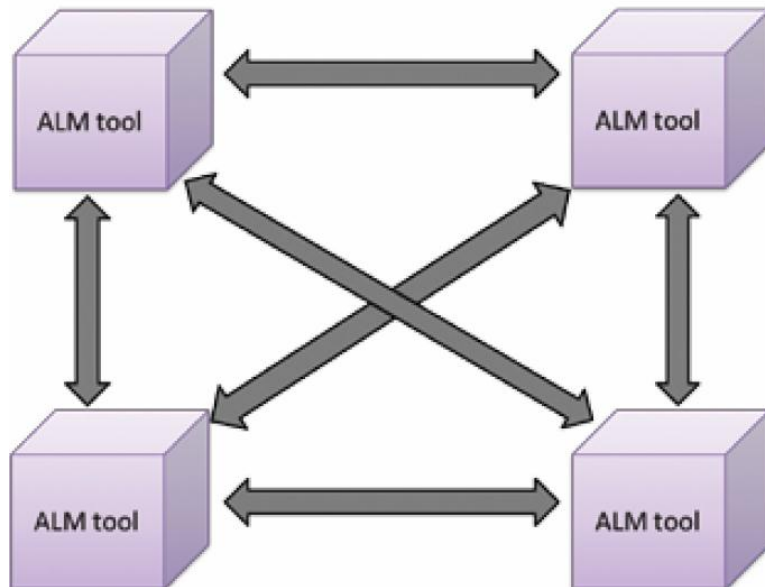


Abbildung 6 – Der Report Builder 3.0 in Aktion

Die Bereitstellung von Daten in visuellen Formaten ist eine der wichtigsten Grundlagen für besser informierte Entscheidungen. Menschen realisieren wichtige Informationen viel einfacher über ein Diagramm oder eine Abbildung als über eine Tabelle. Die Erstellung von brauchbaren Grafiken kann jedoch schwierig sein.

Mit SQL Server 2008 R2 und Visual Studio 2010 hat das Reporting einen großen Schritt nach vorne gemacht. Neben den bestehenden Assistenten, OLAP-Cube-Funktionalitäten und anderen Features unterstützt SQL Server Reporting Services (SSRS) nun auch Report Builder 3.0.

Report Builder 3.0 bietet den Benutzern eine Office-ähnliche Umgebung, über die sie selbst ohne Unterstützung durch die IT Berichte erstellen können. Die Benutzer legen fest, wo die Daten herkommen, welche Daten relevant sind, und wie sie angezeigt werden sollen. Hierbei sind viele verschiedene Datenquellen möglich (inklusive Daten aus SharePoint-Listen und SQL Azure). Zur Auswahl der anzuzeigenden Daten stehen umfangreiche interaktive Tools wie beispielsweise Abfragegeneratoren bereit. Zu Anzeige der Daten können die Benutzer dann umfangreiche Visualisierungstools einsetzen (inklusive geografischen Beziehungen und Darstellungen und anderen Möglichkeiten). Es ist außerdem ganz leicht möglich, auf Detailinformationen aus den Berichten zuzugreifen.

Einfachere Weitergabe von Erkenntnissen

Die Möglichkeit, durch die schnelle, effiziente und effektive Analyse von Daten mehr geschäftlichen Einblick zu realisieren, ist bereits für sich genommen von hohem Wert. Können diese Erkenntnisse jedoch auch leicht mit anderen Benutzern im Unternehmen geteilt und von anderen auf unterschiedliche Arten vermischt werden, so ist es möglich, die gewonnenen Erkenntnisse in konkrete Aktionen umzuwandeln. Genau dann profitiert das Unternehmen wirklich von neu gewonnen Erkenntnissen.

Historisch verbreiten sich neue geschäftliche Erkenntnisse in einem Unternehmen eher langsam. Damit die richtigen Informationen in die passenden Hände gelangen, werden Excel-Tabellen per E-Mail verschickt, die IT erstellt Websites, oder Tabellen werden über ein Portal gehostet. Mit Visual Studio 2010 und SharePoint gehört dies alles der Vergangenheit an. Für den Endbenutzer, der mit Excel oder PowerPivot arbeitet, ist die Nutzung von SharePoint 2010 sowohl einfach als auch effektiv. Mit SharePoint 2010 und Excel Data Services können die Benutzer Diagramme und andere visuelle Komponenten aus Excel direkt auf einer SharePoint-Seite hosten.

Die Endbenutzer können also Portale mit vielen unterschiedlichen Ansichten ihrer Daten erstellen. Und zwar auf einer selbst erstellten SharePoint-Seite sowie ohne Unterstützung durch die IT. Auch PowerPivot unterstützt SharePoint. So können Sie nicht nur visualisierte Daten bereitstellen, sondern diese Daten können auch sehr leicht direkt über die SharePoint-Seite bearbeitet werden. PowerPivot für SharePoint stellt des Weiteren ein Dashboard bereit, mit dem die IT-Manager den Nutzungsverlauf nachverfolgen können. Da diese Funktionalitäten, wenn sie zur Verfügung stehen, mit der Zeit immer umfangreicher genutzt werden, ist diese Nachverfolgungsmöglichkeit sehr wichtig.

Auch die Benutzer von Report Builder 3.0 haben die Möglichkeit, ihre Lösungen in SharePoint hochzuladen und über das Portal anzuzeigen. Sie können daher viele der Vorteile von Excel- und PowerPivot-Lösungen realisieren. Für Report Builder 3.0 stehen jedoch noch zusätzliche Optionen



Abbildung 7 – PowerPivot-Verwaltungsdashboard

bereit. Visual Studio 2010 umfasst verschiedene Report Viewer-Komponenten, mit denen Reporting Services-Berichte in jeder Benutzeroberfläche angezeigt werden können (inklusive Web, Windows und WPF). Die Report Viewer-Komponente für das Web stellt einige leistungsstarke Erweiterungspunkte bereit (unter anderem die Möglichkeit zum Einsatz von JavaScript, zur Verwendung von AJAX für asynchrone Datenupdates und viele zusätzliche Optionen zur Steuerung des Look-and-Feel von Berichten). Die Web-Komponente wird von vielen verschiedenen Browsern

unterstützt (unter anderem Internet Explorer 6 und höher, Firefox 3.5 und höher und Safari 4). Mit zusätzlicher Unterstützung durch Entwickler können vom Benutzer erstellte Berichte so in eine Webseite oder Smart-Client-Anwendungen eingebunden und für große Zielgruppen bereitgestellt werden.

Egal, ob über Excel, PowerPivot oder Report Builder – das Erstellen und Weitergeben von neuen Datenanalyselösungen war noch nie so einfach wie heute. Sowohl SharePoint 2010 als auch Visual Studio 2010 bieten einfache Funktionalitäten für Endbenutzer und Entwickler, mit denen diese ihre Lösungen einem großen Publikum zur Verfügung stellen können.

Starten Sie noch heute

Die Tools, die Sie zur Realisierung von besseren Erkenntnissen benötigen, sind heute verfügbar. Durch die Integration von Features und Tools von SQL Server 2008 R2 und Visual Studio 2010 können Sie bereits heute Lösungen erstellen, die aus Daten neue Erkenntnisse und informierte Aktionen machen. In diesem Whitepaper haben Sie erfahren, welche Vorteile Ihnen die Durchdringung von Daten und Anwendungen über die Microsoft-Plattform bietet:

- Qualitativ hochwertigere Daten und Anwendungen
- Stabilere Entwicklungs- und Reporting-Tools für das gesamte Unternehmen
- Schnellere Marktreife für Anwendungen
- Self-Service-Reporting

Sie haben außerdem erfahren, wie im gesamten Unternehmen verfügbare Daten die geschäftliche Tätigkeit des Unternehmens revolutionieren können. Fangen Sie daher noch heute damit an, Ihre Daten in echte Aktivitäten umzuwandeln.

Weitere Informationen finden Sie unter <http://www.microsoft.com/germany/visualstudio> und <http://www.microsoft.com/germany/sql/2008>.