



**Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH**

**Document**  
D-95-06

**Integriertes Produktmodell  
für Behälter aus Faserverbundwerkstoffen**

**Markus Steffens, Ansgar Bernardi**

**Juni 1995**

**Deutsches Forschungszentrum für Künstliche Intelligenz  
GmbH**

Postfach 20 80  
67608 Kaiserslautern, FRG  
Tel.: (+49 631) 205-3211/13  
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3  
66123 Saarbrücken, FRG  
Tel.: (+49 681) 302-5252  
Fax: (+49 681) 302-5341

# Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, SEMA Group, and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Computer Linguistics
- Programming Systems
- Deduction and Multiagent Systems
- Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Dr. Dr. D. Ruland  
Director

# **Integriertes Produktmodell für Behälter aus Faserverbundwerkstoffen**

**Markus Steffens, Ansgar Bernardi**

DFKI-D-95-06

Diese Arbeit wurde finanziell unterstützt durch das Bundesministerium für Forschung und Technologie (FKZ ITW-9304/3).

© Deutsches Forschungszentrum für Künstliche Intelligenz 1995

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.

ISSN 0946-0098

## **Kurzfassung**

Dieses Dokument definiert ein Produktmodell für Druckbehälter aus Faserverbundwerkstoffen. Die hier aufgeführten Formalismen erlauben die symbolische Repräsentation der produktbezogenen Informationen wie etwa geometrische Gestalt, Topologie, Technologie und Materialaufbau.

## **Abstract**

This document defines a product model for pressure tanks made of composites. We present a set of representation formalisms which allow the symbolic representation of product-related information, e.g. geometrical shape, topology, technology, structure of material layers.

## Inhalt

Kurzfassung .....	i
Abstract .....	i
Inhalt .....	ii
1 Einleitung .....	1
2 Ausführung verschiedener Behälter-Bauteile .....	4
2.1 Vereinfachungen .....	5
3 Systematische Einteilung des Bauteiles 'Behälter' .....	6
4 Explizite Beschreibung der Geometrie durch Primitive (Geometriemodell) .....	7
4.1 Syntax .....	7
4.2 Allgemeine Produktinformationen .....	9
4.3 Globale Produktinformationen .....	9
4.4 Globales Koordinatensystem .....	10
4.5 Lokales Koordinatensystem .....	10
4.6 Geometrie- und Technologiemakros .....	12
4.7 Flächenelemente für rotationssymmetrische Teile .....	13
4.7.1 Zylindermantelfläche .....	13
4.7.2 Kreisfläche .....	14
4.7.3 Kegelmantelfläche .....	15
4.7.4 Torusabschnittfläche .....	16
4.7.5 Kugelkalotte .....	18
4.7.6 Kreisringfläche .....	19
4.7.7 Verrundung .....	20
4.8 Öffnungen (Bohrungen) .....	21
4.9 Gewinde .....	23
4.10 Wandstärke .....	25
5 Topologiemodell .....	27
6 Produktstruktur- und Materialmodell .....	30

---

7	Oberflächenmodell .....	33
7.1	Oberflächengüte (Surface Finish) .....	33
7.2	Härte (Hardness) .....	33
8	Toleranzmodell .....	34
8.1	Maßtoleranz (Size Standard) .....	35
8.2	Allgemeintoleranz DIN 7168 (General Tolerance) .....	36
8.3	Formtoleranz DIN 7184 (Shape Tolerance) .....	37
8.4	Lagetoleranzen (Positional Tolerance) .....	39
9	Literaturverzeichnis.....	42
10	Beispiel-Geometrie- und Topologiebeschreibung anhand eines erdachten Behälters .....	43





## 1 Einleitung

Entscheidende Voraussetzung für die elektronische Unterstützung eines Produktlebenszyklus ist die geeignete Repräsentation der produktbezogenen Daten. Die hierzu notwendigen Formalismen werden vom sogenannten Produktmodell zur Verfügung gestellt. Die Vielfalt der während des Produktlebenszyklus anfallenden Aufgaben erfordert u. U. eine Beschreibung unter verschiedenen Gesichtspunkten, mithin die Repräsentation unterschiedlicher Sichtweisen ein und desselben Produkts. Die gemeinsame Behandlung der verschiedenen Beschreibungsmöglichkeiten führt zur Definition sogenannter integrierter Produktmodelle. Aufgrund ihrer zentralen Rolle im CIM-Bereich sind integrierte Produktmodelle Gegenstand vielfältiger Forschungsanstrengungen und Objekt eines internationalen Standardisierungsprozesses. Insgesamt beschreibt das Produktmodell die Struktur eines Repositoriums für alle während des Produktlebenszyklus anfallenden produktbezogenen Daten. Das einzelne Produkt wird durch Instanzen der im Produktmodell bereitgestellten Beschreibungsobjekte in für die anfallenden Aufgaben ausreichender Genauigkeit dargestellt.

Das vorliegende Dokument definiert ein Produktmodell für Druckbehälter aus Faserverbundwerkstoffen. Der hier festgelegte Formalismus bildet die Basis für die Arbeiten des Forschungsprojekts IMCOD, soweit diese auf konkreten produktbezogenen Daten aufsetzen. Daher stellt das Produktmodell im wesentlichen Objekte zur Repräsentation von Gestalt, Materialaufbau und technologischen Informationen von Druckbehältern aus Faserverbundwerkstoffen zur Verfügung.

Das vorliegende Produktmodell basiert auf dem im ARC-TEC Projekt entwickelten Repräsentationsformalismus TEC-REP [1]. Die bei der Gestaltung von TEC-REP verwendeten Prinzipien wurden im wesentlichen übernommen: Grundlage ist die Beschreibung des Produktes durch eine Folge von eindeutig nummerierten symbolischen Entities. Die Darstellung der Werkstückgeometrie erfolgt durch die Kombination von primitiven Oberflächenelementen (also Boundary Representation - BREP), die Werkstücktopologie wird durch Nachbarschaftsrelationen zwischen diesen Elementen explizit modelliert. Technologische Angaben werden als oberflächenbezogene Attribute verwaltet. Die syntaktische Darstellung der verwendeten Entities ist einerseits auf eine einfache Anwendung ausgerichtet, deshalb die LISP-Ähnlichkeit der gewählten Darstellung. Andererseits soll das Erzeugen einfacher ASCII-Files mit

struktureller STEP<sup>1</sup>-Verträglichkeit möglich sein, deshalb werden geschachtelte Strukturen, Listen u. ä. weitgehend vermieden und Referenzen durch die eindeutige Numerierung der Entities aufgelöst.

TEC-REP wurde im Hinblick auf rotationssymmetrische Teile optimiert. Dies führte zur Verwendung eines globalen kartesischen Koordinatensystems, dessen x-Achse die Rotationsachse des Werkstücks repräsentierte. Ferner wurden die geometrischen Primitive in einer hierauf abgestimmten vereinfachten Form zur Verfügung gestellt: Zur Ortsbeschreibung genügten einfache Koordinatenangaben. Für die Beschreibung nicht rotationssymmetrischer Teile wurden die gleichen Entities noch in einer zweiten Form bereitgestellt, hier traten an Stelle der Koordinaten vollständige Ortsvektoren.

Für die effiziente Darstellung von Druckbehältern aus Faserverbundwerkstoffen wurde diese Unterscheidung nicht mehr beibehalten. Die Rotationssymmetrie ist in diesem Zusammenhang nicht von zentraler Bedeutung, Anschlüsse und Öffnungen stören in der Regel die Symmetrie des Behälters. Aus der Art der Behälter folgt ferner die relativ häufige Verwendung von Zylinder- und Kugelkoordinatensystemen.

Als Konsequenz werden in diesem Produktmodell den einzelnen geometrischen Primitiven jeweils eigene Koordinatensysteme zugeordnet. Die Gestalt der Ortsvektoren richtet sich dann nach dem Typ des jeweiligen Koordinatensystems. Die Positionierung eines einzelnen geometrischen Elements im Gesamtverbund wird durch die entsprechende Lagebestimmung seines lokalen Koordinatensystems erreicht. Dieser Ansatz ermöglicht nicht nur die einfache Beschreibung beliebig orientierter Elemente, sondern unterstützt auch die Gruppierung und relative Verschiebung komplexer Baugruppen.

In TEC-REP war die Verwaltung einer expliziten Materialrichtung sinnvoll. Die geometrischen Primitive bekamen damit auch die Bedeutung von unvollständig beschriebenen Volumenelementen.

Im hier vorliegenden Modell wird dieser Ansatz nicht mehr weiter verfolgt. Die geometrischen Beschreibungen haben nun keine Materialbedeutung mehr, sondern beschreiben eine reine Geometrie. Implizit ist allerdings angenom-

---

<sup>1</sup> **STEP** (STandard for the Exchange of Product model data) ist der projektierte ISO-Standard für den Datenaustausch in der CIM-Welt. Die Verwendung dieses Standards sollte u.a. die Anbindung existierender CAD-Systeme ermöglichen. Für eine Beschreibung von STEP vgl. /2/, /3/

men, daß ein eventueller Materialauftrag immer auf der Außenseite der Kontur erfolgt (durch die lokalen Koordinatensysteme ist 'außen' wohldefiniert).

Die Dicke des Materials wird durch eigene Primitive beschrieben. Hier liegt die Idee zugrunde, Meßpunkte anzugeben, an denen ein Maß festgelegt wird; zwischen den Punkten ist eine Interpolation möglich.

Der Materialaufbau eines Behälters aus Faserverbundwerkstoffen ist relativ komplex und von zentraler Bedeutung. Dem trägt eine Reihe von Primitiven Rechnung, die die Modellierung beliebiger Materialien und Schichtenfolgen erlauben: Teile des Behälters können unterschiedlich aufgebaut sein, das Produkt-Struktur-Element PSE erlaubt die Referenzierung entsprechender Teilgeometrien. Jede einzelne Lage wird explizit beschrieben und ermöglicht die Angabe der Materialkenngrößen. Die Lagenfolge wird (vergleichbar der Topologie) durch Ordnungselemente bestimmt.

Wichtige Grundgrößen, die den ganzen Behälter betreffen (wie etwa minimaler und maximaler Druck), sind in Form von Attribut-Wert-Paaren dem Behälter als ganzem zugeordnet (Vessel).

Da das IMCOD-Projekt die Unterstützung des Produktentwurfs zum Ziel hat, ist es notwendig, im Produktmodell die notwendigen Mittel zur Darstellung von erst partiell bestimmten Produkten bereitzuhalten. Zu diesem Zweck kann jeder Wert durch eine freie Variable ersetzt werden. Der Gültigkeitsbereich einer Variablen umfaßt das gesamte Produktmodell. Die weitere Behandlung der Variablen, insbesondere die Notation zusätzlicher Constraints über solche Variablen, geht über das vorliegende Dokument hinaus und ist Gegenstand eigener Untersuchungen.

## 2 Ausführung verschiedener Behälter-Bauteile

Ein Behälter besteht im allgemeinen aus einer Behälterschale, aufgegliedert in Mantel und Boden, diversen Öffnungen, Anschlüssen (Stutzen) und verschiedenen Anbauteilen, wie Befestigungsflansche oder Standbeine. Im Laufe der Zeit haben sich im Apparatebau verschiedene Standards herausgebildet, die zu bestimmten bevorzugten Gestaltungsarten o.g. Bauteile geführt haben. Diese sind im folgenden in Tabelle 1 aufgelistet.

Behälter-Bauteil	Ausführung
Mantelarten	<ul style="list-style-type: none"> <li>- zylindrisch</li> <li>- kugelig</li> <li>- ballig (tonnenförmig)</li> <li>- rechteckig</li> <li>- elliptisch (oval)</li> </ul>
Bodenarten (Krempe + Kalotte)	<ul style="list-style-type: none"> <li>- ebener Boden</li> <li>- Tellerboden</li> <li>- Flachboden (mit Krempe)</li> <li>- flachgewölbter Boden</li> <li>- normal gewölbter Boden</li> <li>- Klöpperboden</li> <li>- Korbbogenboden</li> <li>- Ellipsenboden</li> <li>- Halbkugelboden</li> </ul>
Stutzenarten (ausschl. zylindrisch)	<ul style="list-style-type: none"> <li>- durchgesteckt</li> <li>- eingesteckt</li> <li>- aufgesetzt</li> <li>- aufgesetzt mit Aushalsung</li> </ul>
Befestigungsarten	<ul style="list-style-type: none"> <li>- Tragpratzen</li> <li>- Standbeine</li> </ul>

Tabelle 1. Versch. Ausführungen von Behälter-Komponenten

Ein explizite Darstellung aller Behälterböden, die von einem Kugel-, Flach- oder ebenen Boden abweichen, ist aufgrund deren komplexer Geometrie unumgänglich.

Bild 1 zeigt beispielhaft einen Faserverbundwerkstoff(FVW)-Druckbehälter mit den wichtigsten Komponenten und dem internen Lagenaufbau.

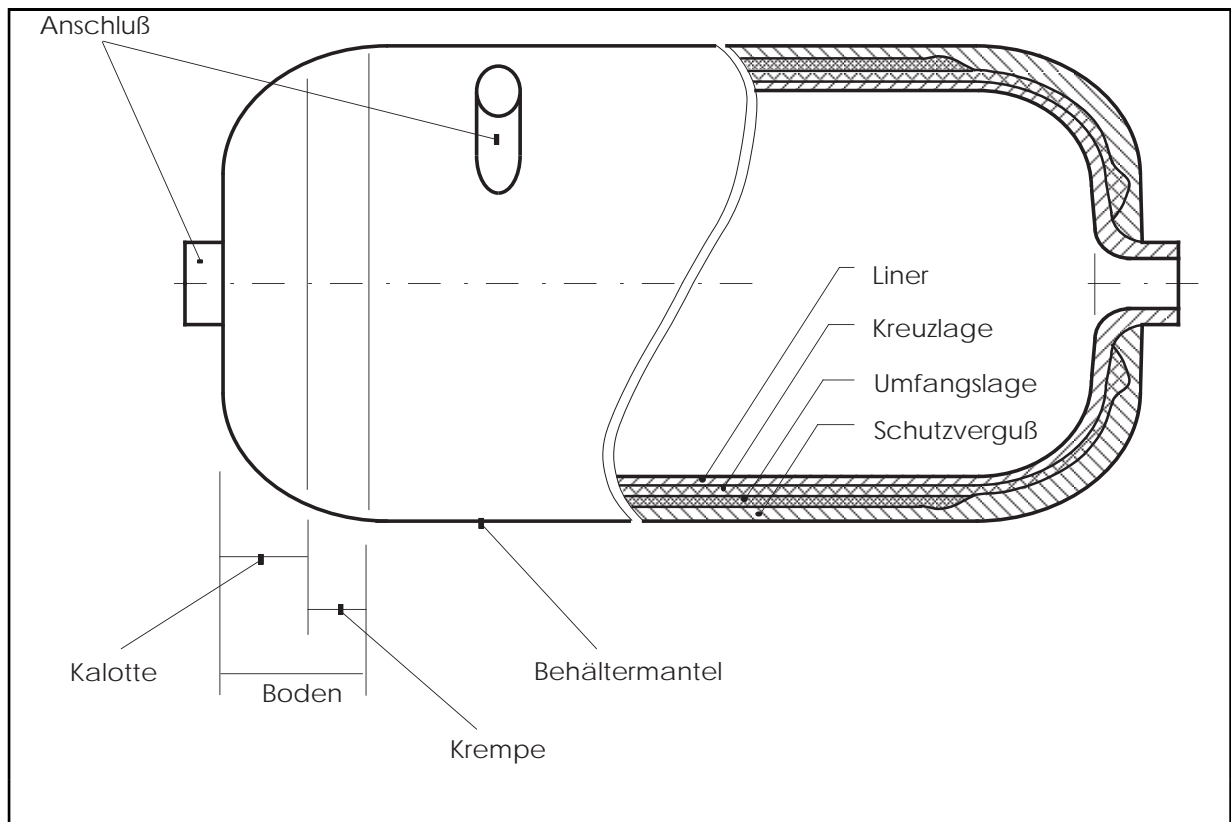


Bild 1. Beispiel für den Aufbau eines Druckbehälter

## 2.1 Vereinfachungen

Zur Reduzierung der Variantenvielzahl sind folgende Einschränkungen bei der Repräsentation eines solchen Behälters sinnvoll:

- Wir behandeln nur Behälter mit rotationssymmetrischem Behältermantel.
- Aushalsungen für Stutzenverbindungen und Krempe müssen als Torus-Teilflächen modellierbar sein.

### 3 Systematische Einteilung des Bauteiles 'Behälter'

		BEHÄLTER, bestehend aus													
		Liner			Wicklung			Deckschicht			Anschluß				Befestigung
		Mantel	Boden		Mantel	Lagen		Mantel	Lagen		Aushalung	Mantel	Flansch	Gewinde	
			Krempe	Kalotte		Krempe	Kalotte		Krempe	Kalotte					
BEHÄLTER-BAUTEIL, beschrieben unter dem Gesichtspunkt	Geometrie	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13	1,14
	Topologie	2,1	...	...	...										
	Material	3,1	...												
	Oberfläche	4,1	...												
	Toleranz	5,1													

Tabelle 2. Systematische Einteilung des Bauteiles 'Behälter'

Tabelle 2 zeigt die systematische Einteilung des zu repräsentierenden Modells 'Behälter', betrachtet unter den einzelnen Gesichtspunkten der Partialmodelle Geometrie, Topologie, Material, Oberfläche und Toleranz.

Im folgenden sollen die Inhalte der Tabelle 2 ( Pos. [1,1] bis [5,14] ) genauer betrachtet werden.

## 4 Explizite Beschreibung der Geometrie durch Primitive (Geometriemodell)

Die Geometrie eines Behälters läßt sich i.d.R. aus sog. Primitiven zusammensetzen. Diese Primitive sind vordefinierte Geometrie-Grundelemente (hier Flächenelemente), die ein eigenes lokales Koordinatensystem besitzen können (s. nächster Punkt 'Koordinatensystem') und über dessen Ursprung in Bezug auf das globale Koordinatensystem des Behälters positioniert werden. Durch Verschneiden aneinandergrenzender Primitive (BOOLEsche Verknüpfungen) werden anschließend - im Falle von Flächen - Schnittkurven gebildet, welche die Berandungslinien zweier benachbarter Elemente darstellen.

Die hier zur Definition eines Behälters benötigten Primitive, unter Berücksichtigung der im Punkt 'Einschränkungen' erläuterten Annahmen, sind die Zylindermantelfläche, Konusmantelfläche, Kugelfläche und die Torusfläche sowie von Fall zu Fall Abschnitte (Teile) dieser Grundelemente. Im folgenden sind die für eine eindeutige Definition dieser Elemente nötigen vektoriellen und skalaren Größen, wie Referenzpunkte (sog. Aufpunkte), Richtungsvektoren und Radien erläutert.

Die Definitionen beziehen sich auf das dargestellte lokale Koordinatensystem des Primitivs.

### 4.1 Syntax

Die einzelnen Entities<sup>2</sup> werden durch Angabe ihres Namens Kürzels und der dazugehörigen Attribute beschrieben. Jedes Beschreibungselement erhält außerdem eine eindeutige Nummer (Identifikation), über die es referenziert werden kann. Die Reihenfolge der Nummern selbst ist ohne Bedeutung, jedoch müssen bei gleichzeitig mehreren beschriebenen Bauteilen die Menge der zu jedem Bauteil gehörenden Entity-Nummern disjunkt sein.

---

<sup>2</sup> **Entity** (engl.) ist gleichbedeutend mit dem Begriff 'Beschreibungselement'

In Anlehnung an die STEP-Konventionen wird in diesem Papier folgende Syntax verwendet:

*@Nummer := Namenskürzel ( Attributname: Attributwert, ...)*

Dabei treten als Attributwertebereiche folgende Bezeichnungen auf:

<string>	Zeichenreihe
<char>	Buchstabe
<num>	eine (reelle) Zahl
<vector>	Vektor: kartesisch(x,y,z); zylindrisch(z,R,φ)

Wird in den Attributen eine Nummer referenziert, so wird dies durch ein vorangestelltes Kreuz '#' kenntlich gemacht. Vgl. hierzu das Beispiel im Anhang.

In den Definitionen treten folgende Metazeichen auf:

{...}	Der zwischen geschweifte Klammern eingeschlossene Teil ist optional und kann entfallen.
*	bezeichnet beliebige Wiederholungen (mindestens 0).

(<num> {,<num>}\*) beschreibt also eine in Klammern eingeschlossene Liste beliebig vieler durch Kommata getrennter Zahlen, wobei mindestens eine Zahl vorhanden sein muß.

Bemerkungen (Remarks) werden, der STEP-Konvention folgend, zwischen die Zeichenfolge '(\*' und '\* )' gefügt.



## 4.2 Allgemeine Produktinformationen

Zur Repräsentation eines gegebenen Produktes, hier eines Behälters aus Faserverbundwerkstoff, sind neben den konkreten Geometrie- und Technologiedaten auch einige globale Informationen nötig. Dazu gehören Angaben zum Behälter als Ganzes (wie Bezeichnung oder Gesamtlänge) und Verwaltungsinformationen des Repräsentationsformalismus (wie Maßeinheit und Nummerierung). Weiterhin ist es oftmals notwendig, Informationen über die Produktstruktur des beschriebenen Behälters zu repräsentieren.

## 4.3 Globale Produktinformationen

Dieses Beschreibungselement faßt die zu einem Behälter gehörenden, nicht aus den folgenden Produktbeschreibungen ersichtlichen, Informationen zu dessen erster Charakterisierung zusammen und spezifiziert die für den gesamten Behälter gültigen Angaben, wie Maßeinheit, Bezeichnung, Gewicht, Volumen, Druck, Inhalt, Material und allgemeine Informationen. Hier kann das gesamte Schriftfeld einer technischen Zeichnung abgebildet werden.

### Behälter (Vessel):

```
V(   name: <string>,
      application: <string>,
      material: <string>,
      Pmin: <num>,
      Pmax: <num>,
      info: <string>,
      entities: ( <num>* ) )
```

### Attribute:

name:	Bezeichnung des Behälters
application:	Anwendung
material:	Materialbezeichnung (speziell des Faserverbundwerkstoffes FVW)
Pmin:	Minimal zulässiger Druck (= maximaler Unterdruck)
Pmax:	Maximal zulässiger Druck
info:	Beliebige weitere Informationen für den gesamten Behälter

entities: Liste der Nummern aller Beschreibungsentities für diesen Behälter

Konsistenzbedingungen: keine

#### 4.4 Globales Koordinatensystem

**Globales Koordinatensystem (Global Coordinate System):**

**GCS ( type: [KS; CS; SS],  
unit: <string> )**

Attribute:

type: Typ des globalen Koordinatensystems  
KS  $\Leftrightarrow$  kartesisches System  
CS  $\Leftrightarrow$  zylindrisches System  
SS  $\Leftrightarrow$  Kugel-System

unit: Die Einheit des Koordinatensystems (m, cm ,mm ...)  
Winkelangaben sind immer im Bogenmaß (rad)!

Konsistenzbedingungen: keine

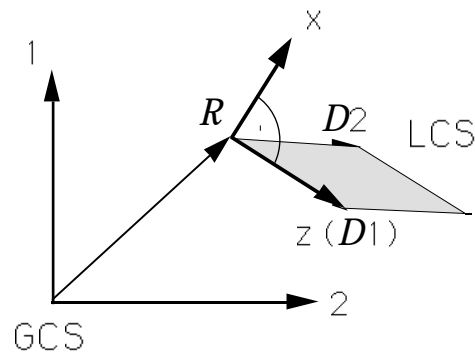
Bei Verwendung eines zylindrischen oder Kugel-Koordinatensystemes sind die notwendigen Winkelgrößen immer in Bogenmaß [rad] angegeben. Demnach beziehen sich Einheiten-Angaben bei der Koordinatensystemdefinition ausschließlich auf die Längenangaben.

Aufgrund der getroffenen Annahme, ausschließlich rotationssymmetrische Behälter zuzulassen, bietet sich zumeist die Verwendung eines Zylinder-Koordinatensystemes (CS) an.

#### 4.5 Lokales Koordinatensystem

Die Definition lokaler Koordinatensysteme, die selbst wiederum Bezugssystem sein können, erscheint sinnvoll, um einerseits eine wie vor dargestellte Repräsentation einzelner Elemente, unabhängig von deren Position im Raum, zu ermöglichen und weiter die Möglichkeit der Baugruppenbildung zu erschließen, indem mehrere Elemente auf ein Bezugskoordinatensystem referenziert und durch Verschieben dieses Systemes im gesamten positioniert werden können.

## Lokales Koordinatensystem (Local Coordinate System)



LCS ( **coordinate\_system**: <num>,  
**origin**: <vector>,  
**direction\_vector1**: <vector>  
**direction\_vector2**: <vector>,  
**type**: [KS; CS; SS],  
**unit**: <string> )

Attribute:

<b>coordinate_system</b> :	Bezugskoordinatensystem
<b>origin</b> :	Koordinatenursprung R des lokalen Systems
<b>direction_vector1</b> :	Richtungsvektor D1 der z-Achse
<b>direction_vector2</b> :	Richtungsvektor D2 zur Definition der $\varphi = 0$ -Ebene zusammen mit der z-Achse
<b>type</b> :	Typ des lokalen Koordinatensystems KS $\Leftrightarrow$ kartesisches System CS $\Leftrightarrow$ zylindrisches System SS $\Leftrightarrow$ Kugel-System
<b>unit</b> :	Einheit des Systems

Definition: Im kartesischen System gilt:  
x-Achse  $\perp$  D1, D2; y-Achse  $\perp$  x-Achse, z-Achse

## 4.6 Geometrie- und Technologiemakros

In diesem Beschreibungselement sollen zusammenfassend einige wenige, für die Grobbeschreibung des Behälters wichtige Daten repräsentiert werden können, die aus anderen Angaben der Produktbeschreibung zwar ersichtlich (explizit oder implizit), jedoch maßgeblich für die Vorstellung und Einordnung des späteren Produktes sind.

Dies sind z.B. die Gesamtlänge des Behälters, der kleinste und größte Durchmesser, die höchste Toleranz der Abmaße, die beste Oberflächenangabe, usw. Sämtliche Angaben sind optional!

Geometrie- und Technologiemakro (**G**eometry and **T**echnology **M**acros):

```
GTM( {Lmax: <num>},  
      {Dmax: <num>},  
      {Dmin: <num>},  
      {hmin: <num>},  
      {hmax: <num>},  
      {Fmin: <num>},  
      {ITmin: <num>} )
```

Attribute:

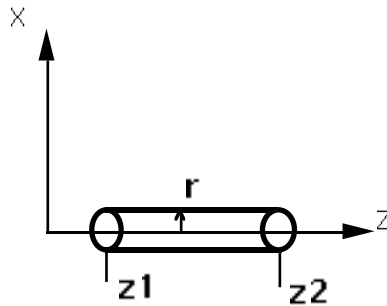
Lmax:	die maximale Länge des Behälters
Dmax:	der maximale Durchmesser des Behälters
Dmin:	der minimale Durchmesser des Behälters
hmin:	die minimale Dicke der Behälter-Wandstärke
hmax:	die maximale Dicke der Behälter-Wandstärke
Fmin:	die minimale (beste) Oberflächengüte des Behälters
ITmin:	die minimale (höchste) Toleranz der Abmaße des Behälters

Konsistenzbedingungen: keine

## 4.7 Flächenelemente für rotationssymmetrische Teile

### 4.7.1 Zylindermantelfläche

#### Zylindermantel (Cylinder Jacket)



**CJ ( coordinate\_system: <num>,  
reference\_point1: <num>,  
reference\_point2: <num>,  
radius: <num> )**

Attribute:

coordinate_system:	Referenznummer des Koordinatensystems
reference_point1:	Referenzpunkt z1
reference_point2:	Referenzpunkt z2
radius:	Radius r

Implizit:

$L = | z_2 - z_1 |$       Länge des Zylinders

Definition: Richtung des Zylinders ist die Richtung der z-Achse

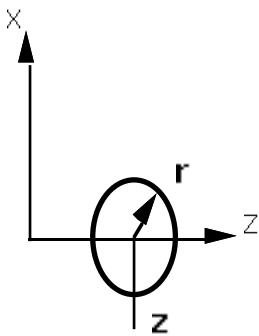
Konsistenzbedingungen:

$z_1 \neq z_2$

$r > 0$

### 4.7.2 Kreisfläche

#### Kreisfläche (CIRcular surface)



CIR ( coordinate\_system: <num>,  
reference\_point: <num>,  
radius: <num> )

Attribute:

coordinate\_system: Referenznummer des Koordinatensystems

reference\_point: Referenzpunkt z: Lage des Mittelpunktes

radius: Radius r

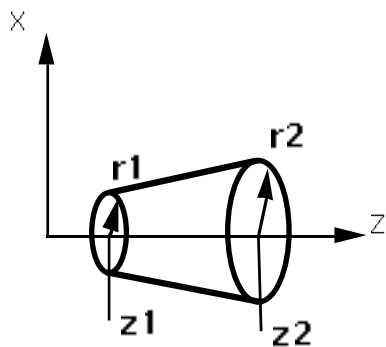
Definition: Richtung der Flächennormalen ist die Richtung der z-Achse

Konsistenzbedingungen:

$r > 0$

### 4.7.3 Kegelmantelfläche

#### Kegelmantelfläche (Envelope of Cone)



EC( coordinate\_system: <num>,  
 reference\_point1: <num>,  
 reference\_point2: <num>,  
 radius1: <num>,  
 radius2: <num> )

Attribute:

coordinate_system:	Referenznummer des Koordinatensystems
reference_point1:	Referenzpunkt z1
reference_point2:	Referenzpunkt z2
radius1:	Radius r1
radius2:	Radius r2

Implizit:

$$L = | z1 - z2 | \quad \text{Länge des Kegels}$$

Definition: Richtung der Kegelmantelfläche ist die Richtung der z-Achse

Konsistenzbedingungen:

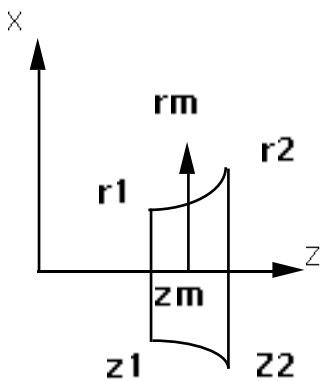
$$z1 \neq z2$$

$$r1 \geq 0, r2 \geq 0$$

#### 4.7.4 Torusabschnittfläche

Eine Torusfläche ist diejenige Fläche, die eine Kreislinie (die Erzeugende) beschreibt, wenn deren Mittelpunkt längs einer anderen Kreislinie (der Direktionslinie) bewegt wird. Die Torusabschnittfläche ist ein Ausschnitt einer Torusfläche.

#### Torusabschnittfläche (Toroidal Shell)



```

TS (   coordinate_system: <num>,
      reference_point1: <num>,
      reference_point2: <num>,
      reference_point3: <num>,
      radius1: <num>,
      radius2: <num>,
      radius3: <num>
      direction_of_arch: [v,x] )

```

Attribute:

coordinate_system:	Referenznummer des Koordinatensystems
reference_point1:	Referenzpunkt z1
reference_point2:	Referenzpunkt z2
reference_point3:	Referenzpunkt zm: Mittelpunkt der Direktionslinie des Torus
radius1:	Radius r1
radius2:	Radius r2
radius3:	Radius der Direktionslinie des Torus rm



---

direction\_of\_arch: Beschreibung der Wölbungsrichtung  
x  $\Leftrightarrow$  die Wölbung ist konvex.  
v  $\Leftrightarrow$  die Wölbung ist konkav

Definition: Die von  $r_1$  und  $r_2$  aufgespannten Kreisflächen stehen senkrecht auf der z-Achse;

$$r_{\text{Erz}} = \sqrt{(r_1 - r_m)^2 + |z_1 - z_m|^2} \quad \text{Radius der Erzeugenden}$$

Konsistenzbedingungen:

$$z_1 \neq z_2 \neq z_m;$$

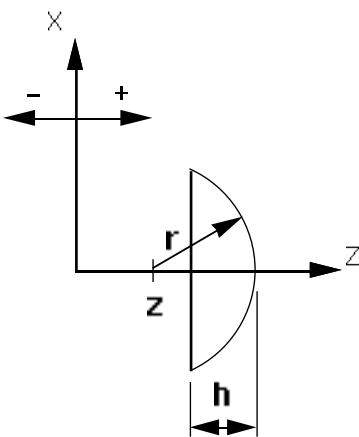
$r_1 \geq 0, r_2 \geq 0, r_m \geq 0$ , wobei entweder  $r_1 > 0$  oder  $r_2 > 0$ ;

$$(r_2 - r_m)^2 + |z_2 - z_m|^2 = (r_1 - r_m)^2 + |z_1 - z_m|^2$$

### 4.7.5 Kugelkalotte

Eine Kugelkalotte ist eine Abschnittsfläche einer Kugel. Sie dient oftmals als Teil des Behälterbodens und ist über den Ursprungsradius der Kugel und die Kalottenhöhe definiert.

#### Kugelkalotte (Ball Joint)



```
BJ (  coordinate_system: <num>,
      center: <vector>,
      radius: <num>,
      height: <num>,
      joint_direction: [+; -] )
```

Attribute:

coordinate_system:	Referenznummer des Koordinatensystems
center:	Kugelmittelpunkt z
radius:	Radius r der Kugel
height:	Kalottenhöhe h
joint_direction:	Kalotten-Richtung
	+    ⇔    in z-Richtung
	-    ⇔    entgegen z-Richtung

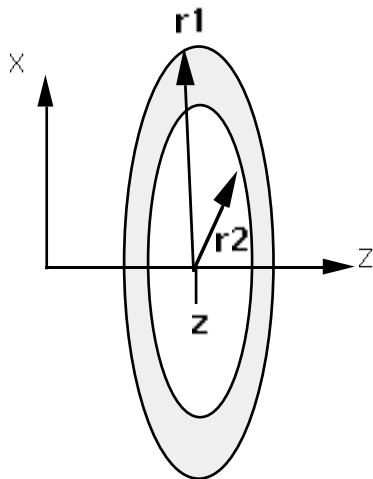
Konsistenzbedingungen:

$r, h > 0$

#### 4.7.6 Kreisringfläche

Eine in der Praxis häufig vorkommende gelochte Fläche ist die Kreisringfläche. Deshalb wird sie extra repräsentiert, um den Beschreibungsaufwand zu minimieren.

##### Kreisringfläche (RIng)



```
RI ( coordinate_system: <num>,
      reference_point: <num>,
      radius1: <num>
      radius2: <num> )
```

Attribute:

coordinate_system:	Referenznummer des Koordinatensystems
reference_point:	Referenzpunkt z
radius1:	Radius r1
radius2:	Radius r2

Definition: Die Flächennormale wird durch die Richtung der z-Achse definiert

Konsistenzbedingungen:

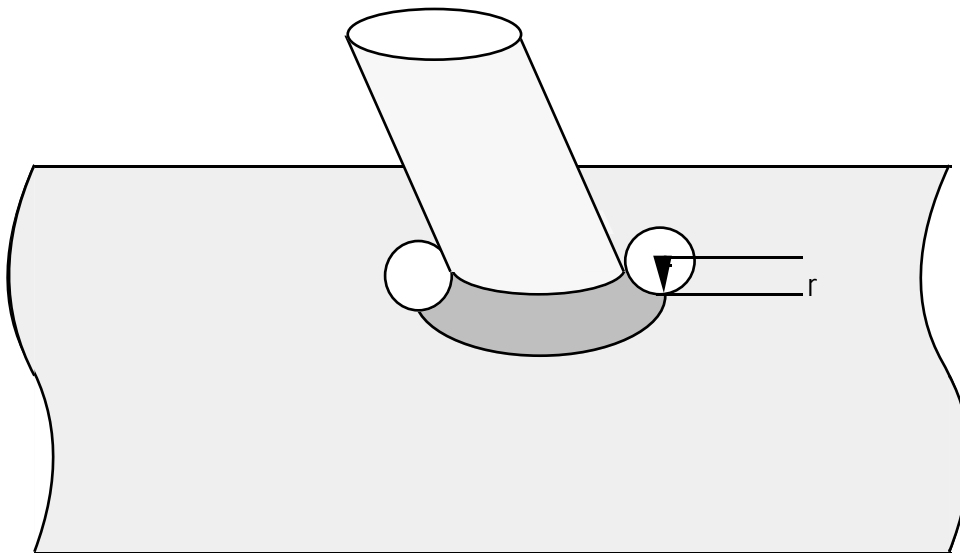
$$r1 > r2 > 0$$

### 4.7.7 Verrundung

Vielfach sind im Behälterbau Anbindungen zu Rohrleitungen oder Meßinstrumenten zu schaffen, wozu oftmals sog. Anschlußstutzen am Behälter angebracht werden. Um Spannungsüberhöhungen im Fußbereich dieser Stutzen zu vermeiden, sind sie dort meist mit einem Radius verrundet.

Die Entstehung einer Verrundung im Fußbereich eines Anschlußstutzens kann mit dem Abrollen einer Kugel mit einem Radius entsprechend des geforderten Rundungsradius gleichgesetzt werden. Die Berührungsbahn der Kugel auf Stutzen und Behältermantel stellt demnach den entsprechenden Auslauf der Verrundung dar.

Die folgende Abbildung zeigt die resultierende Verrundungszone im Fußbereich eines Stutzens, der nicht radial in einen Behältermantel mündet.



#### Verrundung (CURvature)

```
CUR ( element_reference_no1 : <num> ,  
      element_reference_no2 : <num> ,  
      radius : <num> )
```

Attribute:

element\_reference\_no1: Referenzelement 1

element\_reference\_no2: Referenzelement 2

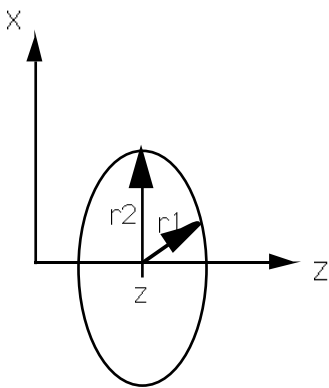
radius: Radius r der Verrundung

#### 4.8 Öffnungen (Bohrungen)

Es wurden sowohl runde als auch ovale Behälteröffnungen vorgesehen, da diese Öffnungen typisch für das Anbinden von Stützen oder die Anbringung von Deckeln ist. Ovale oder allgemein nicht runde Öffnungen eignen sich insbesondere für die Montage von innenliegenden Deckeln, vorzugsweise druckbeaufschlagter Behälter, da es so möglich ist, den Deckel zunächst in das Behälterinnere einzuführen und anschließend nach dessen Verdrehung gegen einen Anschlag zu fixieren.

Zur Vereinfachung der Öffnungsbeschreibung ist die Bezeichnung rund und oval jeweils in Bezug auf eine Ansicht in Richtung der Öffnungsachse  $z$  zu sehen. Entsteht demnach beispielsweise eine Öffnung aufgrund des gedachten (imaginären) Eindringens eines kleinen Zylinders in einen größeren, wie dies CAD-technisch zur Erzeugung von komplexen Öffnungen üblich ist, so ist die Projektion der Eindringstelle ausschließlich in Richtung des eindringenden Zylinders - also dessen  $z$ -Achse - kreisförmig. Demnach wird die Öffnung als Kreis repräsentiert, obwohl sie in der Abwicklung oder bei Betrachtung aus einer anderen Richtung nicht kreisförmig ist. Dasselbe gilt für ovale Rohre.

## Öffnung (HOle)



HO ( coordinate\_system: <num>,  
 element\_reference\_no: <num>,  
 center: <num>,  
 radius\_min: <num>,  
 radius\_max: <num> )

### Attribute:

coordinate_system:	Bezugsnummer des Koordinatensystems
element_reference_no:	Nummer des Bezugselementes
center:	Mittelpunkt z der Öffnung
radius_min:	Radius r1 der Öffnung senkrecht auf z und x
radius_max:	Max. Radius r2 der Öffnung, senkrecht auf r1

### Implizit:

$r1 = r2 \Rightarrow$  Öffnung ist rund

### Definition:

$r1 \perp x, z; r1 \perp r2$

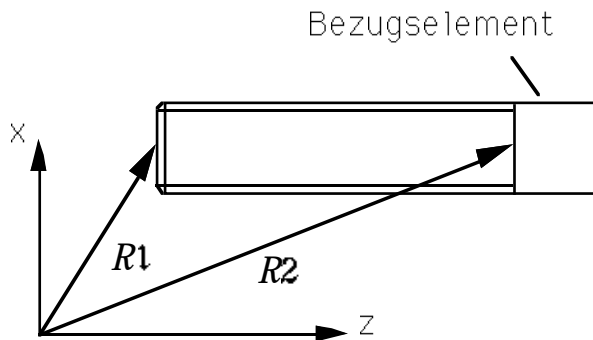
### Konsistenzbedingungen:

$r1, r2 > 0$

## 4.9 Gewinde

Betrachtet werden sollen ausschließlich zylindrische Gewinde. Konische Gewinde sind für den Behälterbau bedeutungslos. Gewindeanfang und -auslauf können durch zwei separate Attribute näher spezifiziert werden.

### Gewinde (Thread)



```

TH (  element_reference_no: <num>,
      reference_point1: <vector>,
      reference_point2: <vector>,
      location: [ +; - ],
      type: [ M; I ],
      size : <num>
      { thread_begin: <string> },
      { thread_end: <string> } )

```

Attribute:

element_reference_no:	Element-Bezugsnummer
reference_point1:	Anfangspunkt des Gewindes $R1$
reference_point2:	Endpunkt des Gewindes $R2$
location:	Gewindeposition + $\Leftrightarrow$ Außengewinde - $\Leftrightarrow$ Innengewinde
type:	Gewindeart M $\Leftrightarrow$ Metrisches Gewinde I $\Leftrightarrow$ Zoll-Gewinde
size:	Gewinde-Durchmesser d

thread\_begin: Gewinde-Anfang  
thread\_end: Gewinde-Auslauf

Implizit: Länge des Gewindes  $L = |\mathbf{R}_2 - \mathbf{R}_1|$

Definition: Vektorielle Größen beziehen sich auf das Koordinatensystem des referenzierten Elementes; nur zylindrische Elemente sind als element\_reference\_no zugelassen

Konsistenzbedingungen:  $L, d > 0$



#### 4.10 Wandstärke

Es bieten sich mehrere Repräsentationsmöglichkeiten der Wandstärke der Behälterschale an. Dabei kann die Behälterschale beispielsweise aus zwei im Abstand der Wandstärke zueinander positionierten Grundgeometrien gebildet werden. Ein optionaler Liner, der für sich gesehen ebenso einen Behälter darstellt, kann ebenso in der vorgenannten Form repräsentiert werden. Bei der einzelnen Repräsentation jeder Einzellage eines Behälters führt dies zu einer Explosion der Geometriedaten. Daher erscheint die Darstellung der Behälterschale durch jeweils eine Grundgeometrie mit punktuell zugeordneten Wandstärke-Bezugspunkten sinnvoll. Diese Punkte beziehen sich dann jeweils auf ein Geometrieelement und geben die Gesamtwandstärke in diesem Punkt an. Sollen anstatt Gesamtwandstärken hingegen Wandstärken einzelner Schichten (Schichtdicken) angegeben werden, so wird anstelle eines Geometrieelementes ein Layer-Design-Element (LDE, vgl. Kap. 6: Produktstruktur- und Materialmodell) referenziert.

Eine Überprüfung der Wandstärke bei der Qualitätssicherung ist meßtechnisch ebenfalls nur punktuell möglich. Somit können die referenzierten Wandstärken gezielt angefahren und überprüft werden. Bei Übergängen von einer Wandstärke zu einer anderen soll nur zwischen direkt benachbarten Dickenangaben interpoliert werden, nicht jedoch über die Gesamtmenge aller Dicken-Referenzpunkte.

Es soll vorausgesetzt werden, daß alle Geometrie Größen *innere* Abmaße sind und damit *Dickenangaben grundsätzlich nach außen gerichtet sind* (das ist für die hier betrachteten rotationssymmetrische Teile eindeutig; vgl. Kap. 2.1: Vereinfachungen).

##### Wandstärke-Bezugspunkt (Thickness Reference Point)

```
TRP (  coordinate_system: <num>,
       element_reference-no: <num>,
       reference_point: <vector>,
       thickness: <num> )
```

Attribute:

coordinate_system:	Bezugsnummer des Koordinatensystems
element_reference_no:	Element-Bezugsnummer

reference\_point:            Bezugspunkt R der Dickenangabe  
thickness:                 Dicke s im referenzierten Punkt

Definition: Die Dicke ist immer in Richtung der Flächennormalen gerichtet.

Konsistenzbedingungen:  $s > 0$

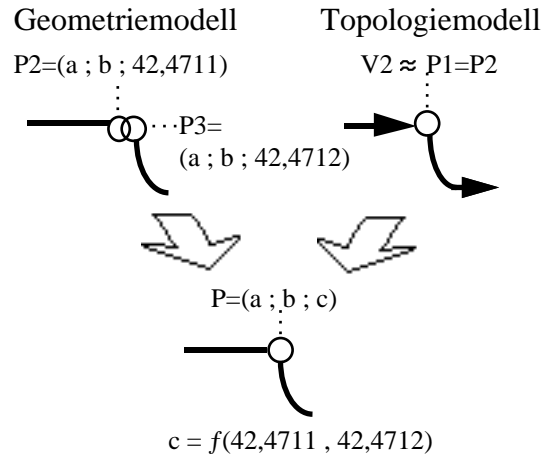
## 5 Topologiemodell

Das Topologiemodell stellt die logischen Beziehungen (Nachbarschaften) der im Geometriemodell repräsentierten Geometrieelemente dar. Geometrische Abmaße sind beim Topologiemodell irrelevant. Durch das Topologiemodell wird ein geometrisch dargestelltes Objekt erst physikalisch präsent.

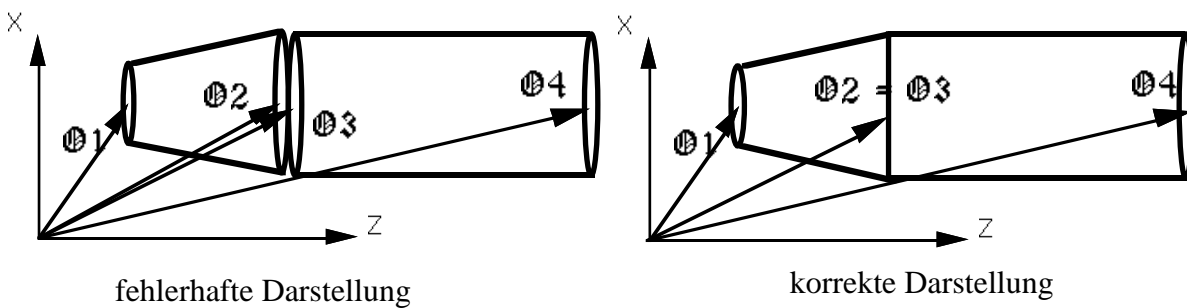
Befinden sich zwei rein geometrisch repräsentierte Gebilde im Raum und führen ihre räumliche Lage und die Abmaße zu einer Verschneidung der Gebilde, so wird dies erst im Topologiemodell deutlich. Dies führt zur Bildung von Schnittkanten und -flächen sowie zur Definition von Nachbarschaftsbeziehungen der beiden Gebilde untereinander. Die Definition von Nachbarschaften geschieht in Bezug auf die Gesamtgeometrie sowie auf Flächen, Kanten und Knoten. Dies bedeutet, daß zunächst einmal zwei geometrische Gebilde als gesamtes als benachbart, also aneinander angrenzend, erkannt werden. Jede der Flächen, aus der ein einzelnes Gebilde besteht, wird zudem durch ihre Begrenzungskanten und diese wiederum durch ihre Knotenpunkte definiert. Dadurch, daß beispielsweise eine Kante zwei unterschiedlichen Flächen zugeordnet ist, wird erkennbar, daß diese Kante die Begrenzungskante beider Flächen ist und sie demnach benachbart sein müssen. Entsprechend kann ein Knotenpunkt, der Element dreier Kanten ist, als Eckpunkt eines (dreidimensionalen) geometrischen Gebildes identifiziert werden (vgl. Würfel mit 8 solcher Knoten).

Zweck der Bildung eines Topologiemodells ist es, dem Rechner eine klare Vorstellung des geometrischen Gebildes zu geben. 'Innen' und 'Außen' wird so erkennbar und Rundungsfehler bei der Berechnung können erkannt und korrigiert werden, wenn durch das Topologiemodell beispielsweise das Aufeinanderliegen zweier Knotenpunkte definiert wurde, dies aber rechnerisch durch numerische Fehler nicht der Fall ist (vgl. nachfolgende Abbildungen).

In der hier beschriebenen Anwendung wird von der theoretischen Annahme der Unendlichkeit von beschriebenen Geometrieelementen abgewichen. Die beschriebenen Geometrie-Primitive sind bereits in endlicher Ausdehnung repräsentiert und besitzen somit bereits Berandungslinien. Die Beschreibung sich ergebender Schnittkurven ist hier irrelevant, da dies von einem vorausgesetzten CAD-System übernommen werden kann.



Der nachfolgend abgebildete Fehler kann somit in dieser Repräsentation nicht auftreten.



Geometrische Punkte (engl. *point*) des Geometriemodells werden durch das Topologiemodell zu Knoten (engl. *vertex*), Linien (engl. *line*, *track*) werden zu Kanten (engl. *edge*) und Ebenen (engl. *surface*) zu begrenzten Flächen (engl. *face*).

Topologische Nachbarschaftsbeziehungen werden i.d.R. durch sog. Nachbarschaftsgraphen dargestellt. Die Baumstruktur eines solchen Graphen besteht aus einzelnen Knotenpunkten (engl. *vertex*, nicht mit einem *vertex* (Knoten) eines topologisch definierten geometrischen Gebildes zu verwechseln!). Jeder dieser Knoten repräsentiert vor der Definition der Topologie lediglich geometrische Gebilde, wie Oberflächen, Linien und Punkte. Erst durch die Nachbarschaftsbeziehungen werden diese einzelnen Baum-Knoten miteinander vernetzt und das repräsentierte Gebilde kann auch als solches identifiziert werden. Dabei wandeln sich die Punkte zu Knoten, die Linien zu Kanten und die Oberflächen zu abgeschlossenen Flächenelementen um.

---

Im hier beschriebenen Formalismus wurde zur Definition von Nachbarschaftsbeziehungen ein ungerichteter Nachbarschaftsgraph gewählt, der mit Hilfe einer Menge von im folgenden definierten Entities aufgebaut wird:

**Nachbarschafts-Beziehung (Neighbourhood Relation):**

**NR ( element\_reference\_no1: <num>,  
element\_reference\_no2: <num> )**

Attribute:

element\_reference\_no1: Element-Nr. 1

element\_reference\_no2: Element-Nr. 2

Konsistenzbedingungen: Die referenzierten Elemente müssen direkt benachbart sein, also eine gemeinsame (topologische) Kante besitzen.

## 6 Produktstruktur- und Materialmodell

Bei der Repräsentation der Faserverbundwerkstoffe im Materialmodell soll nicht auf die unterste Ebene bis hin zur Faser und Matrix selbst, sondern nur auf die Halbzeugebene eines fertigen Faser-Matrix-Verbundes (Laminat) zurückgegangen werden. Die einzelnen Werkstoffkomponenten sollen zwar durch entsprechende Bezeichner transparent bleiben, jedoch werden Kennwerte (hier: Ingenieurkonstanten) nur für das gesamte Laminat angegeben und berücksichtigt. Zur eindeutigen Materialbeschreibung wird im Modell zwischen (anisotropen) Faserverbundwerkstoffen (FVW) und isotropen Materialien unterschieden.

Da der Behälter als Hohlkörper in seiner Struktur prinzipiell immer als eine Aufeinandererschichtung von Lagen verschiedenster Materialien - auch Nicht-Faserverbundwerkstoff-Lagen - gesehen werden kann, wurde diese Betrachtungsweise, unterstützt von der im Faserverbundsektor ohnehin üblichen lagenweisen Charakterisierung des Materialaufbaues, für die vollständige Struktur- und Materialbeschreibung des Behälters übernommen. Dazu wurden drei Produkt-Struktur-Entities und drei Material-Entities entwickelt, die eine Produktbeschreibung möglichst ohne Verwendung von Listen, bei weiter unbegrenzter Anzahl von Lagen, zum Ziel hatte. Dies wurde, aufgrund der Übersichtlichkeit mit einer Ausnahme, verwirklicht.

Das **Produkt-Struktur-Element** (**Product-Structural-Element** PSE) dient der Gruppierung von Teilgeometrie-Elementen zu einer Lage, indem es Geometrielemente einer zusammengehörenden Behälterlage in Form einer Liste beschreibt. Damit wird erreicht, daß geometrisch getrennte Lagen als eine zusammenhängende Lage, die sich über mehrere Geometrielemente erstrecken kann, erkannt werden.

Das **Lagen-Abfolge-Element** (**Layer-Order-Element** LOE) beschreibt die Aufeinanderfolge der einzelnen in der Lagenstruktur LSE beschriebenen Behälterlagen. Somit können auch solche Fälle beschrieben werden, wo eine Lage zwei oder mehrere verschiedene Lagen überdeckt, wie beispielsweise eine Lackschicht, welche sich auf einer abschließenden Umfangswicklung im zylindrischen Teil des Behälters und einer zum Teil darunterliegenden Kreuzwicklung an den Behälterböden befindet. Demnach hat die Lackschicht zwei nachfolgende Lagen-Struktur-Elemente, was durch die gewählte Beschreibungsform problemlos bewältigt werden kann. Für die erste Lage wird als Vorgängerelement *NIL*. angegeben.

Das **Lagen-Design-Element** (**Layer-Design-Element** LDE) beschreibt fortführend den Materialaufbau, beziehungsweise auf ein im voraus definiertes Produkt-Struktur-Element.

Das **Material-Design-Element** (**Material-Design-Element** MDE) wird im vorgenannten LDE referenziert und beschreibt - einem weiter folgenden Material-Eigenschaften-Element für die Beschreibung von sog. Elementarlagern übergeordnet - das verwendete Material der Lagenstruktur nebst der Anzahl verwendeter Struktur-Lagen und - bei Bedarf - dem Faser-Wickelwinkel. Wird ein Nicht-Faserverbundwerkstoff referenziert, so bekommt das Attribut 'Wickelwinkel' den Wert *NIL*.

Als letztes Glied der Materialhierarchie spezifiziert das **Basis-Material-Eigenschaften-Element** (**Base-Material-Properties** BMP) die genauen Materialeigenschaften des im Material-Design-Element referenzierten Materials. Dabei wird zwischen Verbund- (**Composite-Base-Material-Properties** CBMP) und Nicht-Verbundwerkstoffen (nur BMP) differenziert, um der unterschiedlichen Charakterisierung aufgrund anisotroper Eigenschaften aufseiten der Faserverbunde einerseits und deren hybriden Aufbau aus mindestens zwei Materialien andererseits gerecht zu werden.

Mit dieser letzten Entity kann ein Grundstock an verschiedenen Materialien in Form einer Bibliothek aufgebaut werden, auf die produktunabhängig nach Bedarf zugegriffen werden kann.

**Product-Structural-Element** (PSE):

```
PSE ( <num> {, <num>}* )
```

**Layer-Order-Element** (LOE):

```
LOE ( LDE_no1: <num>,
      LDE_no2: <num> )
```

**Layer-Design-Element** (LDE):

```
LDE ( layer_name: <string>,
      PSE_no: <num>,
      MDE_no: <num> )
```

**Material-Design-Element MDE)**

```
MDE ( type: [BMP; CBMP],  
      mat_no: <num>,  
      no_elem_layers: <num>,  
      winding_angle: <num> )
```

**Base-Material-Properties (BMP):**

```
BMP ( mat_name: <string>,  
      elt: <num>,  
      s: <num>,  
      epsf: <num>,  
      e_mod: <num> )
```

**Composite-Base-Material-Properties (CBMP):**

```
CBMP( c_mat_name: <string>,  
      elt: <num>,  
      phivol: <num>,  
      df: <num>,  
      sf1: <num>,  
      sf2: <num>,  
      e_modf1: <num>,  
      e_modf2: <num>,  
      epsff1: <num>,  
      epsff2: <num>,  
      sm: <num>,  
      e_mod: <num>,  
      epsfm: <num> )
```



## 7 Oberflächenmodell

### 7.1 Oberflächengüte (Surface Finish)

Oberflächengüten können nach verschiedenen Ermittlungsverfahren angegeben werden, wobei eine Umrechnung zwischen den einzelnen Verfahren nicht korrekt durchgeführt werden kann. Deshalb muß zusätzlich zum angegebenen Wert noch die Bezeichnung des Meßverfahrens angegeben werden.

```
SF(    surface_number: <num>,
      surface_finish: [Rt, Rz, Rp, Ra,...],
      value: <num>)
```

Attribute:

surface_number	Flächennummer
surface_finish	Oberflächenangabe, wie etwa Rt (Rauhtiefe) oder Rp (Glättungstiefe)
value	Maß der Oberflächengüte

Konsistenzbedingungen: Der Wert value muß im Intervall der Oberflächengüte surface\_finish liegen

### 7.2 Härte (Hardness)

Für Härteangaben gilt das gleiche wie für die Angabe von Oberflächenangaben.

```
H(    surface_number: <num>,
      hardness: [HRC, HRB, HV, HB, ...],
      value: <num>)
```

Attribute:

surface_number	Flächennummer
hardness	Härteangabe, wie etwa HRC (Härte in Rockwell Cone) oder HV (Vickers Härte)
value	Maß der Härteangabe

Konsistenzbedingungen: Der Wert value muß im Intervall der Härteangabe hardness liegen

## 8 Toleranzmodell

Das Toleranzmodell unterscheidet Form- und Lagetoleranz. Auf einer höheren Ebene ist die Maßtoleranz im allgemeinen zu sehen, die zunächst einmal die Genauigkeit des Maßstabes selbst definiert.

Die Formtoleranz gibt die erlaubten Abweichungen der Realgeometrie gegenüber der geforderten Geometrie an. Darunter fallen die Begriffe

- Geradheit
- Ebenheit
- Rundheit
- Zylindrizität
- Linienform
- Flächenform.

Die Lagetoleranz hingegen drückt die erlaubte Abweichung der realen Position von der Sollposition aus. Darunter fallen die Begriffe

- Rechtwinkligkeit
- Neigung
- Konzentrizität.

Zur Überprüfung der Forderungen von Form- und Lagetoleranz werden Kontrollflächen mit den Grenzabmessungen und -lagen in geeigneter Weise um die jeweiligen Geometrien gelegt. Ergibt sich ein Schnitt zwischen der zu überprüfenden Geometrie und einer der Kontrollflächen, so ist die jeweilige Toleranz nicht erfüllt.

Im folgenden sind die entsprechenden Toleranzformen näher erläutert.

## 8.1 Maßtoleranz (Size Standard)

Maßtoleranzen beziehen sich entweder auf die Maße eines Geometrieelements oder sie beziehen sich auf Maße zwischen zwei Geometrieelementen. Aus diesem Grund werden die Beschreibungselemente SS und SSD benutzt. Mit SS lassen sich vor allem Durchmesser toleranzen beschreiben, während mit SSD z.B. die Längentoleranz eines Zylinders über den Abstand der ihn begrenzenden Kreisflächen beschrieben werden kann.

```
SS(  surface_number: <num>,
     nominal_size: <num>,
     min_size: <num>,
     max_size: <num>)
```

Attribute:

surface_number	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$$\text{min\_size} \leq \text{nominal\_size} \leq \text{max\_size}$$

```
SSD( surface_number1: <num>,
     surface_number2: <num>,
     nominal_size: <num>,
     min_size: <num>,
     max_size: <num>)
```

Attribute:

surface_number1	Elementnummer der Bezugsfläche
surface_number2	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$$\text{min\_size} \leq \text{nominal\_size} \leq \text{max\_size}$$

$$\text{surface\_number1} \neq \text{surface\_number2}$$

## 8.2 Allgemeintoleranz DIN 7168 (General Tolerance)

Allgemeintoleranzen bestehen generell zwischen allen bemaßten Elementen eines Produkts und müssen prinzipiell immer in entsprechende Maßtoleranzen umgerechnet werden, um sie überprüfen zu können (vgl. Maßtoleranzen).

**GT( surface\_number: <num>,  
nominal\_size: <num>,  
min\_size: <num>,  
max\_size: <num>)**

Attribute:

surface_number	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$$\text{min\_size} \leq \text{nominal\_size} \leq \text{max\_size}$$

**GTD( surface\_number1: <num>,  
surface\_number2: <num>,  
nominal\_size: <num>,  
min\_size: <num>,  
max\_size: <num>)**

Attribute:

surface_number1	Elementnummer der Bezugsfläche
surface_number2	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$$\text{min\_size} \leq \text{nominal\_size} \leq \text{max\_size}$$

$$\text{surface\_number1} \neq \text{surface\_number2}$$

### 8.3 Formtoleranz DIN 7184 (Shape Tolerance)

Mit diesen Beschreibungselementen soll die geometrische Ausprägung von bestimmten Bereichen eines Werkstücks genauer beschrieben werden können.

```
ST(   surface_number: <num>,
      tolerance_type: [ SN, EN, RN, C, LD, PD],
      tolerance_size: <num>)
```

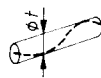
Attribute:

surface_number	Elementnummer der Bezugsfläche
tolerance_type	Toleranztyp
tolerance_size	Toleranzmaß

Konsistenzbedingungen: tolerance\_type muß mit der Bezugsfläche surface\_number kompatibel sein.

Im folgenden sind die möglichen Toleranztypen aufgezählt. Die Toleranzangaben sind immer in mm zu lesen!

**SN**



**Geradheit (Straightness)**

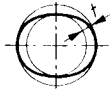
Die tolerierte Achse des äußeren Zylinders muß innerhalb eines Zylinders vom Durchmesser t liegen.

**EN**



**Ebenheit (Evenness)**

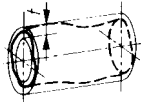
Die tolerierte Fläche muß zwischen zwei parallelen Ebenen vom Abstand t liegen.



RN

**Rundheit (Roundness)**

In jeder achsenkrechten Schnittebene muß die tolerierte Umfangslinie zwischen zwei konzentrischen Kreisen vom Abstand  $t$  liegen.



C

**Zylinder (Cylinder)**

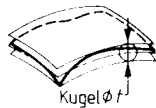
Die tolerierte Zylindermantelfläche muß zwischen zwei coaxialen Zylindern liegen, die einen Abstand von  $t$  haben.



LD

**Linienform (Line Divergency)**

In jeder Schnittebene, parallel zur Zeichenebene, muß das tolerierte Profil zwischen Hülllinien an Kreisen vom Durchmesser  $t$  liegen, deren Mittelpunkte auf der geometrisch idealen Linie liegen.



PD

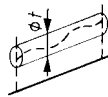
**Flächenform (Plane Divergency)**

Die tolerierte Fläche muß zwischen zwei Hüllflächen an Kugeln vom Durchmesser  $t$  liegen, deren Mittelpunkte auf der geometrisch idealen Fläche liegen.

## 8.4 Lagetoleranzen (Positional Tolerance)

Durch diese Beschreibungselemente sollen Bezüge zwischen einzelnen Bereichen eines Werkstücks dargestellt werden. In der vorgestellten Repräsentation werden Bezugsachsen nicht explizit repräsentiert, sondern werden durch die Bestimmung von Richtungsvektoren, die nach angegebenen Vorschriften aus den Attributen der Elementbeschreibungen berechnet werden, implizit dargestellt. Zur Angabe der Bezugsachse genügt somit die Angabe des Elements, auf dessen Bezugsachse (Richtungsvektor) referenziert werden soll.

Bei rotationssymmetrischen Elementen ist der Richtungsvektor immer die z-Achse und muß somit nicht mehr berechnet werden.



### Parallelität (Parallelism)

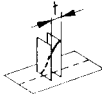
Die tolerierte Achse eines Elements muß zwischen zwei zur Bezugsachse des Elements parallelen Ebenen vom Abstand  $t$  liegen.

```
PM(  surface_number: <num>,
      reference_axis: <num>,
      tolerance_size: <num>)
```

Attribute:

surface_number	Flächennummer
reference_axis	Die Nummer der Fläche, auf deren Achse Bezug genommen wird
tolerance_size	Toleranzmaß

Konsistenzbedingungen: Die Fläche `surface_number` und die Achse `reference_axis` müssen mit dem Toleranztyp kompatibel sein.

**Rechtwinkligkeit (Orthogonality)**

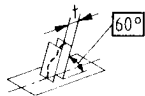
Eine tolerierte Planfläche muß zwischen zwei parallelen und zur Bezugsachse senkrechten Ebenen vom Abstand  $t$  liegen.

**O( surface\_number: <num>,  
reference\_axis: <num>,  
tolerance\_size: <num>)**

Attribute:

surface_number	Flächennummer
reference_axis	Die Nummer der Fläche, auf deren Achse Bezug genommen wird
tolerance_size	Toleranzmaß

Konsistenzbedingungen: Die Fläche `surface_number` und die Achse `reference_axis` müssen mit dem Toleranztyp kompatibel sein.

**Neigung (Gradient)**

Die tolerierte Achse eines Elements muß zwischen zwei parallelen und im Winkel von  $a$  Grad zur Bezugsfläche geneigten Ebenen liegen, die einen Abstand von  $t$  mm haben.

**G( surface\_number1: <num>,  
surface\_number2: <num>,  
tolerance\_size: <num>  
angle: <num>)**

Attribute:

surface_number1	Flächennummer 1
surface_number2	Flächennummer 2

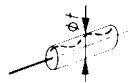


tolerance_size	Toleranzmaß
angle	Winkel zwischen den beiden Flächen

Konsistenzbedingungen:

surface\_number1  $\neq$  surface\_number2

Die beiden Flächen müssen mit dem Toleranztyp kompatibel sein



### Konzentrität (ConCentricity)

Die tolerierte Achse eines Elements muß innerhalb eines zur Bezugsachse koaxialen Zylinders vom Durchmesser t liegen.

**CC( surface\_number: <num>,  
reference\_axis: <num>,  
tolerance\_size: <num>)**

Attribute:

surface_number	Flächennummer
reference_axis	Die Nummer der Fläche, auf deren Achse Bezug genommen wird
tolerance_size	Toleranzmaß

Konsistenzbedingungen: Die Fläche surface\_number und die Achse reference\_axis müssen mit dem Toleranztyp kompatibel sein.

## 9 Literaturverzeichnis

- /1/ Bernardi, A., Klauck, Ch., Legleitner, R.: STEP: Überblick über eine zukünftige Schnittstelle zum Produktdatenaustausch. Dokument D-90-04, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Postfach 2080, D-67608 Kaiserslautern, September 1990.
- /2/ Anderl, R., Grabowski, H., Schilli, B., Schmitt, M.: STEP - Entwicklung einer Schnittstelle zum Produktdatenaustausch. VDI-Z 131 (1989), Nr. 9 - September 1989, p. 68-76
- /3/ Bernardi, A., Klauck, Ch., Legleitner, R.: TEC-REP: Repräsentation von Geometrie- und Technologieinformationen. Dokument D-91-07, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Postfach 2080, D-67608 Kaiserslautern, Juni 1991

## 10 Beispiel-Geometrie- und Topologiebeschreibung anhand eines erdachten Behälters

Der in der Skizze ersichtliche Beispiel-Behälter besteht aus einem zylindrischen Mantel. Dieser ist beidseitig jeweils mit einem Boden, bestehend aus einer torusabschnittförmigen Krempe und einer sich anschließenden Kugelkalotte, abgeschlossen. Radial mittig in Bezug auf die Behälter-Hauptachse (Rotationsachse, Koordinate z) sitzt ein zylindrischer, auf einer Aushalsung befindlicher Stutzen.

Unter Verwendung der im voraus definierten Beschreibungen der Primitive läßt sich der Behälter wie folgt darstellen. Die Abmaße sind der Skizze zu entnehmen.

Das gedachte globale Bezugskordinatensystem für die Positionsbeschreibung der lokalen Koordinatensysteme befindet sich am linken Ende auf der Rotationsachse des zylindrischen Kegelmantels.

(\* GEOMETRY \*)

```
@ 0 := GCS (
      type: CS,
      unit: mm )

@ 1 := V (
      name: Beispiel_Behälter,
      application: Storage_Vessel,
      material: GFRP,
      entities: ( 1 2 3 4 5 6 7 8 11 12 13 14 15 100 110 120 130 140 150
                  160 500 510 600 610 700 710 1000 1010 1500 1510 ),
      info: " Beispiel eines Druckbehälters" )

@ 2 := CJ (
      coordinate_system: #11,
      reference_point1: 0,
      reference_point2: 400,
      radius: 125 )
```

@ 3 := TS (  
coordinate\_system: #11  
reference\_point1: 0,  
reference\_point2: -39.65  
reference\_point3: 0,  
radius1: 125,  
radius2: 105.45,  
radius3: 75  
direction\_of\_arch: x )

@ 4 := BJ (  
coordinate\_system: #12,  
center: (189.65,0,0),  
radius: 217,  
height: 25,  
joint\_direction: - )

@ 5 := TS (  
coordinate\_system: #13  
reference\_point1: 0,  
reference\_point2: 39.65  
reference\_point3: 0,  
radius1: 125,  
radius2: 85,  
radius3: 75  
direction\_of\_arch: x )

@ 6 := BJ (  
coordinate\_system: #14,  
center: (-189.65,0,0),  
radius: 217,  
height: 25,  
joint\_direction: + )

---

@ 7 := CJ (  
coordinate\_system: #15,  
reference\_point1: 0,  
reference\_point2: 50,  
radius: 25 )

@ 8 := CUR (  
element\_reference\_no1 : #2,  
element\_reference\_no2 : #7,  
radius : 10 )

@ 11 := LCS (  
coordinate\_system: #0  
origin: (0,0,0),  
direction\_vector1: (1,0,0)  
direction\_vector2: (0,1,0),  
lcs\_type: CS,  
unit: mm )

@ 12 := LCS (  
coordinate\_system: #0  
origin: (-39.65,0,0),  
direction\_vector1: (-1,0,0)  
direction\_vector2: (0,1, $\pi$ ),  
lcs\_type: CS,  
unit: mm )

@ 13 := LCS (  
coordinate\_system: #0  
origin: (400,0,0),  
direction\_vector1: (1,0,0)  
direction\_vector2: (0,1,0),  
lcs\_type: CS,  
unit: mm )

@ 14 := LCS (  
coordinate\_system: #0  
origin: (439.65,0,0),  
lcs\_type: CS,  
direction\_vector1: (1,0,0)  
direction\_vector2: (0,1,0) )

@ 15 := LCS (  
coordinate\_system: #0  
origin: (200,130, $\pi/2$ ),  
direction\_vector1: (200,135, $\pi/2$ )  
direction\_vector2: (200,135, $\pi/4$ ),  
lcs\_type: CS,  
unit: mm )

(\* TOPOLOGY \*)

@ 100 := NR (  
element\_reference\_no1: 3,  
element\_reference\_no2: 4 )

@ 110 := NR (  
element\_reference\_no1: 2,  
element\_reference\_no2: 3 )

@ 120:= NR (  
element\_reference\_no1: 2,  
element\_reference\_no2: 5 )

@ 130:= NR (  
element\_reference\_no1: 5,  
element\_reference\_no2: 6 )

@ 140:= NR (  
element\_reference\_no1: 2,

---

element\_reference\_no2: 8 )

@ 150:= NR (  
element\_reference\_no1: 7,  
element\_reference\_no2: 8 )

@ 160:= NR (  
element\_reference\_no1: 2,  
element\_reference\_no2: 5 )

(\* VESSEL DESIGN \*)

@ 500:= PSE (  
#2,#3,#4,#5,#6,#7,#8 )

@ 510:= PSE (  
#2,#3,#4,#5,#6,#8 )

@ 600:= LDE (  
layer\_name: LINER,  
PSE\_no: #500,  
MDE\_no: #1000 )

@ 610:= LDE (  
layer\_name: LAYER\_1,  
PSE\_no: #510,  
MDE\_no: #1010 )

@ 700:= LOE (  
LDE\_no1: NIL,  
LDE\_no2: #600 )

@ 710:= LOE (  
LDE\_no1: #600,  
LDE\_no2: #610 )

@ 1000:= MDE (

BMP\_no: #1500,  
no\_elem\_layers: 1,  
winding\_angle: NIL )

@ 1010:= MDE (  
CBMP\_no: #1510,  
no\_elem\_layers: 3,  
winding\_angle: 20 )

@ 1500:= BMP (  
mat\_name: POLYETHYLEN\_PE,  
elt: 3,  
s: 20,  
epsf: 5,  
e\_mod: 700 )

@ 1510:= CBMP (  
c\_mat\_name: GF\_PE,  
elt: 0.4,  
phivol: 0.3,  
df: 10e-3,  
sf1: 2000,  
sf2: 2000,  
e\_modf1: 76000,  
e\_modf2: 76000,  
epsff1: 0,02,  
epsff2: 0,02,  
sm: 20,  
e\_mod: 700,  
epsfm: 5 )