

# **Praxisbericht: Horizontale und vertikale Integration am Beispiel des Auftragsmanagements bei der MCG der DaimlerChrysler AG**

Rainer Schrapel / Thomas Sauter

ITP/FO CoC Global Ordering  
DaimlerChrysler AG  
HPC 050 G202  
Fronäckerstr. 40  
71059 Sindelfingen  
rainer.schrapel@daimlerchrysler.com

**Abstract:** In diesem Beitrag wird kurz der weltumspannende Kundenauftragsprozeß der Mercedes Car Group mit den Hauptbeteiligten, den Märkten, der Vertriebszentrale und den Werken, und die diesen Prozeß unterstützende Systemwelt dargestellt. Es wird dann im Hauptteil gezeigt, welche IT-Mechanismen eingesetzt werden, um die verschiedenen, teils heterogenen, Systeme oder Teilsysteme zu einem integrierten Gesamtsystem werden zu lassen.

## **1 Aufgabenstellung GO**

GO ist die Abkürzung für Global Ordering und steht zum einen für den neugestalteten Kundenauftragsprozess der MCG (Mercedes Car Group: Mercedes, Maybach, smart). Der Begriff GO steht zum anderen auch für das IT-System im Zentralvertrieb, das diesen Prozeß unterstützt. GO ist ein Großrechner-basiertes Client-Server-System. Eingebettet in eine heterogene Systemlandschaft bildet GO im Sinne einer vertikalen Integration das Bindeglied des Auftragsmanagements vom Kunden über die Vertriebsstufen zum Werk und zurück, sowie im Sinne einer horizontalen Integration die Verbindung der zentralen Prozesse Bedarfs- und Volumenplanung, Auftragsmanagement incl. Baubarkeitsprüfung, Fahrzeugdistribution und Fakturierung. Durch GO wurden und werden nach und nach bestehende Systeme abgelöst.

Die Hauptziele des neugestalteten Kundenauftragsprozesses sind die folgenden:

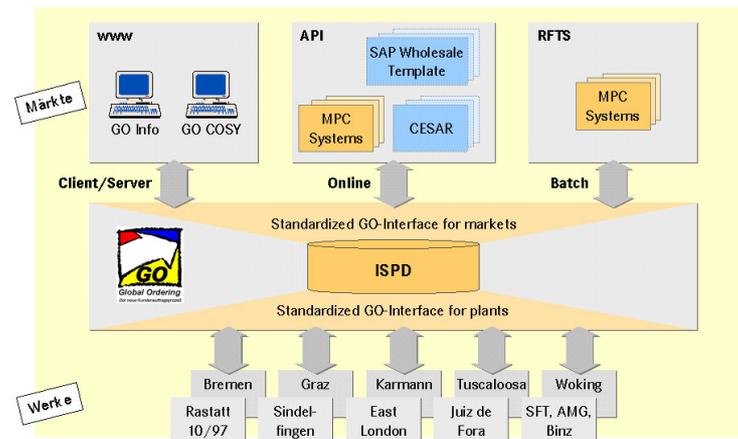
- **Verbindlichkeit** (sichere Zusage eines Liefertermins)
- **Flexibilität** (späte Änderungsmöglichkeit von Aufträgen)
- **Transparenz** (Statusinformation zu Aufträgen und deren Historie)
- **Prozessqualität** (durchgehende IT-Unterstützung, redundanzfreie Stammdaten)
- **Prozessgeschwindigkeit** (kurze Durchlaufzeiten, Online-Kopplungen)

Im Kern basiert das System GO technisch auf folgenden Prinzipien:

- **Integration aller Funktionen** des zentralen Vertriebs in einem System.
- **Integration aller Daten** von Vertrieb und Produktion in einer gemeinsamen Datenbank **ISPD** (Integrated Sales and Production Database)
- **Modularer Aufbau** sowie Bereitstellung **Standardisierter Schnittstellen** zu Nachbarsystemen unterstützt durch eine „Prozesssteuerung“

## 2 Systemeinbindung

Das System GO ist bzgl. des Auftragsmanagements wie eine zentrale Datendrehscheibe zu sehen:



## 3 Verwendete Mechanismen

### 3.1 API-Dienste

Ein wesentlicher Mechanismus in GO zur Anbindung insbesondere externer Nutzer sind API-Schnittstellen. Diese Schnittstellen sind durch folgende Punkte gekennzeichnet:

- Die Schnittstelle ist durch die Verwendung von Records von der fachlichen Geschäftsvorfall-Schnittstelle entkoppelt. Die Dienste-Schnittstelle enthält damit keine GO-intern genutzten Datentypen. Eine Änderung von GO-Datentypen oder von Entitäten hat daher keine direkte Auswirkung auf die Dienste-Schnittstelle.

- Die Geschäftsvorfälle sind versioniert. Ein Nutzer sendet zu einem Geschäftsvorfall nur das Geschäftsvorfall-Kennzeichen und die gewünschte Version. Er hat keine Kenntnis darüber, welches Modul aufgerufen wird. Die GO-Vorbereiterschicht leistet die fachliche Umsetzung der aktuellen Geschäftsvorfall-Schnittstelle ins bestehende Record-Format.
- In GO werden gleichzeitig bis zu drei unterschiedliche API-Versionen eines einzelnen fachlichen APIs unterstützt.
- Neben synchronen Schnittstellen sind teilweise auch asynchrone Rückschnittstellen erforderlich, da eine Änderungsinitiative an einem Geschäftsobjekt auch von einer im Regelfall gerufenen Ebene ausgehen kann. (Z.B. Änderung eines Auftrags durch Entstehung der Fahrzeugidentifikationsnummer im Werk.)

### 3.2 Steuermodul

Das Steuermodul in GO dient:

- **der Versorgung von GO mit Daten aus Nachbarsystemen**
- **der Versorgung von Nachbarsystemen mit Daten aus GO**
- **der asynchronen Nachverarbeitung von Daten innerhalb von GO**

Die Geschäftsvorfälle publizieren Nachrichten (Telegramme = Metadaten + Record), wenn sich Zustände von Geschäftsobjekten geändert haben. Die Nachrichten werden nicht direkt von Sender zu Empfänger übertragen, um das Wissen über solche Beziehungen bewußt vor den betroffenen Partnern zu verbergen. Stattdessen wird ein Steuermodul adressiert, das die Beziehungsdetails kennt und durch Parametrierbarkeit eine leichte Änderung oder Erweiterung von Beziehungen ermöglicht.

Die GO-interne Verwendung des Steuermoduls nennen wir **Prozeßmanager (PROM)**. Die Prozesssteuerung realisiert auf oberster Ebene die Modularisierung von GO in Teilsysteme. Darüber hinaus kann das Steuermodul in anderen Rollen eingesetzt werden:

**Importeure** nehmen Daten aus Nachbarsystemen entgegen, machen entsprechende Aktualisierungen der Datenbank und geben Trigger an den PROM weiter.

**Exporteure** nehmen Trigger vom PROM entgegen und bauen Nachrichten für fremde Systeme auf.

**Worker** dient dazu, interne asynchrone Folgeverarbeitungen zu GO-Geschäftsvorfällen durchzuführen.

Als Middleware für den Nachrichtenaustausch und damit die Prozesssteuerung kommt MQS zum Einsatz.

Ein weiteres wichtiges Prinzip bei der Kommunikation zwischen Ebenen oder Teilsystemen ist das Prinzip des Subsystembestandes. Hiermit wird registriert, welches Geschäftsobjekt bei welchem Teilsystem bekannt ist. Damit können Aktualisierungen von Geschäftsobjekten zielgerichtet durchgeführt werden.

### 3.3 Prozess- und Datenübertragungseckpunkte

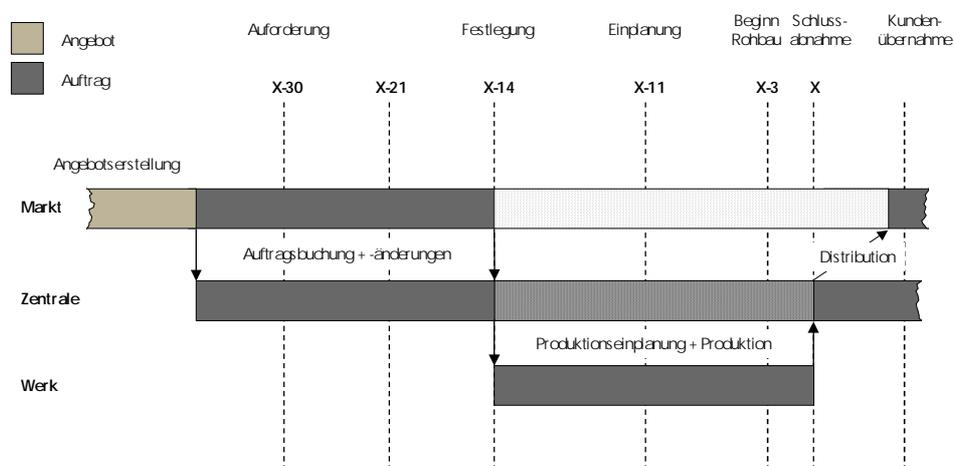


Abbildung 1: Prozesseckpunkte eines Auftrages

Das Bild zeigt, dass ein Auftrag auf bis zu drei Systemebenen bekannt sein kann. Falls er auf mehreren Ebenen existiert, werden alle Repräsentationen ständig konsistent gehalten. Dies wird durch den Subsystembestand gesteuert. Sobald ein unverbindliches Angebot zu einem verbindlichen Auftrag wird, wird dieser in der Zentrale abgelegt. Der Weiterverkauf an das ausführende Werk erfolgt mit einem festen Vorlauf bzgl. des geplanten Schlußabnahmetermins. Selbst wenn der Auftrag bereits im Werk bekannt ist, kann er bis zur Einplanung noch geändert werden.

### 3.4 Records

Records dienen dazu

- die Schnittstelle variabel zu machen, z. B. wenn eine variable Anzahl von Elementen einer Datenstruktur übergeben werden sollen.
- interne Repräsentationen von Daten zu verbergen
- den zu bearbeitenden Geschäftsvorfall (GV) und die zugehörigen Daten auf einer Meta-Ebene sprachunabhängig abzubilden



Abbildung 2: Format eines Records

Records werden in synchronen wie auch asynchronen Schnittstellen eingesetzt.

#### 4 Ist-Zustand

Aktuell sind ca. 20 Märkte über API online angebunden. Dadurch wird die große Mehrzahl der weltweit existierenden Aufträge abgedeckt. Mittels einer asynchronen Dateischnittstelle sind weitere 10 Märkte bzw. Marktgruppen an GO angebunden, die die verbleibende Lücke fast vollständig schließen. In den verbleibenden Märkten werden GO-Java-Clients für das Auftragsmanagement eingesetzt.

Alle Werksstandorte sind nachrichtengestützt angebunden.

#### 5 Ausblick

In der vertikalen Integration ist es das Ziel hinsichtlich der Marktanbindungen den Verbreitungsgrad des Online API, sowie die Anzahl der bereitgestellten und genutzten Geschäftsvorfälle zu erhöhen. Bei den Werksanbindungen ist es das Ziel die Komplexität der Integration nicht nur durch standardisierter Schnittstelle, sondern auch durch standardisierte Systeme zu reduzieren.

In der horizontale Integration ist es das Ziel einige verbliebene vor GO bestehende Systeme zu integrieren, wie z.B. die Systeme der Volumenplanung der Märkte und der Fakturierung. Dabei wird sowohl eine technische Bereinigung als auch eine Anpassung auf neue Prozeßanforderungen verfolgt.