

Datenmodelle für dynamische Datengenossenschaften:

Zum Umgang mit anspruchsvollen Anforderungen an Agilität und Vertraulichkeit

Eine Datengenossenschaft erfordert eine agile und skalierbare technische Basis für das von ihr unterstützte Daten-Ökosystem. Hieraus resultiert die Forderung, dass die zugrundeliegende Plattform und hier insbesondere die Datenhaltung dynamisch erweitert werden können, sei es zur Ergänzung neuer Partner, einer neuen Datenquelle (etwa eine neue Maschine), eines neuen Sensors oder Sensortyps, einer neuen Kennzahl oder auch zusätzlicher Kontextdaten (etwa Daten zur Prozesskonfiguration).

Abstraktionsschritte für eine problemlose Erweiterbarkeit

Eine Lösung, die im Projekt gewählt wurde, ist das Einziehen von Abstraktionsschritten in das Datenmodell: Statt mit konkreten Tabellen zu arbeiten (etwa „Luftdrucksensor bei Schneidemaschine x“), werden Tabellen für *generische Objekte* geschaffen (DT_Objekt), denen dann jeweils ein oder mehrere *Sensoren* (DT_Sensor) zugeordnet werden. Jeder Sensor spezifiziert wiederum Zeilen einer Tabelle „Messung“ (FT_Messung), wobei eine Messung neben dem Sensor noch eine Messzeit zugeordnet ist (erfasst in „DT_Uhrzeit“ und „DT_Datum“ erfasst). Diese Tabellen können bei Bedarf durch weitere, beschreibende Tabellen spezialisiert oder erweitert werden (vgl. Abbildung 1).

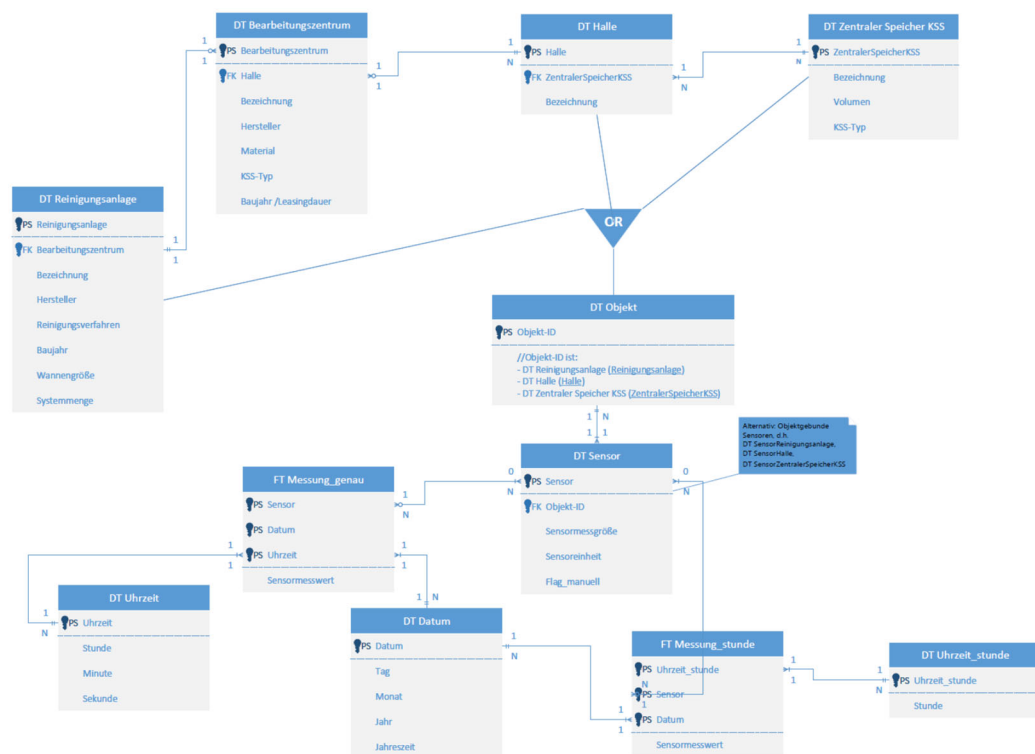


Abbildung 1: Datenmodell im Kühlschmierstoff-Fall

Die Abkürzung „DT“ steht für „Dimensionstabelle“, da die entsprechenden Einträge der Tabelle zur Selektion und Aggregation der Messwerte/Fakten der „Faktentabelle“ (FT) *Messung* genutzt werden können und damit zum Navigieren im Datenraum. Beispielsweise können über die Dimensionstabellen alle Messungen eines Sensors an einer bestimmten Maschine ausgewählt werden, die jeweils montags zwischen 13 und 14 Uhr erfasst wurden.

Die Einbindung eines neuen Sensors und ggf. eines neuen Sensortyps erfordert bei diesem Ansatz keine Modifikation an bestehenden Tabellen: Es wird lediglich die Sensortabelle um eine neue Zeile ergänzt, in der neben einer neuen Sensor-ID die Messgröße und die Maßeinheit des Sensors festgehalten werden. Dadurch ist dann auch spezifiziert, welche Semantik diejenigen Zeilen in der „Messung“-Tabelle haben, die diesem Sensor später zugeordnet sind.

Jeder Sensor ist des Weiteren einem Objekt zugeordnet, in dem er verbaut ist und dessen genaue Ausprägung variieren kann. Im Kühl-/Schmierstoffszenario wären das beispielsweise eine *Reinigungsanlage*, ein *Schmierstoffspeicher* oder eine *Halle*.

Alle Objekte eines Wertschöpfungsszenarios werden zunächst ohne weitere Beschreibung in der DT_Objekt-Tabelle mit reinen Identifikationsdaten („ObjektID“) erfasst. Diese wird jedoch bei Bedarf um weitere Tabellen ergänzt, in denen spezifische Angaben zu den konkreten Objekttypen festgehalten werden *können* (etwa in „DR_Reinigungsanlage“ Daten zum Baujahr und zum umgesetzten Reinigungsverfahren). Auf diese Weise werden einerseits alle Objekte einheitlich verwaltet, andererseits hat das Ergänzen einer Tabelle für einen neuen Objekttyp keinerlei Auswirkungen auf bestehende Tabellen. Es ist dabei z. B. ohne weiteres möglich, Daten zu einem Schmierstoff-Entsorger mit *Entsorgungsbehältern* zu ergänzen. Jeder Entsorgungsbehälter wäre wieder ein Objekt mit einer eigenen Objekt-ID, zu dem ggf. zusätzliche Informationen in einer neuen Tabelle „DT_Entsorgungsbehälter“ erfasst werden können. So kann das Wertschöpfungsszenario dynamisch erweitert werden

Da der Sensor selbst ein Objekt ist, können einem Sensor zudem auch selbst Sensoren zugeordnet werden, etwa zur Überwachung des Sensors oder zur Abbildung eines sog. „virtuellen Sensors“, der aus den Messungen mehrerer physischer Sensoren einen neuen, kombinierten Messwert errechnet.

Möglichkeit zur Separierung von Datenbereichen

Die Trennung von unverfänglichen Objekt-Identifikationsdaten und spezifischen Kontextdaten in eigenen Tabellen erleichtert außerdem die Aufteilung des Datenraums in allgemein für die Genossen zugängliche Tabellen und Kontexttabellen, die jeweils spezifischen Data Ownern zugeordnet sein können. Hierbei zu beachten ist, dass einem Objekt ohne Weiteres auch mehrere Kontexttabellen zugeordnet werden können (etwa bei einem Hallen-Objekt „Halle“ neben einer Tabelle „DT_Halle“ auch eine zusätzliche zur „DT_Elektroinstallation_Halle“ oder eine zur „DT_Netzanbindung_Halle“).

Beachtet werden muss, dass auch die Zustandsdaten vertraulich sein können, da hieraus etwa Beschäftigungsintensitäten oder -zeiten abgeleitet werden können. Um dem zu begegnen, werden die Messungen in unterschiedlichen Tabellen für unterschiedliche Verdichtungsstufen festgehalten, für die jeweils wieder andere Zugangsregeln definiert werden können. Im Beispiel erfasst die Tabelle „FT_Messung_genau“ Messungen auf Sekundenebene, die Tabelle „DT_Messung_Stunde“ jedoch nur auf Stundenebene aggregierte Daten. Weitere Aggregationsstufen sind hier ebenfalls denkbar.

Weitere Besonderheiten: Messmodalitäten, Historisierung und weitere Abstraktion

In dem Datenmodell lassen sich des Weiteren besondere Erfassungsmodalitäten festhalten. So war es im Kühlschmierstoff-Fall erforderlich, zwischen automatisch und manuell erfassten Daten unterscheiden zu können, zum einen zur Einschätzung der Datenqualität, zum anderen zur Kalibrierung von Sensoren (mit manuellen Daten als Referenzdaten). Hierfür wurde eine entsprechende Angabe („Flag“) in die Sensortabelle eingezogen („Flag_manuell“), mit der sich manuelle Daten markieren oder auch in Gänze ein- oder ausblenden lassen. Dieser Ansatz ist erweiterbar etwa für die Kennzeichnung weiterer Eigenschaften der Messerte: etwa bei korrigierten

Daten, rein synthetischen Daten, Daten, die in einer Simulation in einem Digitalen Zwilling erzeugt wurden, etc.

Nicht im in Abbildung 1 gezeigten Datenmodell abgebildet ist eine weitergehende Historisierung der Daten, um etwa Änderungen von Eigenschaften oder Zuordnungen über die Zeit verfolgen zu können, beispielsweise bei der Umpositionierung einer Maschine. Dies lässt sich jedoch problemlos durch die Ergänzung weiterer Attribute für Gültigkeiten („gültig_von“ und „gültig_bis“, „aktuell“) realisieren. Zu beachten dabei ist, dass die identifizierenden Schlüssel grundsätzlich unabhängig von den konkreten Identifikatoren der physischen Objekte gewählt werden sollten. Beispielsweise sollte eine ObjektID keine „reale“ Maschinenummer sein, da diese im Laufe der Zeit neu vergeben werden kann.

Ergänzung und Fazit

Wir haben bewusst davon abgesehen, das Modell noch weiter zu abstrahieren, wie es etwa die Datenmodellierungsansätze „DataVault“ (Lindstedt et al. 2009, S. 1 ff., S. 56 ff.; Hultgren 2012, S. 19 ff., Hahne 2014) oder das „Anchor-Modell“ (Rönnebeck et al., 2010) vorsehen, die unser Modell maßgeblich inspiriert haben. Der Hintergrund ist, dass ein Datenmodell für eine Datengenossenschaft trotz aller Flexibilität für alle Genossen verständlich bleiben muss – auch für solche ohne vertiefte Datenmodellierungkenntnisse. Wir haben uns deshalb hier für einen Kompromiss aus Flexibilität und Zugänglichkeit entschieden.

Abgesehen davon halten wir in einer Datengenossenschaft ein Datenmodell für erforderlich, das einerseits ohne Modifikations- und Neuentwicklungsaufwand erweitert werden kann, andererseits aber eine gezielte Aufteilung in verschiedene Datenbereiche für unterschiedliche Data Owner, Verwendungskontexte und Vertraulichkeitsstufen erlaubt. Ein entsprechend agil handhabbares Datenmodell ist ein weiterer Baustein für eine agile Datengenossenschaft auch für schnellwachsende und dynamische Datenökosysteme.

Quellen

- Hahne, M.: Logische Modellierung mehrdimensionaler Datenbank-systeme. Deutscher Universitätsverlag, Wiesbaden, 2002, S. 237 f.
- Hultgren, H.: Modeling the Agile Data Warehouse with Data Vault. New Hamilton, Denver, 2012.
- Lindstedt, D., Graziano, K., Hultgren, H.: The Business of Data Vault Modeling, 2. Aufl., lulu.com, Morrisville, 2009.
- Rönnbäck, L., Regardt, O., Bergholtz, M., Johannesson, P., Wohed, P.: Anchor modeling – agile information modeling in evolving data environments. Data Knowl. Eng. 69(12), 2010, S. 1229-1253