



KfK 4916
September 1991

VDAFS-STEP-Konvertierung

**Ein Beitrag zum Übergang zu einem
neuen Produktdatenstandard**

H. Heinrich
Institut für Reaktorentwicklung

Kernforschungszentrum Karlsruhe

Kernforschungszentrum Karlsruhe

Institut für Reaktorentwicklung

KfK 4916

VDAFS - STEP - Konvertierung

Ein Beitrag zum Übergang

zu einem neuen Produktdatenstandard

Heike Heinrich

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Zusammenfassung

Dieser Bericht beinhaltet sowohl einen Abbildungsvorschlag von VDAFS auf STEP als auch die Beschreibung der dazugehörigen Konvertierungssoftware.

Die Aufgabe, VDAFS- in STEP-Files zu übertragen, wurde im Zusammenhang mit dem NIRO Projekt (ESPRIT Projekt 2614) interessant, weil sie die Benutzung vorhandener VDAFS-Files gestattet.

Abschnitt 2 beinhaltet eine Erläuterung der neutralen Schnittstellen VDAFS und STEP.

Abschnitt 3 stellt den Abbildungsvorschlag vor. Um die VDAFS-Entities auf eine entsprechende Menge in STEP abzubilden, wurden die Schemata `geometry_schema` und `topology_schema` ausgewählt.

Abschnitt 4 gibt einen Überblick über die abgebildeten Entities.

Abschnitt 5 beschreibt verwendete Softwaresysteme und die entsprechenden Verweise.

Abschnitt 6 stellt die beschriebene Lösung in einen Zusammenhang zur VDAFS-VDAIS-Konvertierung.

Abschnitt 7 gibt einen Überblick über die implementierte Software.

Abschnitt 8 beinhaltet ein Abbildungsbeispiel bestehend aus zwei VDAFS-Files und einem STEP-File.

Abschnitt 9 faßt offene Fragen zusammen.

Die beschriebenen Arbeiten entstanden als Gemeinschaftsarbeit des Instituts für Informatik und Rechentechnik und des Kernforschungszentrums Karlsruhe in Zusammenhang mit dem ESPRIT Projekt NIRO.

Abstract

VDAFS - STEP - Conversion

A Contribution to the Transition to a new Product Data Standard

This report includes a mapping proposal of VDAFS to STEP as well as the description of the corresponding converter software.

The task to transfer VDAFS to STEP files became interesting in connection with the NIRO project (ESPRIT project 2614) for it allows the use of existing VDAFS files.

Chapter 2 includes an illustration of the neutral interfaces VDAFS and STEP.

Chapter 3 introduces the mapping proposal. To map the VDAFS entities onto a similar set of entities in STEP the schemes `geometry_schema` and `topology_schema` were selected.

Chapter 4 gives an overview of the transferred entities.

Chapter 5 describes the used software systems.

Chapter 6 puts the described solution into the context of VDAFS-VDAIS conversion.

Chapter 7 gives an overview of the implemented software.

Chapter 8 includes an example consisting of two VDAFS files and one STEP file.

Chapter 9 gives a summary of open questions.

The described work was done as a joint effort of Institut für Informatik und Rechentechnik and Kernforschungszentrum Karlsruhe in the context of ESPRIT project NIRO.

Vorwort

Von August 1990 bis Juli 1991 nutzte ich die Möglichkeit eines Gastwissenschaftleraufenthalts an der Kernforschungszentrum Karlsruhe GmbH.

Meine Aufgabe hier bestand in der Erstellung eines Konverters zwischen den neutralen CAD-Schnittstellen VDAFS und STEP.

Obwohl dies ein eigenständiges Thema war, konnten diese Arbeiten in laufende Entwicklungen am CAD/CAM-Labor integriert werden.

Ebenfalls enge Beziehungen bestanden zum NIRO-Projekt, da auch in diesem Umfeld Implementierungen zu STEP erfolgen. Speziell betrifft das die Verwendung von Software zum Lesen der STEP-Files.

Im Zusammenhang mit der Definition der Abbildung setzte ich mich mit den Abbildungsvorschlägen, die innerhalb des CADEX-Projektes erarbeitet worden waren, auseinander. Den dort entwickelten STEP-Parser nutzte ich zur Überprüfung der Syntax.

Ich möchte diese Gelegenheit nutzen, allen Mitarbeitern des IRE-6 und des CAD/CAM-Labors für ihre hilfreiche Unterstützung zu danken.

Mein besonderer Dank gilt Herrn Dr. Schlechtendahl für seine fachliche Beratung, Herrn Haas, Frau Karl und Herrn Welte für die zur Verfügung gestellte Software sowie Herrn Dr. Ferrero und Herrn Becker für ihre Hinweise im Bereich der Freiformflächenmodellierung.

Inhaltsverzeichnis

	Seite
1. Einleitung	1
2. Beschreibung der neutralen Interfaces	1
2.1. VDAFS	1
2.2. STEP	3
3. Abbildungsvorschlag für die VDAFS-Entities	5
3.1. Inhaltliche Gesichtspunkte des Abbildungsvorschlags	5
3.2. Abbildungsübersicht VDAFS 1.0	6
3.3. Abbildungsübersicht VDAFS 2.0	6
4. Realisierung der Abbildung	7
4.1. Übersicht	7
4.2. Formatbeschreibung CLF	8
4.3. realisierte Entities VDAFS -> CLF	9
4.4. realisierte Entities CLF -> STEP	9
4.5. realisierte Entities STEP -> CLF	10
4.6. realisierte Entities CLF -> VDAFS	10
5. Verwendete Software	10
5.1. EXAKON	10
5.2. STEP-Parser	11
6. Einbindung in die VDAIS-Konvertierung	11
7. Programmbeschreibung	12
7.1. Übersicht über Programme und Moduln	12
7.2. Beschreibung der VMS-Shell	13
7.3. Moduln aus CLFSTEP	13
7.4. Moduln aus STEPCLF	14
8. Abbildungsbeispiel	16
9. Mögliche Erweiterungen	22
Literaturverzeichnis	23

1. Einleitung

Seit 1984 wird innerhalb der ISO (International Standardisation Organisation) vom Komitee ISO/TC184/SC4 an der Entwicklung eines internationalen Standards für technische Produkte gearbeitet. Ziel der Definition dieses künftigen Standards STEP (Standard for the External Representation of Product Definition Data) ist die Spezifikation eines einheitlichen Informationsmodells, das alle relevanten Daten innerhalb des Lebenszyklus' eines Produktes beinhaltet /ISO3/.

Aus dieser Definition sind verschiedene Realisierungen ableitbar. Eine davon ist der Austausch über eine sequentielle Datei /ISO2/. Weitere Darstellungsformen sind Programmierschnittstellen und Datenbank.

Die Vorteile des Datenaustausches über neutrale Schnittstellen wurden seit Anfang der 70er Jahre erkannt. Bekannte nationale Standards wie IGES (Initial Graphic Exchange Specification), VDAFS (VDA-Flächenschnittstelle) und SET (Standard d'Exchange et de Transfert) stellen sequentielle Fileformate dar, die dazu dienen, technische Zeichnungen oder geometrische Modelle auszutauschen, Aufgaben, die von STEP natürlich auch abgedeckt werden müssen.

Die VDAFS wurde vom CAD/CAM-Arbeitskreis des VDA erarbeitet, um den Austausch von Flächendaten zu ermöglichen. Es erfolgte bewußt eine Einschränkung auf geometrische Elemente. Sie ist im DIN-Standard 66301 (Ausgabe Juli 1986) spezifiziert.

Aus Erfahrungen mit VDAFS 1.0 entstand 1983 die Version 2.0, die es zusätzlich gestattet, Topologie- und Strukturinformationen zu übertragen.

Da die Nutzung dieser neutralen Files in der Industrie verbreitet ist (alle bekannten CAD-Systeme verfügen über IGES- und VDAFS-Prozessoren), ist es eine zentrale Aufgabe, ihre Abbildung in das STEP-Modell zu spezifizieren. Arbeiten in dieser Richtung werden innerhalb des ESPRIT-Projektes CADEX ausgeführt /CAD1/. Hier erfolgte eine Gegenüberstellung von STEP-Entities mit VDAFS und VDAIS (eine vom VDA definierte Teilmenge von IGES).

Aufgaben in dieser Richtung wurden auch am Kernforschungszentrum Karlsruhe bearbeitet. Sie hatten einmal die Spezifikation der Abbildung zum Ziel und desweiteren die Implementierung eines Konvertierungsprogramms von VDAFS nach STEP. Inhalt des vorliegenden Berichts ist die Beschreibung dieses Programmpakets.

2. Beschreibung der neutralen Interfaces

2.1. VDAFS

Die Definition der VDAFS /VDA1/ diente in ihrer ersten Version vorrangig dem Austausch von Geometriedaten zur Freiformflächenbeschreibung. Dieses Ziel spiegelt sich in der Entitymenge der VDAFS 1.0 wider. Definiert wurden Spliniekurven und -flächen (CURVE bzw. SURF) mit Hilfe der polynomialen Splinedarstellung.

Kurven werden durch die globalen Parameterwerte an den jeweiligen Anfangs- und Endpunkten der Segmente und die Koeffizienten pro Segment bestimmt. Die Koeffizienten entsprechen der Darstellungsform mit lokalem Parameter.

$$x(u) = \sum_{i=0}^{k-1} ax_i u^i$$

$$y(u) = \sum_{i=0}^{k-1} ay_i u^i$$

$$z(u) = \sum_{i=0}^{k-1} az_i u^i$$

u: lokaler Parameter, $u \in [0,1]$
 k: Ordnung (Polynomgrad-1)
 ax_i : Koeffizienten für x-Komponente
 ay_i : Koeffizienten für y-Komponente
 az_i : Koeffizienten für z-Komponente

Flächen werden durch die globalen Parameterwerte an den jeweiligen Patchrändern und die Koeffizienten pro Patch bestimmt. Die Koeffizienten entsprechen der Darstellungsform mit lokalen Parametern.

$$x(u,v) = \sum_{i=0}^{k-1} \sum_{j=0}^{l-1} ax_{ij} u^i v^j$$

$$y(u,v) = \sum_{i=0}^{k-1} \sum_{j=0}^{l-1} ay_{ij} u^i v^j$$

$$z(u,v) = \sum_{i=0}^{k-1} \sum_{j=0}^{l-1} az_{ij} u^i v^j$$

u: lokaler Parameter, $u \in [0,1]$
 v: lokaler Parameter, $v \in [0,1]$
 k: Ordnung (Polynomgrad+1) in u-Richtung
 l: Ordnung (Polynomgrad+1) in v-Richtung
 ax_{ij} : Koeffizienten für x-Komponente
 ay_{ij} : Koeffizienten für y-Komponente
 az_{ij} : Koeffizienten für z-Komponente

Weitere Entities sind der Punkt (POINT), die Punktmenge (PSET) und eine Punkt-Vektor-Folge (MDI). Letztere Entities dienen unter anderem dem Übertragen von Kontrolldaten. VDAFS 1.0 gestattet eine Zusammenfassung von Entities mit Hilfe des SET-Konzepts. Hierbei kann an inhaltlich zusammenhängende Entities die gleiche Bezeichnung vergeben werden.

Mit der Erweiterung auf VDAFS 2.0 wurden die Möglichkeiten der Übertragung von Topologie- und Strukturinformationen mit aufgenommen. Zur Topologiebeschreibung dienen die Entities CONS, FACE und TOP. CONS beschreibt eine Flächenkurve, die einmal in primärer Darstellung im Parameterraum und zweitens als Raumkurve gegeben ist. FACE ist eine von Kurven begrenzte Fläche und TOP ein Flächenverband.

GROUP ist ein Strukturentity, das eine assoziative Zusammenfassung von Elementen beschreibt.

Ein Kreisbogen kann in VDAFS 2.0 durch das Entity CIRCLE abgebildet werden.

TMAT definiert eine Transformationsmatrix, diese wird mit Hilfe des Entity's TLIST mit den zu transformierenden Elementen in Zusammenhang gebracht.

POINT: Punkt im Raum
PSET: Punktmenge
MDI: Punkt- Vektorfolge
CURVE: polynomiale Splinekurve
SURF: polynomiale Splinefläche
CIRCLE: Kreisbogen
CONS: Flächenkurve
FACE: von Flächenkurven begrenztes Flächenstück
TMAT: Transformationsmatrix
TLIST: Zusammenfassung geometrischer Entities mit gemeinsamer Transformationsmatrix
GROUP: Zusammenfassung beliebiger Entities

2.2. STEP

Das STEP-Produktmodell wird mit Hilfe der Spezifikationsprache EXPRESS /ISO1/ in einzelnen Schemata definiert. Die für die Abbildung von VDAFS relevanten Schemata sind `geometry_schema`, `topology_schema` und `shape_representation_schema`, die man in Teil 42 der STEP-Dokumentation finden kann /ISO3/.

Aufgrund der großen Anzahl der Entities wird im folgenden nur auf die zur Abbildung benötigten eingegangen.

b_spline_curve: zwei- oder dreidimensionale rationale B-Splinekurve, Anfangs- und Endparameter entsprechen denen im Knotenvektor

Die rationalen B-Splinekurven werden in STEP durch die Angabe des Knotenvektors, der Gewichte und der de Boor-Punkte definiert.

$$U = (u_0, u_1, \dots, u_n, \dots, u_{n+k})$$

$$K(u) = \frac{\sum_{i=0}^n \beta_i b_i N_i^k(u)}{\sum_{i=0}^n \beta_i N_i^k(u)}$$

n: oberer Summenindex
k: Ordnung (Grad+1)
U: Knotenvektor
 β_i : Gewichte
 b_i : de Boor-Punkte

b_spline_surface: rationale B-Splinefläche, Anfangs- und Endparameter entsprechen denen der Knotenvektoren

Die rationalen B-Splineflächen werden in STEP durch die Angabe der Knotenvektoren, der Gewichte und der de Boor-Punkte definiert.

$$U = (u_0, u_1, \dots, u_n, \dots, u_{n+k})$$

$$V = (v_0, v_1, \dots, v_n, \dots, v_{m+l})$$

$$F(u,v) = \frac{\sum_{i=0}^n \sum_{j=0}^m \beta_{ij} b_{ij} N_i^k(u) N_j^l(v)}{\sum_{i=0}^n \sum_{j=0}^m \beta_{ij} N_i^k(u) N_j^l(v)}$$

- n: oberer Summenindex in u-Richtung
- m: oberer Summenindex in v-Richtung
- k: Ordnung (Grad + 1) in u-Richtung
- l: Ordnung (Grad + 1) in v-Richtung
- U: Knotenvektor in u-Richtung
- V: Knotenvektor in v-Richtung
- β_{ij} : Gewichte
- b_{ij} : de Boor-Punkte

cartesian_point: Punkt im zwei- oder dreidimensionalen Raum, der durch seine drei (zwei) Koordinaten beschrieben wird

composite_curve_on_surface: aus Flächenkurven zusammengesetzte Kurve

curve_bounded_surface: Fläche, die von Flächenkurven begrenzt wird

curve_logical_structure: Orientierungsflag für Kurven

direction: Richtungsvektor, gleicher syntaktischer Aufbau wie cartesian_point

edge: topologische Kante, definiert durch Eckpunkte und eine geometrische Kurve

edge_logical_structure: Orientierungsflag für edge

edge_loop: geschlossener Kantenzug von Entities vom Typ edge

face: topologisches Flächenstück, definiert durch geschlossene Kantenzüge und eine geometrische Fläche

geometric_3d_curve_set: Menge von Kurven und Punkten im Raum, wird im folgenden als Punktmenge verwendet

index_entry: weist einem Entity eine Bezeichnung (String) zu, dient der Erstellung eines Entityverzeichnisses auf dem File

loop_logical_structure: Orientierungsflag für loop

pcurve: Flächenkurve, die durch eine parametrische Fläche und eine Kurve im Parameterraum definiert ist

surface_curve: Flächenkurve, die sowohl durch eine Parameterkurve als auch durch eine Raumkurve beschrieben wird

surface_logical_structure: Orientierungsflag für Flächen

trimmed_curve: Kurve, deren Anfangs- und Endpunkt in Form von Parameterwerten oder Kurvenpunkten vorgegeben sind

vertex: topologischer Eckpunkt

3. Abbildungsvorschlag für die VDAFS-Entities

3.1. Inhaltliche Gesichtspunkte des Abbildungsvorschlags

Für einige der Entities aus VDAFS 1.0 gibt es ein korrespondierendes STEP-Entity. Das trifft auf POINT, CURVE und SURF zu. Hierbei ist zu erwähnen, daß Start- und Endparameterwerte einer Kurve (Fläche) im VDAFS im jeweiligen Entity integriert sind, STEP verlangt in diesem Fall die Verwendung einer trimmed_curve (trimmed_surface).

Für die Entities PSET und MDI gibt es keine eindeutigen Entsprechungen in STEP. Es wird vorgeschlagen, PSET auf geometric_3d_curve_set mit leerer Kurvenliste abzubilden, da dieses Entity eine Zusammenfassung von Punkten und Kurven beschreibt.

MDI wird auf ein nutzereigenes Entity abgebildet, da mögliche andere Konstruktionen ebenfalls Vereinbarungen erfordern.

VDAFS-Entity- und -SET-Namen werden als String zusammengefaßt und innerhalb eines index_entry abgelegt.

Die Abbildung von CIRCLE aus VDAFS 2.0 auf trimmed_curve und circle ist eindeutig.

In Zusammenhang mit der Topologie entstehen einige Fragen. Will man topologische Zusammenhänge in dem Umfang abbilden, den die VDAFS Spezifikation erlaubt, so erfordert das Entity TOP das Schema topology_schema. Daraus ergibt sich die Verwendung der Entities face, edge_loop, edge und vertex. VDAFS beschreibt den Zusammenhang von Kurven und Flächen mit Hilfe des Konzeptes der Flächenkurve. Eine derartige Kurve wird durch die primäre Darstellung im Parameterraum, die sekundäre Darstellung als Raumkurve und die Fläche definiert. Da die Zusammenhänge innerhalb des STEP-Topologieschemas anders definiert sind, bringt die Übertragung dieser Informationen eine komplizierte Datenstruktur mit sich.

Wir schlagen deshalb auch eine (zusätzliche) zweite Variante der Abbildung von Topologieinformationen vor. Die Entities surface_curve, composite_curve_on_surface und curve_bounded_surface aus dem STEP-Geometrieschema entsprechen dem oben erwähnten Konzept. Verzichtet man auf das Entity TOP, dann kann FACE auf curve_bounded_surface abgebildet werden.

Die Entsprechung von TMAP und transformation ist wieder eindeutig.

3.2. Abbildungsübersicht VDAFS 1.0

POINT	cartesian_point
PSET	geometric_3d_curve_set cartesian_point
MDI	!mdi (nutzereigenes Entity) cartesian_point
CURVE	trimmed_curve b_spline_curve cartesian_point
SURF	b_spline_surface cartesian_point
SET	index_entry

3.3. Abbildungsübersicht VDAFS 2.0

POINT	cartesian_point
PSET	geometric_3d_curve_set cartesian_point
MDI	!mdi (nutzereigenes Entity) cartesian_point
CURVE	trimmed_curve b_spline_curve cartesian_point
SURF	b_spline_surface cartesian_point
SET	index_entry
CIRCLE	trimmed_curve circle axis2_placement cartesian_point
CONS	cartesian_point surface_curve b_spline_surface cartesian_point trimmed_curve b_spline_curve cartesian_point pcurve b_spline_surface cartesian_point trimmed_curve b_spline_curve cartesian_point

FACE (1. Variante, Abbildung in das Schema topology_schema)

```
face
  loop_logical_structure
    edge_loop
      edge_logical_structure
        edge
          vertex
            cartesian_point
          curve_logical_structure
            surface_curve
              b_spline_surface
                cartesian_point
              trimmed_curve
                b_spline_curve
                  cartesian_point
            pcurve
              b_spline_surface
                cartesian_point
              trimmed_curve
                b_spline_curve
                  cartesian_point
```

FACE (2. Variante, Abbildung in das Schema geometry_schema)

```
curve_bounded_surface
  composite_curve_on_surface
    b_spline_surface
      cartesian_point
    surface_curve
      b_spline_surface
        cartesian_point
      trimmed_curve
        b_spline_curve
          cartesian_point
    pcurve
      b_spline_surface
        cartesian_point
      trimmed_curve
        b_spline_curve
          cartesian_point
```

```
TOP      connected_face_set
         face
         ...
TMAT     transformation
         direction
         cartesian_point
TLIST    transformation
GROUP    !group (nutzereigenes Entity)
```

4. Realisierung der Abbildung

4.1. Übersicht

Die Realisierung der Konvertierung VDAFS <-> STEP erfolgte in Abstimmung mit laufenden Arbeiten am CAD/CAM-Labor. Dort ist für die Transformation von VDAFS nach VDAIS und für Genauigkeitsprüfungen ein neutrales Fileformat CLF (CAD/CAM-Labor

Format) definiert worden, das als Schnittstelle zwischen den beiden Standards dient /LUT1/. Die CLF-Entities entsprechen inhaltlich denen aus der VDAIS-Spezifikation. So werden Freiformelemente in der B-Splinedarstellung beschrieben.

Zur Übertragung von VDAFS nach STEP wurde ebenfalls das Zwischenformat CLF verwendet. Das bietet die Möglichkeit einer Kopplung mit VDAIS und damit der Übertragung von Daten zwischen VDAIS und STEP.

Desweiteren bot sich damit an, schon vorhandene Moduln zur Geometriekonvertierung zu nutzen.

Für genauere Informationen hierzu sei auf das Programmpaket EXAKON verwiesen, das die Kopplung VDAFS <-> CLF realisiert /KAR1/.

Für das Lesen der VDAFS-Files wird korrektes VDAFS-Format vorausgesetzt. Die Rückabbildung, d.h. das Lesen der STEP-Files erfolgt mit Hilfe von Scanner/Parser-Techniken, syntaktisch falsche Files können deshalb erkannt werden. Benutzt wurde dafür die Anpassung des CAD*I-Parsers an STEP /HAA1/.

4.2. Formatbeschreibung CLF

Ziel der Definition des CL-Formats war die gemeinsame Darstellung von VDAFS- und VDAIS-Elementen. Inhaltlich lehnt sich CLF an die VDAIS-Entities an, wobei die Daten der Sektionen directory entry section und parameter data section übernommen wurden. Syntaktisch gesehen erfolgte eine Vereinfachung der komplizierten Verzeigerung des IGES-Formats. Es werden sowohl die VDAFS als auch die VDAIS-Bezeichnungen verwendet.

Jedes Entity besteht aus einem Header, welcher Namen, einen eventuellen Verweis auf eine Transformationsmatrix und Daten aus der Sektion directory entry section enthält, und einem Parameterteil.

Der Fileheader wurde aus dem VDAFS-Format übernommen.

Wegen der leichteren Verarbeitung von INTEGER-Zahlen wurde jedem CLF-File ein INFO-File zugeordnet. Dieser enthält die wichtigsten Daten des Headers in INTEGER-Form. CLF- und INFO-File sind als eine Einheit zu betrachten.

Die CLF-Definition enthält zur Zeit folgende Entities:

CLF-Typ	VDAIS-Typ	Element
1	100	Circular Arc
2	102	Composite Curve
3	104	Conic Arc
4	106/2	Copious Data
	106/3	Copious Data
5	108	Plane
6	110	Line
7	112	Parametric Spline Curve
8	114	Parametric Spline Surface
9	116	Point
10	118	Ruled Surface
11	120	Surface of Revolution
12	122	Tabulated Cylinder
13	124	Transformation Matrix
14	402/7	Group
15	126	Rational B-Spline Curve
16	128	Rational B-Spline Surface
17	142	Curve on Surface
18	144	Trimmed Surface

4.3. Realisierte Entities VDAFS -> CLF

Im Programm EXAKON /KAR1/ sind die unten angegebenen Abbildungen realisiert. Das VDAFS-Entity CONS ist eine einfache Raumkurve, während das CLF-Entity Curve on Surface eine aus mehreren Segmenten zusammengesetzte Raumkurve repräsentiert. Deshalb erfolgt die Abbildung von CONS im allgemeinen auf Rational B-Spline Curve. CONS-Entities, die nicht von einer FACE referenziert werden, werden auf Curve on Surface abgebildet.

POINT	->	Point
PSET	->	Copious Data (106/2)
MDI	->	Copious Data (106/3)
TMAT	->	Transformation Matrix
CURVE	->	Rational B-Spline Curve
SURF	->	Rational B-Spline Surface
CONS	->	Rational B-Spline Curve
	->	Curve on Surface
FACE	->	Face

4.4. Realisierte Entities CLF -> STEP

Die Abbildung der topologischen Entities Curve on Surface und Face erfolgt in den beiden oben erwähnten Varianten in die Schemata geometry_schema bzw. topology_schema.

Das Element Curve on Surface wird, wenn es aus einem CONS-Entity entstanden ist, auf surface_curve abgebildet, anderenfalls auf curve_bounded_surface bzw. edge_loop.

Soweit wie möglich werden geometrisch identische Kurven, die sich nur durch unterschiedliche Start- und Endparameter unterscheiden, mit Hilfe des Konzepts der trimmed_curve zusammengefaßt.

Für die Entity- und die SET-Namen aus VDAFS wird ein index_entry erzeugt.

Für die Entities b_spline_curve, b_spline_surface, mdi und geometric_3d_curve_set ist es sinnvoll, das SCOPE Konzept zu nutzen. Die Verwendung von SCOPE Bereichen ist optional.

Point	->	cartesian_point
Copious Data (106/2)	->	geometric_3d_curve_set
Copious Data (106/3)	->	mdi
Rat. B-Spline Curve	->	trimmed_curve
		b_spline_curve
Rat. B-Spline Surface	->	b_spline_surface

1. Variante für Topologie (Abbildung in das Schema geometry_schema)

Curve on Surface	->	composite_curve_on_surface
	->	surface_curve
Face	->	curve_bounded_surface

2. Variante für Topologie (Abbildung in das Schema topology_schema)

Curve on Surface -> edge_loop
-> surface_curve
Face -> face

4.5. Realisierte Entities STEP -> CLF

Die Entities `geometric_3d_curve_set` und `trimmed_curve` werden nur in den hier relevanten Spezialfällen verarbeitet.

Ebenso wird ein Entity vom Typ `index_entry` überlesen, falls es nicht mögliche VDAFS-Entity- und -SET-Namen beinhaltet.

Verweisen mehrere `trimmed_curve`-Entities auf dieselbe geometrische Kurve, so entstehen im CLF entsprechend der VDAIS-Spezifikation mehrere Entities vom Typ Rational B-Spline Curve.

`cartesian_point` -> Point
`geometric_3d_curve_set` -> Copious Data (106/2)
(für Spezialfälle)
`!mdi` -> Copious Data (106/3)
`b_spline_curve` -> Rational B-Spline Curve
`b_spline_surface` -> Rational B-Spline Surface
`trimmed_curve` -> Rational B-Spline Curve
(falls Verweis auf `b_spline_curve`)
`index_entry` Eintrag der Namen in das jeweilige Entity

4.6. Realisierte Entities CLF -> VDAFS

Point -> POINT
Copious Data (106/2) -> PSET
Copious Data (106/3) -> MDI
Rational B-Spline Curve -> CURVE
Rational B-Spline Surface -> SURF

5. Verwendete Software

5.1. EXAKON

Das Programmpaket EXAKON /KAR1/ dient der Übertragung von Daten aus VDAFS nach CLF und umgekehrt. Aufgrund der unterschiedlichen Splinedarstellung sind dazu Konvertierungsprogramme notwendig.

Die Überführung der polynomialen Darstellung in die B-Splinedarstellung erfolgt dabei über die Bezierdarstellung also in zwei Schritten. Ergebnis sind integrale B-Splines, die Gewichte in der rationalen Darstellung werden 1 gesetzt.

Mathematisch gesehen bilden die Bezierbasisfunktionen und die Monome bis zu dem entsprechenden Grad zwei Basissysteme desselben Raumes (Polynome mit festem Grad, hier Kurvensegmente bzw. Flächenpatches). Daher ist die Konvertierung Bezierdarstellung <-> Monomdarstellung eine lineare Transformation der Koeffizienten. Die Segmente (Patches)

werden unabhängig voneinander betrachtet, die Konvertierung erfolgt segment- bzw. patchweise.

In der B-Splinedarstellung werden die Segmente (Patches) als Einheit angesehen. Übergangsbedingungen an den Rändern sind durch die B-Splinebasisfunktionen vorgegeben. Bezierkurven und -flächen treten hierbei als der Spezialfall auf, daß keinerlei Übergangsbedingungen gefordert sind. Für die Basisfunktionen bedeutet das maximale Knotenvielfachheit. Die Überführung der B-Spline- in die Bezierdarstellung bedeutet also eine Erhöhung der Knotenvielfachheit bis zur maximalen Anzahl. Dafür wird der Algorithmus von Böhm verwendet. Bei entsprechenden Stetigkeitsbedingungen kann durch Inversion dieses Algorithmus' eine Knotenreduzierung erfolgen.

Bezüglich der erwähnten mathematischen Probleme sei auf /HOS1/ verwiesen.

5.2. STEP-Parser

Zum Lesen des STEP-Files wird der STEP-Parser verwendet /HAA1/. Das Lesen erfolgt in den zwei Schritten Front-End und Back-End.

Voraussetzung dafür war die Erstellung der Parsertabellen unter Nutzung des Tabellengenerators TABGEN. Die dafür verwendete STEP-Grammatik wurde mit der Grammatik auf der Grundlage der NIRO-STEP-Spezifikation abgestimmt.

Der Name des Parsers ist STPARS. Er überprüft einen STEP-File auf syntaktische Korrektheit und überführt ihn in eine Direktzugriffsdatei (**Treefile**). Diese wird durch die Extension STTREE gekennzeichnet. Fehlermitteilungen erscheinen in der Datei STPARS.MESS.

Folgende Back-End-Routinen wurden genutzt:

NPSRCT: liest den Inhalt eines Treefiles

READ: liest ein Entity, dies kann sequentiell oder im Direktzugriff erfolgen

GET: analysiert ein einzelnes Token innerhalb eines Entity's

6. Einbindung in die VDAIS-Konvertierung

Die Transformation VDAFS <-> VDAIS ist am CAD/CAM-Labor über die Schnittstelle CLF realisiert worden. Die Anbindung von VDAIS an CLF wurde von Beatrice Schwarzer und Gerhard Welte implementiert. Hierbei können die VDAIS-Entities:

Circular Arc	100
Composite Curve	102
Conic Arc	104
Copious Data	106/2
Copious Data	106/3
Plane	108
Line	110
Parametric Spline Curve	112
Parametric Spline Surface	114
Point	116
Ruled Surface	118
Surface of Revolution	120
Tabulated Cylinder	122
Transformation Matrix	124
Group	402/7
Rational B-Spline Curve	126
Rational B-Spline Surface	128

und die CLF-Entities:

Composite Curve	2
Copious Data	4
Point	9
Transformation Matrix	13
Group	14
Rational B-Spline Curve	15
Rational B-Spline Surface	16
Curve on Surface	17
Trimmed Surface	18

konvertiert werden. Damit ergibt sich die Möglichkeit, eine Kopplung von STEP mit VDAIS in dem oben angegebenen Entityumfang vorzunehmen.

7. Programmbeschreibung

7.1. Übersicht über Programme und Moduln

Das Programmpaket STEP besteht aus vier Komponenten, die unabhängig voneinander ausführbare Programme darstellen. Es sind dies EXAKON, CLFSTEPG, CLFSTEPT, STPARS und STEPCLF.

EXAKON realisiert die Abbildungen zwischen VDAFS und STEP. Dabei werden die Filenamen der zu übertragenden Files interaktiv abgefragt. Für eine genaue Beschreibung dieses Programmpakets verweisen wir auf /KAR1/.

CLFSTEPG und CLFSTEPT sind unterschiedliche Versionen der Konvertierung von CLF auf STEP, von deren Moduln sich nur CLFSTEP.PAS unterscheidet. CLFSTEPG überträgt die topologischen Entities in das Geometrieschema und CLFSTEPT in das Topologieschema. Diese beiden Programme enthalten je die Moduln:

```
ENVIRONMENT.PAS
STEP.PAS
CLFSTEP.PAS
CLFIN.PAS
LOWLEVEL.PAS
BOEING.PAS
SERVICE.PAS
PRINTSTEP.PAS.
```

STPARS beinhaltet das Front-End des Scanner-Parsers und liest demzufolge STEP-Files, die in einen Treefile übertragen werden. Man findet eine genaue Beschreibung der Arbeitsweise des Parsers in /HAA1/.

Das Programm CLFSTEP liest den vom Parser erzeugten Treefile und erzeugt daraus einen CLF-File. Es besteht aus folgenden Moduln:

```
STEPCLF.FOR
STEPIN.FOR
CLFFOUT.FOR
```

und enthält auch die Front-End-Routinen zum Lesen des Treefiles. Die Beschreibung der Prozeduraufufe befindet sich im Anhang B.

7.2. Beschreibung der VMS-Shell

Zur einfacheren Handhabung der einzelnen Programmkomponenten wurde die VMS-Shell geschrieben. Sie besteht aus den Kommandoprozeduren:

STEP.COM
HEADER.COM
SUBLIB.COM
PARSE.COM

und wird durch den Aufruf @STEP gestartet.

Die Namen der Ein- und Ausgabefiles werden von der Kommandoprozedur abgefragt, sie übernimmt auch die Kontrolle der Existenz dieser Files und das Löschen von Files, die in Zwischenschritten entstehen.

STEP.COM greift auf die ausführbaren Programme:

EXAKON.EXE
CLFSTEPG.EXE
CLFSTEPT.EXE
STEPCLF.EXE
STPARS.EXE

und den externen File

KEYWORD.TBL

zu.

Voraussetzung für den Aufruf von STEP.COM ist die Definition der logischen Namen STEP\$ROOT und STEP\$COMMAND. STEP\$ROOT bezeichnet das Directory mit den ausführbaren Programmen, STEP\$COMMAND das Directory mit den Kommandoprozeduren HEADER.COM, SUBLIB.COM und PARSE.COM.

Der File KEYWORD.TBL muß sich im selben Verzeichnis befinden wie die Prozedur STEP.COM. Im allgemeinen sollte dies ein Testdirectory sein, das auch die zu übertragenden Files enthält.

Fehlermitteilungen der Abbildung von VDAFS nach STEP werden im File FEHLER1.TXT ausgegeben, die der Rückrichtung im File FEHLER2.TXT.

7.3. Moduln aus CLFSTEP

ENVIRONMENT.PAS

Der File ENVIRONMENT.PAS enthält Konstanten-, Typ- und Variablendefinitionen, die für die Module von globalem Interesse sind. Das betrifft Konstanten für die Felddimensionierung und für die Ausgabe im CL-Format, Typ- und Variablendefinitionen für CLF-Entities. Ebenfalls werden in ENVIRONMENT.PAS externe Routinen vereinbart.

STEP.PAS

Dieser File enthält den Rahmen, der die beiden Routinen clf_step und printstep ruft, sowie eine diesbezügliche Abfrage.

CLFSTEP.PAS

Aufgrund der lokalen Bedeutung werden Typen für STEP-Entities in CLFSTEP.PAS definiert.

Die Routine `clf_step` öffnet und schließt Ein- und Ausgabefiles und ruft die Routine `map_all_elements`, welche die Steuerung der Abbildung CLF -> STEP übernimmt.

Für die CLF-Entities Copious Data (106/2), Copious Data (106/3), Point, B-Spline Surface und Face gibt es je eine Routine zur Abbildung dieser Entities, für B-Spline Curve zwei in Abhängigkeit von der Planarität und für Curve on Surface ebenfalls zwei abhängig davon, ob eine Referenz zu einem Entity Face besteht. Kennzeichen der Namen dieser Funktionen ist der Präfix `map_`.

Pro STEP-Entity gibt es eine Funktion zum Schreiben dieses Entity's. Die Namen dieser Funktionen beginnen mit `add_`. Ausnahmen bilden wegen ihrer einfachen Struktur die Entities `cartesian_point` und `vector`.

Die Felder `ccurlist` und `maplist` gemeinsam mit ihren Zugriffsroutinen `el` und `ccur` dienen dem Speichern von Strukturinformationen.

CLFIN.PAS

Pro CLF-Entity gibt es eine Lesefunktion. Diese Funktionen sind im File CLFIN.PAS enthalten und namentlich mit dem Präfix `read_` gekennzeichnet.

LOWLEVEL.PAS

Die Entity-orientierten Schreibfunktionen für das STEP-Format greifen auf Schreibfunktionen auf niedrigem Level zu. Diese Routinen wurden von Antje Lorenz im Rahmen eines Praktikums erstellt. Ihre Verwendung erfordert den File `KEYWORD.TBL`, der die Schlüsselwörter für die STEP-Entities enthält.

BOEING.PAS

Der Modul `BOEING.PAS` enthält mathematische Berechnungen, die einem Programmpaket zur Splinekonvertierung entnommen wurden. Entwickelt worden ist diese Software von Michael Zanger innerhalb eines Praktikums.

SERVICE.PAS

Die Routine `error_messages` enthält Mitteilungen bei möglichen Fehlerfällen.

PRINTSTEP.PAS

Durch Aufruf der Routine `clf_step` wird eine STEP-Datei im 80 Zeichen Rekord Format erzeugt. Dabei werden die STEP-Drucksteuerzeichen verwendet. Mit Hilfe der Routine `printstep` in `PRINTSTEP.PAS` wird ein STEP-File entsprechend seiner Drucksteuerzeichen in lesbare Form transformiert.

7.4. Moduln aus STEPCLF

STEPCLF.FOR

Zur Konvertierung von STEP nach CLF werden keine globalen Variablen verwendet, da die eingeschränkte Entitymenge das nicht erfordert. `STEPCLF.FOR` enthält sowohl Anweisungen zum Filehandling als auch die Steuerung der Abbildung. Bei einem ersten Filedurchlauf werden potentielle VDAFS-Entity- und -SET-Namen gelesen und die referenzierten Entitynamen gespeichert. Die Weiterverarbeitung erfolgt dann im Direktzugriff. Übertragen werden nur Entities, für die auch ein Name im STEP-File vorhanden ist.

Analog zur Übertragung CLF -> STEP greift `STEPCLF` pro STEP-Entity auf je eine Lese- und pro CLF-Entity auf je eine Schreibfunktion zu.

Zum Verwalten von Strukturinformationen dienen die Felder `IMPLST`, `IMCLST`, `INMLST`, `MNLST` und `IENLST` sowie die Routinen `INSPOINT`, `INSCURVE` und `FNDNM`.

STEPIN.FOR

Pro STEP-Entity gibt es eine Routine zum Lesen dieses Entity's. Kennzeichen ihrer Namen ist der erste Buchstabe R. Zum Lesen wird auf die Back-End-Routine GET zugegriffen.

CLFFOUT.FOR

Für die CLF-Entities dienen die Schreibroutinen aus CLFFOUT.FOR, die mit dem ersten Buchstaben C markiert wurden. Die Routine CHEAD erzeugt automatisch einen Fileheader für den CLF-File.

8. Abbildungsbeispiel

Die folgenden drei Testfiles ANV.DAT, ANV.STEP und ANV.VDA dienen zur Veranschaulichung der Abbildung. ANV.DAT ist ein VDAFS-File, der Entities der Version 1.0 entht. Mit Hilfe der Programme EXAKON und CLFSTEP wurde daraus der STEP-File ANV.STEP erzeugt. Die Rckrichtung ergibt den VDAFS-File ANV.VDA.

Ein Vergleich zeigt, daa sowohl Entitynamen als auch Parameter (innerhalb der Genauigkeitsgrenze 10^{-8}) bereinstimmen. Da die Rckabbildung der SET-Konstruktion im Programm EXAKON noch nicht realisiert ist, kann diese Information zur Zeit nicht zurckgewonnen werden.

Die Reihenfolge der Entities wird nicht beibehalten.

VDAFS-Datei ANV.DAT

```
ANVIL5K =HEADER/ 20 00000005
*****00000010
VDAFS VERSION      : 2.0 00000015
----- ANGABEN UEBER DEN ABSENDER -----00000020
SENDERFIRMA       : ----- 00000025
ANSPRECHPARTNER   : ----- 00000030
    -TELEFON      : ----- 00000035
    -ADRESSE      : ----- 00000040
ERZUEGENDES SYSTEM : ----- 00000045
ERZUEGUNGSDATUM   : ----- 00000050
SENDE-FILENAME    : ----- 00000055
----- ANGABEN UEBER DAS TEIL -----00000060
PROJEKT           : ----- 00000065
OBJEKTKENNUNG     : ----- 00000070
VARIANTE          : ----- 00000075
VERTRAULICHKEIT   : ----- 00000080
GUELTIGKEITSDATUM : ----- 00000085
----- ANGABEN UEBER/FUER DEN EMPFAENGER -----00000090
EMPFAENGERFIRMA   : ----- 00000095
EMPFAENGERNAME/ABT.: ----- 00000100
----- TOLERANZ-ANGABEN -----00000105
$$ 00000110
$$-----00000115
$$ 00000245
LEVL0000=BEGINSET 00000250
SUR0001 =SURF/ 3,1,0. , 3.333333333333333D-01, 6.666666666666666D-01, 00000255
1. ,0. ,1. , 00000260
3,3,-1.000000000000000D+02,-1.492139745096210D-13, 00000265
5.000000000000000D+01, 0.000000000000000D+00, 3.244155399061988D+02, 00000270
-2.433116549296490D+02, 0.000000000000000D+00,-2.494155398887570D+02, 00000275
1.870616549165677D+02,-6.975736996017265D-14, 1.788854381999832D+02, 00000280
-6.045922471664844D+01,-6.015811206768552D+01,-6.562036077287654D+02, 00000285
2.312990293663325D+02, 8.015811206768562D+01, 5.973181694448915D+02, 00000290
-2.208398046086144D+02, 0.000000000000000D+00, 0.000000000000000D+00, 00000295
0.000000000000000D+00, 1.984311850358137D+02, 6.047071110515390D-08, 00000300
-4.031392109027364D-08, 1.568814964186231D+00,-1.209413085234700D-07, 00000305
8.062761480687186D-08, 00000310
3,3,-5.000000000000000D+01, 1.000000000000000D+02, 00000315
-5.684341886080801D-14, 8.110388497654986D+01,-1.622077699530992D+02, 00000320
0.000000000000000D+00,-6.235388497218935D+01, 1.247077699443784D+02, 00000325
0.000000000000000D+00, 1.184262134833347D+02, 5.796698876668640D+01, 00000330
```

-5.796698876668637D+01, -4.850626904301183D+02, -1.936055489860848D+02, 00000335
1.936055489860848D+02, 4.566364769039625D+02, 1.556385602076314D+02, 00000340
-1.556385602076313D+02, 0.000000000000000D+00, 0.000000000000000D+00, 00000345
0.000000000000000D+00, 1.984311850559705D+02, 2.557953848736361D-13, 00000350
-2.842170943040400D-13, 1.568814923872537D+00, -3.410605131648480D-13, 00000355
2.273736754432320D-13, 00000360
3,3, 4.999999999999992D+01, 1.000000000000000D+02, 00000365
-5.000000000000008D+01, -8.110388497654936D+01, -1.622077698332747D+02, 00000370
2.433116546899994D+02, 6.235388497218909D+01, 1.247077698245539D+02, 00000375
-1.870616546769183D+02, 1.184262134833347D+02, -5.796698876668628D+01, 00000380
-6.045922471664858D+01, -4.850626904301183D+02, 1.936055489959643D+02, 00000385
2.312990293666044D+02, 4.566364769039625D+02, -1.556385602275264D+02, 00000390
-2.208398046088864D+02, 0.000000000000000D+00, 0.000000000000000D+00, 00000395
0.000000000000000D+00, 1.984311850559705D+02, 2.015687528000853D-08, 00000400
-4.031375056001706D-08, 1.568814923872480D+00, -4.031369371659820D-08, 00000405
8.062716005952097D-08, 00000410
C0921001=CURVE/ 1,0.,1., 00000415
4, 1.000000000000000D+02, -1.500000000000000D+02, -1.500000000000000D+02, 00000420
1.000000000000000D+02, 2.000000000000000D+01, 2.400000000000000D+02, 00000425
-2.400000000000000D+02, 0.000000000000000D+00, 2.000000000000000D+02, 00000430
0.000000000000000D+00, 0.000000000000000D+00, 0.000000000000000D+00, 00000435
P0101001=POINT/ -1.000000000000000D+02, 2.000000000000000D+01, 00000440
2.000000000000000D+02, 00000445
P0101002=POINT/ -5.000000000000000D+01, 1.000000000000000D+02, 00000450
2.000000000000000D+02, 00000455
P0101003=POINT/ 5.000000000000000D+01, 1.000000000000000D+02, 00000460
2.000000000000000D+02, 00000465
P0101004=POINT/ 1.000000000000000D+02, 2.000000000000000D+01, 00000470
2.000000000000000D+02, 00000475
L0201001=CURVE/ 1, 0.000000000000000D+00, 1.000000000000000D+00,2, 00000480
1.000000000000000D+02, -2.000000000000000D+02, -1.000000000000000D+01, 00000485
0.000000000000000D+00, 1.000000000000000D+02, 0.000000000000000D+00, 00000490
LEVLO000=ENDSET 00000520
ANVIL5K =END 00000525

STEP-Datei ANV.STEP

```
STEP;
HEADER;
FILE_NAME('INFILE:', '19910624T145627', ('Heinrich'),
('KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH', 'KARLSRUHE FRG'),
'STEP VERSION 1.0', 'VDAFS PROZESSOR VDAFS VERSION 1', 'VDAFS-FILE');
FILE_DESCRIPTION(
('THIS FILE WAS CREATED BY VDAFS-STEP PROCESSING'), 'LEVEL 1.0');
ENDSEC;
DATA;
#1=CARTESIAN_POINT(-1.00000E+02, 2.00000E+01, 2.00000E+02);
#2=INDEX_ENTRY('P0101001', #1);
#3=CARTESIAN_POINT(-5.00000E+01, 1.00000E+02, 2.00000E+02);
#4=INDEX_ENTRY('P0101002', #3);
#5=CARTESIAN_POINT( 5.00000E+01, 1.00000E+02, 2.00000E+02);
#6=INDEX_ENTRY('P0101003', #5);
#7=CARTESIAN_POINT( 1.00000E+02, 2.00000E+01, 2.00000E+02);
#8=INDEX_ENTRY('P0101004', #7);
#9=CARTESIAN_POINT(-1.00000E+02, -6.97574E-14, 0.00000E+00);
#10=CARTESIAN_POINT(-1.00000E+02, -3.00791E+01, 9.92156E+01);
#11=CARTESIAN_POINT(-1.00000E+02, 2.00000E+01, 2.00000E+02);
#12=CARTESIAN_POINT(-1.00000E+02, 8.94427E+01, 0.00000E+00);
#13=CARTESIAN_POINT(-1.88961E+01, -1.04687E+02, 9.92156E+01);
#14=CARTESIAN_POINT(-6.25000E+01, 8.00000E+01, 2.00000E+02);
#15=CARTESIAN_POINT(-5.00000E+01, 1.18426E+02, 0.00000E+00);
#16=CARTESIAN_POINT(-9.44806E+00, -1.24105E+02, 9.92156E+01);
#17=CARTESIAN_POINT(-3.12500E+01, 9.00000E+01, 2.00000E+02);
#18=CARTESIAN_POINT(-4.97380E-14, 1.47410E+02, 0.00000E+00);
#19=CARTESIAN_POINT( 2.00728E-13, -1.43523E+02, 9.92156E+01);
#20=CARTESIAN_POINT( 4.51195E-13, 1.00000E+02, 2.00000E+02);
#21=CARTESIAN_POINT( 5.00000E+01, 1.18426E+02, 0.00000E+00);
#22=CARTESIAN_POINT( 9.44806E+00, -1.24105E+02, 9.92156E+01);
#23=CARTESIAN_POINT( 3.12500E+01, 9.00000E+01, 2.00000E+02);
#24=CARTESIAN_POINT( 1.00000E+02, 8.94427E+01, 0.00000E+00);
#25=CARTESIAN_POINT( 1.88961E+01, -1.04687E+02, 9.92156E+01);
#26=CARTESIAN_POINT( 6.25000E+01, 8.00000E+01, 2.00000E+02);
#27=CARTESIAN_POINT( 1.00000E+02, -1.99840E-13, 0.00000E+00);
#28=CARTESIAN_POINT( 1.00000E+02, -3.00791E+01, 9.92156E+01);
#29=CARTESIAN_POINT( 1.00000E+02, 2.00000E+01, 2.00000E+02);
#30=B_SPLINE_SURFACE(2,2,6,2, ((#9,#10,#11)
, (#12,#13,#14), (#15,#16,#17), (#18,#19,#20), (#21,#22,#23), (#24,#25,#26), (#27,
#28,#29)),
$, $,
4,2,(3,2,2,3),
(3,3),
( 0.00000E+00, 3.33333E-01, 6.66667E-01, 1.00000E+00),
( 0.00000E+00, 1.00000E+00),
$, $, $, $);
#31=INDEX_ENTRY('SUR0001 ', #30);
#32=CARTESIAN_POINT( 1.00000E+02, 2.00000E+01, 2.00000E+02);
#33=CARTESIAN_POINT( 5.00000E+01, 1.00000E+02, 2.00000E+02);
#34=CARTESIAN_POINT(-5.00000E+01, 1.00000E+02, 2.00000E+02);
#35=CARTESIAN_POINT(-1.00000E+02, 2.00000E+01, 2.00000E+02);
#36=B_SPLINE_CURVE(3,3, (#32,#33,#34,#35),
$, 2,(4,4),
( 0.00000E+00, 1.00000E+00),
$, $, $, $);
```

```
#37=TRIMMED_CURVE(#0, 0.00000E+00, 1.00000E+00,$,$,.T.,$);
#38=INDEX_ENTRY('C0921001',#37);
#39=CARTESIAN_POINT( 1.00000E+02,-1.00000E+01, 1.00000E+02);
#40=CARTESIAN_POINT(-1.00000E+02,-1.00000E+01, 1.00000E+02);
#41=B_SPLINE_CURVE(1,1,(#39,#40),
$,2,(2,2),
( 0.00000E+00, 1.00000E+00),
$,,$,$);
#42=TRIMMED_CURVE(#0, 0.00000E+00, 1.00000E+00,$,$,.T.,$);
#43=INDEX_ENTRY('L0201001',#42);
ENDSEC;
ENDSTEP;
```

VDAFS-Datei ANV.VDA

```
VDAFS = HEADER      / 20                                00000010
*****00000020
VDAFS VERSION      : 1.0                                00000030
----- ANGABEN UEBER DEN ABSENDER -----              00000040
SENDEFIRMA        : KERNFORSCHUNGSZENTRUM KARLSRUHE    00000050
ANSPRECHPARTNER   : HEIKE HEINRICH                    00000060
  -TELEFON        : 07247 823093                       00000070
  -ADRESSE        : IRE, POSTFACH 3640, 75 KARLSRUHE    00000080
ERZEUGENDES SYSTEM : STEP-VDAFS KONVERTER             00000090
ERZEUGUNGSDATUM   : 24-JUN-91                         00000100
SENDE-FILENAME    :                                    00000110
----- ANGABEN UEBER DAS TEIL -----                  00000120
PROJEKT           :                                    00000130
OBJEKTKENNUNG     :                                    00000140
VARIANTE          :                                    00000150
VERTRAULICHKEIT   : ALLGEMEIN ZUGAENGLICH             00000160
GUELTIGKEITSDATUM :                                    00000170
----- ANGABEN UEBER/FUER DEN EMPFAENGER -----        00000180
EMPFAENGERFIRMA   :                                    00000190
EMPFAENGERNAME/ABT.:                                  00000200
----- TOLERANZ-ANGABEN -----                        00000210
SURO001 = SURF / 3, 1,                                00000220
  0.000000000000000D+00, 0.333332985639572D+00, 0.666666984558105D+00, 00000230
  0.100000000000000D+01, 0.000000000000000D+00, 0.100000000000000D+01, 00000240
    3, 3,
-0.100000000000000D+03, 0.000000000000000D+00, 0.500000000000000D+02, 00000260
  0.000000000000000D+00, 0.324415600000000D+03, -0.243311720000000D+03, 00000270
  0.000000000000000D+00, -0.249415600000000D+03, 0.187061720000000D+03, 00000280
-0.697574000000000D-13, 0.178885400000000D+03, -0.604594000000001D+02, 00000290
-0.601581999999999D+02, -0.656202400000000D+03, 0.231298600000000D+03, 00000300
  0.801581999999999D+02, 0.597317000000000D+03, -0.220839200000000D+03, 00000310
  0.000000000000000D+00, 0.000000000000000D+00, 0.000000000000000D+00, 00000320
  0.198431200000000D+03, 0.000000000000000D+00, 0.000000000000000D+00, 00000330
  0.156880000000000D+01, 0.000000000000000D+00, 0.000000000000000D+00, 00000340
    3, 3,
-0.500000000000000D+02, 0.999999999999999D+02, 0.140332190312620D-12, 00000360
  0.811038800000000D+02, -0.162207759999999D+03, -0.522248910783674D-12, 00000370
-0.623538800000000D+02, 0.124707760000000D+03, 0.300204305858642D-12, 00000380
  0.118426000000000D+03, 0.579680000000000D+02, -0.579680000000000D+02, 00000390
-0.485062000000000D+03, -0.193608000000000D+03, 0.193608000000000D+03, 00000400
  0.456636000000000D+03, 0.155640000000000D+03, -0.155640000000000D+03, 00000410
  0.000000000000000D+00, 0.000000000000000D+00, 0.000000000000000D+00, 00000420
  0.198431200000000D+03, 0.000000000000000D+00, 0.000000000000000D+00, 00000430
  0.156880000000000D+01, 0.000000000000000D+00, 0.000000000000000D+00, 00000440
    3, 3,
  0.500000000000000D+02, 0.100000000000000D+03, -0.500000000000000D+02, 00000460
-0.811038800000000D+02, -0.162207840000000D+03, 0.243311720000000D+03, 00000470
  0.623538800000000D+02, 0.124707840000000D+03, -0.187061720000000D+03, 00000480
  0.118426000000000D+03, -0.579666000000000D+02, -0.604594000000002D+02, 00000490
-0.485062000000000D+03, 0.193605200000000D+03, 0.231298600000000D+03, 00000500
  0.456636000000000D+03, -0.155638600000000D+03, -0.220839200000000D+03, 00000510
  0.000000000000000D+00, 0.000000000000000D+00, 0.000000000000000D+00, 00000520
  0.198431200000000D+03, 0.000000000000000D+00, 0.000000000000000D+00, 00000530
  0.156880000000000D+01, 0.000000000000000D+00, 0.000000000000000D+00, 00000540
C0921001 = CURVE / 1,                                  00000550
  0.000000000000000D+00, 0.100000000000000D+01, 00000560
```

```

      4,
0.1000000000000000D+03, -0.1500000000000000D+03, -0.1500000000000000D+03, 00000570
0.1000000000000000D+03, 0.2000000000000000D+02, 0.2400000000000000D+03, 00000580
-0.2400000000000000D+03, 0.0000000000000000D+00, 0.2000000000000000D+03, 00000590
0.0000000000000000D+00, 0.0000000000000000D+00, 0.0000000000000000D+00, 00000600
L0201001 = CURVE / 1, 00000610
0.0000000000000000D+00, 0.1000000000000000D+01, 00000620
      2, 00000630
0.1000000000000000D+03, -0.2000000000000000D+03, -0.1000000000000000D+02, 00000640
0.0000000000000000D+00, 0.1000000000000000D+03, 0.0000000000000000D+00, 00000650
P0101001 = POINT / 00000660
-0.1000000000000000D+03, 0.2000000000000000D+02, 0.2000000000000000D+03, 00000670
P0101002 = POINT / 00000680
-0.5000000000000000D+02, 0.1000000000000000D+03, 0.2000000000000000D+03, 00000690
P0101003 = POINT / 00000700
0.5000000000000000D+02, 0.1000000000000000D+03, 0.2000000000000000D+03, 00000710
P0101004 = POINT / 00000720
0.1000000000000000D+03, 0.2000000000000000D+02, 0.2000000000000000D+03, 00000730
VDAFS = END 00000740
00000750
```

9. Mögliche Erweiterungen

In der vorliegenden Version ist nicht die gesamte Entitymenge aus VDAFS 2.0 berücksichtigt worden. Das betrifft vor allem die Strukturentities GROUP, TMAP und TLIST. Das Problem bei GROUP besteht darin, daß es innerhalb von STEP kein Entity gibt, das eine derartige assoziative Zusammenfassung beschreibt. Man könnte sich Konstruktionen mit Hilfe des SCOPE-Konzepts vorstellen /AAS1/ oder auch die Definition eines nutzeigenen Entity's analog wie es zur Abbildung von MDI erfolgt ist (vgl. hierzu 2.3). Die Transformationsmatrix wäre problemlos abbildbar, ihre Anwendung auf geometrische Entities mit Hilfe von TLIST ist ähnlich schwierig abzubilden. Vorgeschlagen wird hier eine Anwendung der Transformation oder die Zusammenfassung unter einem nutzeigenen Entity.

Die Erstellung des Programmes STEPCLF diene dem Zyklustest für Geometriedaten. Es wurde deshalb auch nur eine VDAFS 1.0 entsprechende Entitymenge übertragen. Eine weitere Einschränkung ist die Forderung der Existenz von alphanumerischen Namen, hier wäre eine automatische Generierung denkbar.

a das Programm EXAKON in der Rückrichtung SET-Konstruktionen nicht verarbeitet, kann diese Information nicht wiedergewonnen werden. Es sei in diesem Zusammenhang darauf hingewiesen, daß das STEP-Modell kein Äquivalent hierzu kennt und somit die Rückabbildung nicht allgemeingültig möglich ist.

Literaturverzeichnis

- /AAS1/ J. Aas, Studie zur Abbildung von CAD-Modellen auf STEP
unveröffentlichter Bericht
August 1990
- /HAA1/ S. Haas, Programmdokumentation
unveröffentlichter Bericht
Juni 1991
- /HEI1/ H. Heinrich, Studie zur Abbildung von VDAFS auf STEP
unveröffentlichter Bericht
Oktober 1990
- /HOS1/ J. Hoschek, D. Lasser
Grundlagen der geometrischen Datenverarbeitung
B. G. Teubner Stuttgart 1989
- /ISO1/ ISO CD 10303-11
Product Data Representation and Exchange - Part 11
ISO/TC184/SC4/N83
The EXPRESS Language Reference Manual
April 1991
- /ISO2/ ISO CD 10303-21
Product Data Representation and Exchange - Part 21
Clear Text Encoding of the Exchange Structure
ISO/TC184/SC4/N78
March 1991
- /ISO3/ ISO CD 10313-42
Product Data Representation and Exchange - Part 42
General Resources: Shape Representation
April 1991
- /KAR1/ S. Karl, Programmdokumentation
unveröffentlichter Bericht
März 1991
- /LUT1/ R. Lutz, Beschreibung des CL-Formats
unveröffentlichter Bericht
Juli 1991
- /VDA1/ VDA/VDMA VDA-Flächenschnittstelle (VDAFS) Version 2.0
Frankfurt-Schwanheim
Januar 1987