Voll-Integrierte DG
(Aspekt 2)

Data Governance

Thilo Riegel, München

Zusammenfassung

Data-Governance-Serie, Teil 6.2: Wir haben in uns den letzten Teilen mit verschiedenen Dimensionen der DG befasst: der Informations-Dimensions, der horizontalen, der vertikalen sowie mit dem DQ-Management. In diesem Teil sehen wir uns an, wie die geschickte Kombination diese Dimensionen zu einem effektiven und ganzheitlichen Lösungsansatz für einige der Probleme führt, die wir im ersten Teil aufgezählt haben.

Integration der Systeme und des Contents

Business Glossary

BI- und DWH-Projekte haben typischerweise gemeinsam: Hohe Komplexität und enormen Umfang. Und wenn man unsere Forderungen an einen Data Governor (Stichwort „Auskunftsfähigkeit“) im Sinne der vorigen Artikel ernst nimmt, dann folgt daraus, dass ein DG nicht nur auf allen Ebenen des Modells Bescheid wissen muss, sondern dass die oberste Ebene dabei auch das Business Glossary abdeckt – oder besser noch: Die oberste Ebene *ist* das Glossary.

Deshalb ist es wichtig, dass alle Beteiligten ein einheitliches und aktuelles Verständnis von den Dingen haben, von denen die Rede ist. Auf hoher Abstraktions-Ebene ist das ein Business Glossary – es umfasst dann in unserer Sprechweise die Entitäten, die KPIs und ggf. spezielle Reports und Fachfunnktionen.

Solch ein Repository kann man meistens noch problemlos auf konventionelle Art erstellen und pflegen (Word und Excel). Allerdings hatten wir ja schon den Punkt erwähnt, dass es ja nicht allein nur um die Dinge an sich geht, sondern auch mindestens genau so sehr um die *Beziehungen* zwischen ihnen. Hier wird es mit üblicher Office-Software schon schwieriger. UML-Modellierungs-Tools sind für solche Sachen besser geeignet. Die marktüblichen Tools unterstützen Hyperlinks „von innen nach außen“ (z.B. von der Klasse im UML-Diagramm zu externen Dokumenten oder eben auch zu Einträgen im Meta-Repository), und manchmal, mit ein bisschen Tricksen, auch von “von außen nach innen“ (z.B. vom Eintrag einer Entität im Meta-Repository zu einem separat auf dem Dateisystem gespeicherten UML-Diagramm, das diese Entität genauer erklärt). Die Glossar-Kurzerläuterungen können einfach in das Repository als führendes System übernommen werden, und konventionelle Listen-Glossare können dann per Report erstellt werden. Auf diese Weise nutzt man die Stärken beider Systeme und hält die Datenredundanzen auf ein Minimum.

Im gleichen Gedanken kann man die Sache auch weiter treiben: Das Business Glossary wird detailliert und erweitert, indem man im Repository zu den Entitäten auch fachliche Felder (nicht BDEs) hinzufügt.

These

Rein Text- und Listen-basierte Business Glossaries, die separat in einem Word-Dokument gepflegt werden, gehören der Vergangenheit an. Das ist erstens technisch nicht mehr zeitgemäß, und zweitens ist das bei der heutigen Komplexität von BI- und DWH-Projekten auch meistens gar nicht mehr beherrschbar. Ein modernes Business Glossary ist *integraler Bestandteil* des Gesamtmodells des Repositories.

Management der Redundanzen

Das Management der Redundanzen ist bei Einführung eines Meta-Repositories entscheidend. Allerdings (und hier mal eine „erleichternde" Nachricht): So umfangreich und komplex solch ein Projekt auch werden kann, so relativ einfach ist es, das zu bewerkstelligen. Letztlich kann man, wie wir es schon praktiziert haben, durch ein paar einfache organisatorische Festlegungen, durch geschicktes Verlinken sowie durch Import- und Export-Mechanismen die Redundanzen minimieren bzw. automatisiert managen.

Auf der Entitäts-/BDE-Ebene, z.B., ist das relativ einfach: Hier bestimmt man einfach nach Präferenz organisatorischen Randbedinungen des jew. Hauses das führende System. Daraus ergeben sich dann klar die Workflows, die nicht besonders schwer zu implementieren sind.

Auf der BDE-Ebene ist dann in der Praxis sowieso alles klar: Alles, was hier ist (die BDEs, ihre Verbindungen untereinander sowie die Verbindungen nach „oben" und nach „unten") werden nach meinem Vorschlag originär und führend im Repository gepflegt, weil ich auf dem Markt bislang kein geeignetes Tool gesehen habe.

Auf der technischen Ebene wiederum macht man es sinnvollerweise umgekehrt, weil es sehr gute und breit eingeführte Tools zur technischen Modellierung gibt (die wiederum häufig bestimmte andere Systeme mit Informationen beliefern), so dass das technische Modell im Repository eine Kopie sein sollte, die durch Importe upgedatet wird (damit werden die Redundanzen automatisch gemanagt). Das ist zwar nicht ganz einfach, aber die Komplexität dessen hält sich in Grenzen – der Ansatz ist Praxis-erprobt.

These

Die Minimierung bzw. das Management von Redundanzen ist ein Punkt, der nicht vernachlässigt werden darf. Es gehört nicht zu den einfachsten Sachen, aber mit einem bisschen guten Willen und etwas Disziplin bekommt man das in den Griff.