

Das Unternehmensmodell CyberEnterprise

Das Unternehmensmodell CyberEnterprise

- [Einleitung](#)
- [Reale Objekte und Begriffe: die eigentlichen Geschäftsobjekte](#)
- [Prinzipien der Modellierung](#)
- [Geschäftsobjekte als funktionale Bausteine](#)
- [Grundmechanismen betriebswirtschaftlicher Anwendungen](#)
- [Dynamisierung des Modells durch Transaktionen](#)
- [Der Transaktionsmanager](#)
- [Spezielle Objekte des CyberEnterprise](#)
 - [Attribute](#)
 - [Strukturen](#)
 - [Allokationen](#)
 - [Relationen](#)
 - [COM-Objekte](#)

Einleitung

Das Unternehmensmodell steht im Mittelpunkt der Betrachtungen. Es gilt – unabhängig von einer spezifischen Anwendungsfunktionalität – die Grundbausteine zu definieren, mittels derer man reale Unternehmen in ihrer gesamten Komplexität virtuell modellieren kann: Die Implementation eines kybernetischen Unternehmens, eines CyberEnterprise.

Hierzu müssen reale geschäftliche Daten und Abläufe auf wenige, unterschiedliche Strukturen reduziert bzw. abstrahiert werden. Immer wiederkehrende Datenstrukturen und Berechnungsalgorithmen müssen so aufbereitet werden, um Bausteine jeglicher betrieblicher Anwendungen sein zu können.

Die so gewonnenen Grundbausteine bilden dann – aufgrund ihrer beliebigen Kombinier- und Erweiterbarkeit – die Vielfalt unternehmerischer Informationen und Prozesse dar. Als Teil der CyberEnterprise Architektur ermöglichen sie die effiziente Programmierung komplexer betriebswirtschaftlicher Anwendungen.

In den folgenden Abschnitten wird der Aufbau des Modells von „oben nach unten“, von den komplexeren eigentlichen Geschäftsobjekten zu den sehr spezifischen Basisobjekten her beschrieben. Dieser Weg soll die Abstraktion von realen, geschäftlichen Gegebenheiten besser nachvollziehen lassen.

Reale Objekte und Begriffe: die eigentlichen Geschäftsobjekte

Das statische Abbild von Unternehmen wird beschrieben durch die realen Objekte: Partner und Sachen. Diese Objekte repräsentieren Daten-Entitäten, die für alle am Wirtschaftsprozess teilnehmenden Parteien gleich erscheinen und sind.

Die aus Sicht des zu modellierenden Unternehmens (Mandant) relevanten Informationen und Eigenschaften realer Objekte werden mittels Begriffe abgebildet, die in direkter Beziehung zu einem realen Objekt stehen: z. B. beschreibt der Begriff "Kunde" die Rolle eines Partners, der vom Mandanten Waren oder Dienstleistungen gegen Entgelt bezieht. Die beschreibende Eigenschaft "Kunde" enthält nur die Daten, die für diese Geschäftsbeziehung notwendig sind. Dem gleichen Partner können weitere Begriffe (oder Eigenschaftsobjekte) zugeordnet werden, wie z. B. der Begriff "Lieferant". Auch können gleiche Begriffe mehrfach einem realen Objekt zugeordnet werden (z. B. verschiedene "Kunde" Eigenschaftsobjekte mit unterschiedlichen Konditionen für verschiedene Produktgruppen oder Projekte).

Einem realen Objekt werden – meist aus rein abrechnungstechnischer Absicht – gleichzeitig verschiedene Begriffe zugeordnet. Darüber hinaus repräsentieren Begriffe auch alle organisatorischen Einheiten, wie

Kostenstelle, Auftrag oder Bilanzposition. Begriffe oder Abrechnungsobjekte können als semantische Aspektklassen zu den realen Geschäftsobjekten aufgefasst werden.

Prinzipien der Modellierung

Bei Entwicklung des CyberEnterprise wurden die folgenden – teilweise konkurrierenden - Zielsetzungen beachtet:

- hohe Abstraktion (Wiederverwendbarkeit der Objekte)
- leichte Erweiterbarkeit (klar strukturiertes Modell)
- hohe Flexibilität (beliebige Kombination der Objekte)
- breite Funktionalität (intelligente Objekte)
- große Übersichtlichkeit (nicht zu viele Objekttypen)
- gute Lesbarkeit (leicht verständliches Modell)
- geringer Speicherbedarf (schlanke, performante Objekte)

Gemäß der obigen Reihenfolge gelten im CyberEnterprise Unternehmensmodell die hohe Abstraktion und die leichte Erweiterbarkeit seiner Elemente als wichtigste Prinzipien.

Die C++-Klassen für die Geschäftsobjekte aus dem CyberEnterprise Unternehmensmodell sind daher mit einem hohen Anspruch an Allgemeingültigkeit entworfen und enthalten nur die wesentlichen Datenfelder und Methoden. Sie sind trotzdem als Bausteine auch für sehr individualisierte Anforderungen geeignet, da

- sie durch dynamische Datenfelder beliebigen Typs (sog. Slots) erweiterbar sind
- mittels dieser dynamischer Datenfelder auch beliebige Relationen zwischen Objekten aufgebaut werden können
- die Berechnungsvorschriften und Auswahlregeln aus dem Programmcode in diverse Objekte verschoben sind
- das Zusammenwirken der Objekte auf verschiedene Weise mit InstantView-Anweisungen gesteuert werden kann
- besondere Algorithmen ebenfalls mit InstantView- Anweisungen formuliert werden können.

In den folgenden Abschnitten wird auf die verschiedenen Typen an Geschäftsobjekten im CyberEnterprise Unternehmensmodell eingegangen, eine zusammenfassende Übersicht der Prinzipien für eine Unternehmensmodellierung findet man in Form einer [Grammatik](#).

Geschäftsobjekte als funktionale Bausteine

Innerhalb des CyberEnterprise gibt es drei Ebenen von Objekten verschiedener Komplexität:

1. Basis-Objekte kapseln Informationen über Datum, Zeit und Werte; es sind Objekte sehr spezieller und begrenzter Funktionalität.
2. Informationsobjekte verarbeiten Attribute bzw. Merkmale von z.B. Teilen, speichern Zu- oder Abschläge bei z.B. Preisen, definieren Allokationen (Stücklisten oder Arbeitspläne), verwalten Kalendarien, beinhalten Transaktionsverwaltungsobjekte und Monitore (Konten und multidimensionale Datenwürfel), usw. In dieser Ebene findet man aus funktioneller Sicht die meisten Bausteine.
3. Geschäftsobjekte bedienen sich dann nur noch dieser Basis- und Informationsobjekte, um ihre eigenen Daten und Methoden zu verwalten. Dieser Typ von Geschäftsobjekten ist am ehesten vergleichbar mit den Entitäten traditioneller, betriebswirtschaftlicher Software.

Hinzu kommen diverse Werkzeugklassen, Klassen für Druckausgaben, für den Import und Export von Daten, Klassen für COM-Objekte, XML-Klassen, insbesondere für .docx und .xlsx Office Unterstützung, usw.

Jeder Baustein (C++ Klasse) des CyberEnterprise Unternehmensmodells definiert darüber hinaus sein Verhalten für

- die arithmetischen Operatoren (+, -, *, /).

- die Vergleichsoperatoren.
- die Konvertierung in eine Zeichenkette.
- den Import einer Zeichenkette.
- den Zugriff auf Datenfelder, Bindung der Datenfelder an Variable.
- die Erweiterung um weitere Datenfelder und Referenzen zu anderen Objekten.
- den Aufruf von Methoden.

Grundmechanismen betriebswirtschaftlicher Anwendungen

Die oben eingeführten Basis- und Informations-Objekte des CyberEnterprise stellen eine Reihe von grundsätzlichen Eigenschaften zur Verfügung, um die Modellierung betriebswirtschaftlicher Gegebenheiten zu erleichtern. Im folgenden seien einige besondere Eigenschaften exemplarisch aufgeführt:

- **Einheitenarithmetik**

Für Berechnungen steht eine Klasse (CX_VALUE) zur Verfügung, die mit einem Tupel aus Zahlenwert und Dimensionsangabe (intern repräsentiert als Primzahl) rechnet. Der Programmcode legt nicht fest, mit welchen Einheiten in einer Anwendung gerechnet wird. Diese Einheitenarithmetik ist eine Grundvoraussetzung für die im folgenden Punkt erwähnten Formel-Objekte. Der Anwender kann eigene Einheiten und Umrechnungsvorschriften definieren.

- **Berechnungsvorschriften, Bedingungstabellen**

Berechnungsvorschriften und Bedingungstabellen sollen kein Bestandteil des Programmcodes, sondern vom Anwender zu definierende Daten sein. Ein Formel-Objekt (CX_FORMULA) und ein darauf aufbauendes Objekt für Bedingungstabellen (CX_CONDITIONED_BAG) stehen dafür zur Verfügung.

- **Beträge**

Wenn in den Geschäftsobjekten Beträge vorkommen (Preis, Menge, ...), soll nicht von vornherein festgelegt werden, ob es sich um einen festen Wert handelt, ob er zu berechnen oder ob er in einer Tabelle zu suchen ist. Um all diese und weitere Möglichkeiten offenzuhalten, existiert die Klasse CX_AMOUNT, die den Betrag als eine polymorphe Abstraktion behandelt.

- **freie Definition von Merkmalen**

Aufträge, Produkte usw. werden oft durch Merkmale beschrieben, die vom Anwender festgelegt werden müssen¹. Objekte vom Typ CX_ATTRIBUTE lösen dieses Problem. Mit den oben erwähnten Formel- und Conditioned-Bag-Objekten können Berechnungen und Auswahlalgorithmen von den Merkmalswerten abhängig gemacht werden.

- **Standardwerte**

Oft wird ein Wert nur dann explizit angegeben, wenn er von einer Standardvorgabe abweicht. Der Standardwert kann eine Eigenschaft des Datenfeldes sein; es kann aber auch sein, dass der Wert von einem logisch übergeordnetem Objekt zu übernehmen ist – eine Konstruktion, die sich über beliebig viele Hierarchiestufen erstrecken kann. CyberEnterprise unterstützt einen semantisch gesteuerten Zugriff auf Datenelemente.

- **Enumerationen**

Traditionelle Anwendungen arbeiten mit einer Vielzahl von Schlüsseln und Abkürzungen und muten dem Anwender zu, diese zu kennen. Im CyberEnterprise System kann für solche Datenfelder eine Übersetzungstabelle registriert werden. Auf der Bedienoberfläche erscheint dann immer der Begriff in der aktuell eingestellten Sprache. Die Übersetzungstabellen werden vom Anwender gepflegt.

- **Aspekte**

Eine Relation zwischen zwei Objekten ist oft mit einer Bedeutung belegt, die nur dieser Beziehung eigen ist, nicht aber den beiden in Beziehung stehenden Objekten: Das referenzierte Objekt wird vom referenzierenden Objekt unter einem bestimmten Aspekt gesehen. CyberEnterprise stellt hierfür Objekte vom Typ CX_DESCRIPTIVE_REF für beschreibende Referenzen zur Verfügung.

- **Werte überschreiben**

Wenn aus Stammdaten Bewegungsdaten erzeugt werden, kommt man oft nicht mit einer Referenz auf das Stamm-Objekt aus, da die neu erzeugten Daten in einigen Datenfeldern abweichende Werte haben. Mit einer überschreibenden Referenz (CX_OVERWRITING_REF) wird für solche Fälle eine Kopie vermieden. Mit

diesem Konstrukt wird der Wert eines Datenfeldes eines Objekts davon abhängig, über welchen Navigationspfad man dieses Objekt erreicht.

- **Bedingte Referenzen**

Für Relationen zwischen Objekten, die bestimmten Bedingungen unterliegen, stellt das CyberEnterprise Objekte vom Typ CX_CONDITIONED_REF zur Verfügung.

- **Lose gekoppelte Referenzen**

Zur losen Koppelung von Informationen, wenn diese z.B. in anderen Datenbanken stehen oder durch einen externen Dienst zur Verfügung gestellt werden, können Referenzen aufgebaut werden, die automatisch zum Zeitpunkt der Informationsnachfrage frei definierbare Programmsequenzen aufrufen.

- **Gültigkeit eines Objekts**

Bisweilen stehen für einen Sachverhalt verschiedene Objekte, von denen immer eines zu einem bestimmten Zeitpunkt oder unter einer bestimmten Bedingung gültig ist. Diese Objekte können zu einem Cluster zusammengefasst werden, und eine Referenz auf ein Objekt des Clusters wird automatisch zum gerade gültigen Objekt umgeleitet.

- **Objekte unter einer beliebigen Bezeichnung suchen**

Oft werden die gleichen Dinge in verschiedenen Bereichen eines Unternehmens mit unterschiedlichen Namen bezeichnet (z.B. technisch orientiert - verkaufsorientiert). Geschäftsobjekte können in einem Dictionary-Objekt unter beliebig vielen Bezeichnungen (mehrsprachig) eingetragen und wieder aufgefunden werden.

- **Nummernkreise**

Fast alle Belege eines Unternehmens werden aus organisatorischen Gründen fortlaufend nummeriert. Auch ist es sinnvoll, Stammdaten (z. B. Artikel) mit einer eindeutigen Kennzeichnung (Nummer) zu versehen. CyberEnterprise stellt Zähler-Objekte zur Verfügung, die über Bezeichner ansprechbar sind.

- **COM-Objekte**

Zur nahtlosen Intergration von Office oder anderer OLE basierter Anwendungen können in der CyberEnterprise-Datenbank COM-Objekte direkt gespeichert und mittels dynamischer Datenfelder beliebig mit Geschäftsobjekten verbunden werden. Alle Funktionen, die ein COM-Objekt über das DISP-Interface zur Verfügung stellt, können auch mit InstantView aufgerufen werden.

¹ da sie zum Zeitpunkt der Programmentwicklung noch nicht bekannt sind

Dynamisierung des Modells durch Transaktionen

Jede Tätigkeit in einem Unternehmen, die - aus Gründen der Organisation und einer effizienteren Gestaltung - mittels der EDV abgebildet werden soll, wird durch Transaktionsobjekte beschrieben. Meist werden derartige Vorgänge von Belegen begleitet, wie Auftragsbestätigungen, Lieferscheine, Rechnungen, Materialentnahmescheine, Buchungsbelege im Rechnungswesen oder ausgehende Bestellungen.

Diese Vorgänge oder Transaktionen führen immer zu einer (Zustands-)Änderung von Geschäftsobjekten. Eine Auftragsbestätigung erhöht den Auftragsbestand, vermindert eventuell den verfügbaren Bestand eines bestellten Lagerartikels. Auch das Editieren einzelner Datenfelder (Ändern einer Artikelbezeichnung) kann mittels Transaktionsobjekte ausgeführt werden, um eine vollständige Historie der Geschäftsprozesse (in diesem Beispiel: Pflege von Artikeldaten) zu erhalten.

Meist interessiert neben den Daten jeder einzelnen Transaktion (Datum der Auftragsbestätigung, Höhe der Rechnung, Entnahmemenge eines Artikels) auch die Kumulierung einzelner Werte, und das mit bestimmten Dimensionen (z. B. Auftragseingang pro Monat und Verkaufsgebiet). Man spricht hier von einer mehr- oder multi-dimensionalen Datensicht: Der Wert "Auftragseingang" soll jeweils in der Kombination aus Periode und Verkaufsgebiet gespeichert werden, also z.B. 412.350,-€ Auftragseingang im Monat Juli 2014 im Verkaufsgebiet Nord.

Multidimensionale Daten werden in sogenannten Datenwürfeln oder Zellen gespeichert, in unserem Beispiel der Auftragseingang mit den beiden Dimensionen Periode und Verkaufsgebiet haben. Auf diese Weise findet eine parallele Verarbeitung des OLTP mit OLAP (online analytical processing) statt.

Der Transaktionsmanager

Welche Daten welcher Geschäftsobjekte verändert werden sollen oder welche Werte mit welchen Dimensionen gespeichert werden sollen, wird durch Einträge in einen Transaktionsmanager gesteuert. Jeder Vorgang besitzt seinen eigenen Transaktionsmanager.

Die automatisch entstehenden Datenwürfel - eindimensional (wenn die Periode nicht mitgezählt wird!) meist als Konten bezeichnet - registrieren die Mengen- und Wertflüsse, die durch Transaktionen erzeugt werden. Alle Dimensionen der gewünschten Datenwürfel (z. B. das Erlöskonto) müssen eindeutig aus den Daten innerhalb der Transaktion ermittelt werden können (z. B. aus dem Artikel).

Konsolidierungspfade innerhalb von Dimensionen (Umsatzstatistik auch per Kundengruppe und Jahr) liegen im Ermessen derer, die einen Vorgang mit dem Transaktionsmanager beschreiben und werden nicht automatisch generiert. Auch wenn höher aggregierte Datenwürfel nicht gleich bei jeder Transaktion fortgeschrieben werden, so lassen sich diese Daten jederzeit durch Kumulierung ermitteln. Vollständiges drill-down oder roll-up wird gewährleistet.

Spezielle Objekte des CyberEnterprise

Attribute

Im CyberEnterprise können beliebige Datenfelder dynamisch definiert und in jedem Geschäftsobjekt zusätzlich gespeichert werden (dynamische Datenfelder oder slots). Eine Definition dieser Datenfelder nimmt üblicherweise der Systemadministrator vor, in Verbindung mit Erweiterung der Anwendungen oder Bildschirmmasken, aus denen heraus diese neuen Datenfelder gepflegt werden können.

Dynamische Datenfelder werden wie normale, in den Klassen fest definierte Variablen verwendet und definieren nur den Datentyp. Die Bedeutung des Datenfeldes ergibt sich für den Anwender nur aus dem Prompt auf dem Bildschirm oder dem Titel auf einem Ausdruck.

Ein Objekt der Klasse CX_SLOT_ATTRIBUTE kapselt ein dynamisches Datenfeld und ermöglicht es somit dem Anwender weitere Informationen diesem slot hinzuzufügen: z. B. einen mehrsprachigen Namen, einen Standardwert oder auch einen Gültigkeitsbereich. Auch diese Attribut-Objekte können Geschäftsobjekten beliebig häufig hinzugefügt werden.

So kann z. B. ein Artikel zusätzlich mit den Attributen bzw. Sachmerkmalen für Farbe, Durchmesser oder Gewicht versehen werden. Dabei kann eine Standardfarbe vorgesehen werden oder ein Maximalgewicht definiert sein. Diese Angaben sind auch ohne Änderung oder Erweiterung von Anwendungen oder Bildschirmmasken möglich.

Die Verwaltung mehrerer CX_ATTRIBUTE Objekte übernimmt die Klasse CX_ATTRIBUTE_SET, die quasi eine Erweiterung eines Geschäftsobjektes darstellt. Ein solches CX_ATTRIBUTE_SET Objekt kann auch über ein CX_CONDITIONED_BAG erreicht werden, falls in Abhängigkeit bestimmter Bedingungen verschiedene Attribut-Werträume modelliert werden sollen.

Die Klasse CX_FORMULA_ATTRIBUTE ermittelt den Wert eines dynamischen Datenfeldes immer nur durch Berechnung mit einer Formel, z. B. bei dem Attribut "Fläche" aus den Attributen "Breite" und "Länge". Diese Berechnung findet immer aktuell bei Abfrage des Wertes dieses Attributes statt.

Hat ein dynamisches Datenfeld Werte, die von bestimmten Bedingungen abhängig sind, kann ein Objekt der Klasse CX_CONDITIONED_ATTRIBUTE eingesetzt werden. Diese Bedingungen können Werte anderer Attribute sein. (Beispiel: Wenn Montagepunkt = rechts oder Montagepunkt = beidseitig, dann Länge_rechts = Länge; wenn Montagepunkt = links oder Montagepunkt = beidseitig, dann Länge_links = Länge).

Strukturen

Klassifikationsmerkmale (Teileart, Artikel- oder Kundengruppen) werden als eigenständige Strukturobjekte hinterlegt und sind nicht mehr Sortierkriterium innerhalb der Objekte. Alle Geschäftsobjekte können beliebig (häufig) zugeordnet werden, wobei durch überschreibende oder beschreibende Referenzen eine Identität

geschaffen werden kann.

Strukturobjekte ihrerseits können hierarchisch zugeordnet werden.

Die Funktionalität der Geschäftsobjekte kann hierarchisch durch eine gesamte Struktur hindurch additiv aufgerufen werden (z.B. Umsätze sortiert nach Kundengruppen).

Allokationen

Allokationen sind zu verstehen als (Hilfsobjekte für) mengenmäßige Verteilungs- und/oder Zuordnungsstrukturen.

Ein Objekt der Klasse CX_SINGLE_ALLOCATION besteht im Kern aus dem Tupel aus Referenz zu einem Objekt und einer Menge: Das referenzierte Objekt wird in einer bestimmten Menge "verwendet". Die Menge ist vom Typ CX_AMOUNT (s. o.). Das referenzierte Objekt kann selbst eine Allokation zugeordnet haben, wodurch hierarchische Allokationsbäume entstehen können.

Wird ein Objekt vom Typ CX_SINGLE_ALLOCATION von einem anderen Objekt aus referenziert, dann stellt es sich diesem als "Ressource" dar.

Die Klasse CX_CONDITIONED_ALLOCATION erlaubt es, bedingte Allokationen darzustellen (Variantenverarbeitung).

Objekte vom Type CX_SET_ALLOCATION verwalten Mengen von Allokationen und repräsentieren damit Stücklisten, Arbeitspläne, aber auch Kostenumlagen. Beim Auflösen solcher Allokationen werden die Mengen an den Knoten mittels von CX_PRODUCT_AMOUNT Objekten dargestellt. Die Änderung einer Menge an irgendeinem Knoten bewirkt die sofortige Korrektur aller Mengen tiefer liegender Allokationen.

Relationen

Beziehungen bzw. Relationen zwischen Geschäftsobjekten können mit Objekten der Klassen CX_DESCRIPTOR_REF, CX_OVERWRITING_REF, CX_CONDITIONED_REF und CX_LOOSE_REF individuell verwaltet werden.

COM-Objekte

Die Klasse CX_COM_OBJECT ermöglicht es, beliebige COM-Objekte in der Datenbank zu speichern (sie besitzt eine eigene Implementierung des Interfaces ILockBytes). Alle Funktionen des DISP-Interface können - wie jede andere Funktion eines Geschäftsobjektes - auch mit InstantView aufgerufen werden.

Von CX_COM_OBJECT abgeleitete Klassen für bestimmte COM-Objekte - z.B. Microsoft-Word-Dokumente - benutzen das C++-Interface des speziellen COM-Objekts und können deshalb bestimmte Operationen mit höherer Performance durchführen.