Inhaltsverzeichnis

1 Kurzbeschreibung

Die **CyberSpace Foundation** (CSF) stellt in Form einer Klassenbibliothek *grundlegende Abstraktionen* (Basisklassen) zur Realisierung objektorientierter Systeme bereit.

Im Vordergrund steht dabei der *logisch* notwendige Bedarf zur Realisierung informationsverarbeitender Systeme. Existierende Systemkomponenten wie graphische Oberflchen, Datenbanken usw. finden darin Eingang, ohne notwendigerweise deren jeweils zugrundeliegendes Denkmodell zu reflektieren.

Darberhinaus wird die Entkopplung verschiedener Systemaspekte erleichtert, indem die Prsenz von Objekten in verschiedenen Objektmodellen (*logischen Sichten*) sowie die Transformation zwischen solchen Sichten untersttzt wird.

Die CSF ist offen konzipiert, so da die Koexistenz mit anderen Klassenbibliotheken nicht nur mglich ist, sondern bereits in der CSF selbst Untersttzung findet.

Die **Code Creation Components** ($\mathcal{C}_3$) sind ein offenes bersetzersystem, das die Grundlage fr deskriptive Sprachen oder Modelle zur Code-Generierung fr die CSF implementiert. $\mathcal{C}_3$ basiert selbst auf der CSF .

Die Offenheit ermglicht Spracherweiterungen und die Anpassung des erzeugten Codes, sowie die Koexistenz mit anderen Generatoren. Damit kann ein evolutives Wachstum des Systems stattfinden, etwa in Form projektspezifischer Anpassungen.

$\mathcal{C}_3$ ist nicht an die Verwendung im Zusammenhang mit der CSF gebunden und kann fr verschiedenste Zwecke verwendet werden.

2 Aspekte der CSF

- Das *Object Framework* der CSF dient als homogene Kapselung von Systemdienstleistungen und Funktionalitten anderer Applikationsobjekte. Damit wird die Modellbildung untersttzt, indem zentrale Grundbegriffe vorformuliert werden.

  Ein Object Framework ist die Basis der logischen Sicht eines Objektes auf seine Umgebung (*Object World Model*).

  Mit diesem Modell soll im Unterschied zu *Application Frameworks* die Gleichheit aller Systemkomponenten (Betriebssystem, Applikationen) betont werden, und damit die Homogenitt in der Darstellung dieser Komponenten.

  Damit soll ein Nachteil ausgegelichen werden, der daraus entsteht, da viele heute bestehende Klassenbibliotheken (zunchst gezwungenermaen) auf dem Fundament der existierenden *Service-Einteilung* der DV-Welt aufbauen (DB, GUI), so da Thematik-bergreifende Abstraktionen nur unzureichend gebildet werden knnen.

- *Encapsulation Enhancement* bietet die Grundlage einer Strukturierung und Flexibilisierung von *Objekthllen*, die ber das durch statische Vererbung gegebene Ma hinausgeht.

  Dadurch wird eine Entkopplung von Systemaspekten ermglicht, indem Objekte in unterschiedliche Umgebungen (Objektmodelle als *logische Sichten*) eingebettet werden knnen. Darberhinaus wird die Transformation zwischen solchen Sichten untersttzt.

  Im Bereich des *Object Oriented Design* wird als Ausweg aus den durch statische oder zu groe Modelle entstehenden Problemen zu einer Konzentration auf die Anwendung hin geraten. Dadurch wird ein Objektmodell jedoch nicht nur *objekt-*, sondern eben auch *anwendungsorientiert*. Dies ist zwar hinsichtlich Klarheit und Zweckmigkeit des Modells wnschenswert, nicht jedoch unter Gesichtspunkten der Wiederverwendbarkeit von Objekten, einer zentralen Versprechung der OO.

  Koexistenz mehrerer logischer Objektsichten kann hier Abhilfe schaffen. Ferner erffnet sich durch Koexistenz Freiraum bei der Systemevolution,

da *Objektmodellrestrukturierungen* bzw. *Objektmodellmigration* davon abhngen knnen.

- Die CSF -Klassen definieren logisch richtige Orte zur redundanzfreien Spezifikation von Systemeigenschaften. Diese Eigenschaft ist eine Folge der Kapselung in OO-Systemen.

3 Aufbau von $\mathcal{C}_3$

$\mathcal{C}_3$ besteht aus einer Reihe von Elementen, die ein flexibles, offenes bersetzersystem bilden.

Basierend auf der CSF kann als Input des Systems prinzpiell jede Form von Daten Verwendung finden. Die erste Version beinhaltet einen Parser, der sprachliche Beschreibungen aus Textbausteinen zur Verwendung der anderen $\mathcal{C}_3$ -Komponenten aufbereitet. Die Verwendung von Daten aus Datenbanken, einfachen Data Dictionaries oder graphischen Daten ist mglich.

Die Syntax der verwendeten Sprache ist an C++ angelehnt. Die Textbausteine knnen sowohl getrennt vorliegen als auch in C++-Kommentaren eingebettet sein, die Extraktion aus anderen Quellen kann natrlich prinzipiell entsprechend einfach vorgenommen werden.

Die nchste Stufe in der Verarbeitung durch $\mathcal{C}_3$ besteht aus der Transformation der Eingabedaten in eine logische Sicht, auf der die Ausgabekomponenten operieren knnen.

Die Ausgabe wird durch Textskelette beschrieben, in die C++-Code eingestreut wird. Durch Einbindung vorkonfigurierter Header-Dateien kann die Anbindung dieser Skelette an die aufbereiteten $\mathcal{C}_3$ -Daten erfolgen.

Durch rekursive Anwendung des Code-Generierungs-Schemas ist die Formulierung sowohl der Daten-Transformation als auch des Ausgabeformates in problemadquateren Sprachen mglich.

Das Einfgen der generierten Daten in bestehenden Code wird durch weitere Hilfsmittel bewirkt. Diese ermglichen auch die Regenerierung des erzeugten Codes.

Alle genannten Komponenten von $\mathcal{C}_3$ knnen auch getrennt Verwendung finden oder ausgetauscht werden, wodurch die Offenheit des Systems erreicht wird.

Durch den flexiblen Aufbau von $\mathcal{C}_3$ und die damit verbundene leichte Integrierbarkeit neuer Sprachelemente wird die problemadquate Formulierbarkeit von Systemaspekten mglich, so da der Wiederverwendbarkeitsgrad dieser Informationen steigt und damit die Investitionssicherheit.

4 Die CSF -Komponenten

Bei der Konzeption der CSF wurde versucht, alle relevanten Aspekte der Informationsmodellierung in einem Modell zu integrieren. Sie beinhaltet daher viele bekannte Begriffe, jedoch mit teils modifizierten, teils przisierten Definitionen. Ziel war es, ein praktikables Grundgerst zu erstellen, auf dem effektiv gebaut werden kann.

Technische Erfordernisse aus „tieferen“ Abstraktonsschichten wurden soweit als mglich auf ihre logischen Grundlagen zurckgefhrt.

Die CSF ist in der vorliegenden Form (aus historischen Grnden) an C++ ausgerichtet und soll auch zunchst in dieser Sprache realisiert werden.

4.1 Object Framework

Die folgende bersicht beschreibt die Themenschwerpunkte des *Object Framework* (OF). Aus den erluterten Begriffen leiten sich die Klassen des OF ab.

4.1.1 Object Location

Object Location umschreibt die Problematik der logischen Orte von Objekten.

The Space Als Oberbegriff fr alle mglichen logischen Objektorte bietet sich der Begriff des *CyberSpace* an, dem die CSF ihren Namen verdankt. Dieser Raum kann in *Gebiete* unterteilt werden, deren Modellierung lohnenswert erscheint. Etwa knnen bestimmten Rumen *kontextuelle Informationen* zugeordnet werden, dazu ist die Objektprsenz des Raumes als Zuordnungspunkt ntzlich.

Persistence Die Problematik der Objektpersistenz kann hinter der Ortsabstraktion verborgen werden. Dies kommt der Tatsache entgegen, da persistente Objekte zunehmend in tieferen Systemschichten verfgbar werden.

4.1.2 Basic Object Relations

Zwischen bestimmten Objekten knnen Beziehungen bestehen, die die Zuordnung von Identit und persistenzrelevanten Daten unbersichtlich machen.

Solche Probleme sollten mglichst hinter geeigneten Abstraktionen verborgen bleiben, bisweilen ist der Umgang damit jedoch unumgnglich.

Identity, Original and Copy Die *Identitt* eines Objektes wird logisch durch es selbst reprsentiert. Trotz (scheinbarer) Gleichheit von *Original* und *Kopie* sind diese verschieden.

Pointer *Zeiger* sind die Modellierung einer Relation, bei der das Interesse oder die Verantwortlichkeit fr die Relation nur einseitig besteht. Zeiger knnen auch als Zwischenstufe auf der Skala von einem Originalobjekt hin zu einer getrennt existierenden Kopie aufgefat werden, auf dieser Skala befinden sich auch „live-links“.

Resource Management *Resourcen* zu verwalten gehrt ebenfalls in diesen Bereich. Es ist noch offen, ob die Modellierung dieses Themas in der CSF erfolgen mu oder ausreichend durch die anderen Abstraktionen abgedeckt ist.

4.1.3 Object Identification

Indices & Identification, Search Criteria Zur Referenzierung bisher nicht im Zugriff befindlicher Objekte dienen *Merkmale*; falls diese eindeutig ein Objekt bezeichnen mssen, heien sie *Identifikationsmerkmale*. Dieses Begriffe stecken zusammen mit den Ortsklassen und den Mengen (s.u.) den Datenbankbereich ab.

4.1.4 Communication

Unter *Kommunikation* wird hier sowohl Inter-Objekt-Kommunikation als auch Mensch-Objekt-Kommunikation verstanden. Dem liegt eine Bild zugrunde, da Kommunikation als den Austausch von in einer bestimmten Sprache verfaten Nachrichten beschreibt, was den Bogen von Netzwerken bis hin zu graphischen Benutzeroberflchen spannt.

Object Presentation Objekt*darstellung* ist in diesem Sinne alles, was einen inhaltlichen Zusammenhang zu einem Objekt hat, jede Form von Abbildung des Objektes. Charakteristisch dafr ist in der Regel, da die Transformation von Objekt zu Darstellung mglich ist, der umgekehrte Weg, also die Konstruktion des Objektes aus der Darstellung nicht.

Darstellungen finden in verschiedenster Form Verwendung, etwa zur einfachen Vermittlung des Objektzustandes an einen Betrachter der Darstellung, oder aber auch zur Interaktion in Form der Bereitstellung eines Bezugssystems fr Merkmale („das rote Objekt auf dem Bild...“).

Messages, Operations, Transactions *Nachrichten*, *Operationen* und *Transaktionen* vervollstndigen den Bereich der Kommunikation um entsprechende, offensichtlich notwendige Aspekte.

4.1.5 Structure

Grundbegriffe der Informationsverarbeitung ergeben sich bekanntermaen auch aus der Unterscheidung zwischen Einheit und Vielheit, sowie aus Beziehungen zwischen Objekten. Diese Thematik zieht sich als roter Faden durch die gesamte Entwicklungsgeschichte der Programmierung, beginnend bei Feldern und Zeigern bis hin zu relationalen Modellen, Mengen, Containern, Assoziationen.

Diese Begriffe sind hinlnglich bekannt und werden hier nur kurz erwhnt.

Multitudes *Vielheiten* sind Objekte, die aus mehreren Elementen bestehen. ber ihren inneren Aufbau, die Existenz von Ordnungen, ist im allgemeinsten Fall nichts ausgesagt.

Relations *Relationen* modellieren Beziehungen zwischen Objekten. Ihre Verwendung ist vielfltig.

4.2 Encapsulation Enhancement

Encapsulation Enhancement ist die zweite wesentliche Komponente der CSF .

4.2.1 Object Shell, Object Kernel

Die Objekthlle wird in objektorientierten Systemen durch die Schnittstellen der Klasse des Objektes und deren Basisklassen beschrieben.

Weiteren Freiraum bei der Gestaltung der Schnittstellen und dem inneren Aufbau des Objektes zu schaffen, erfordert zumindest in der Implementierung weitergehende Strukturierungsmglichkeiten der Objekthlle. Hier

sei stichwortartig die Realisierung einer „MayBeA“- als potentielle „IsA“-Beziehung genannt, oder die Implementierung von *Delegations*-Objekten.

Darber hinaus sollte eine Trennung verschiedener Betrachtungsebenen, (mehrere logische oder auch technische) eines Objektes vorgenommen werden knnen, wobei die Identitt des Objektes erhalten bleiben mu.

Diese Ziele verfolgt die Modellierung von *Objektkernen* und -*Hllen*, sowie die Funktionalitt der *Aspekttransformation*.

4.2.2 Object Models

Objektmodelle explizit als Objekte verfgbar zu machen ist ein Schritt hin zu dynamischen Modellen und der Untersttzung von Transformationen.

Classes and Objects *Klassen* und *Objekte* stellen die Grundlage fr eine Modellierung von Objektmodellen dar.

Views and Aspects Objektmodelle knnen auch als *logische Sichten* auf andere Objektmodelle verwendet werden. Die Zugrundeliegende Transformation kann dabei beliebiger Natur sein. Ein Spezialfall hiervon wren etwa logische Sichten in Datenbankmodellen.

Das Thema, unter dem eine logische Sicht erstellt wird, wird in diesem Zusammenhang hier als *Aspekt* bezeichnet.

A Applikationen

Als Pilotanwendungen der CSF seien hier stichwortartig zwei Vorschlge genannt:

- MIS-Applikation
- Editorplattform

Die MIS-Applikation wrde von den Darstellungsobjekten und der Mglichkeit unterschiedlicher Aspekte profitieren, whrend die Editorplattform die prototypische Verwaltung von mit der CSF realisierten Objekten demonstrieren knnte.