ProgressXML

Index:

1 ALLGEMEINES 4
1.1 Anwendungsbereich 4
1.2 Default-Werte 4
1.3 Character Encoding 5
1.4 Versioning 5
1.5 Kompatibilität mit UNICAM 6
2 STRUKTUR-ÜBERSICHT 7
3 DETAIL-SPEZIFIKATIONEN 13
3.1 Global ID 13
3.1.1 Eindeutigkeit der GlobalID 13
3.1.2 Sonderfall: GlobalID der DocInfo-Tabelle 13
3.1.3 Automatisches generieren von GlobalIDs 13
3.2 DocInfo 14
3.2.1 GlobalID 14
3.2.2 Document Version 14
3.2.3 Comment 14
3.2.4 ConvertConventions 14
3.2.5 Mode 14
3.3 Order 16
3.3.1 Specific Order Info 16
3.3.2 Delivery Date 16
3.3.3 GenericOrderInfo 16
3.3.4 Comment 16
3.3.5 ImportSource, ImportSourceType 16
3.3.6 ApplicationName, ApplicationGUID, ApplicationVersion 16
3.4 Product (Element) 17
3.4.1 ProductType 17
3.4.2 TotalThickness, DoubleWallsGap 17
3.4.3 PieceCount 18
3.4.4 TurnWidth, Datenübergabe für Doppelwände 18
3.4.5 Comment 18
3.4.6 RotationPosition 19
3.5 Slab (Elementteil, Platte) 19
3.5.1 SlabNo 19
3.5.2 PartType 19
3.5.3 Various Slab Information 19
3.5.4 Slab-Type 19
3.5.5 ProjectCoordinates 19
3.5.6 Coordinates of Element in Pile 19
3.5.7 ReforcemInfo 20
3.5.8 Generic Slab Info 20
3.5.9 Multi-Layer Elements 20
3.6 Outline 20
3.6.1 X/Y/Z 20
3.6.2 Height 20
3.6.3 Name 20
3.6.4 GenericInfo 21
3.6.5 MountingInstruction (only for mountparts) 21
3.6.6 MountPartType, MountPartArticle(only for mountparts) 21
3.6.7 MountPart Properties (only for mountparts) 21
3.6.8 Concrete Properties (only for lots) 21
3.6.9 Layer 21
3.6.10 Shape, SVertex 21
3.7 Steel 23
3.7.1 X/Y/Z 23
3.7.2 ToTurn (Nur für Matten) 23
3.7.3 StopOnTurningSide (Nur für Matten) 23
3.7.4 Name 23
3.7.5 MeshType 23
3.7.6 WeldingDensity (Nur für Matten) 24

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.7.7 BorderStrength ... 24
3.7.8 Generic Steel Info ... 24
3.7.9 ProdRotX/Y/Z ... 24
3.7.10 Layer ... 25
3.8 Bar ... 25
3.8.1 ShapeMode ... 25
3.8.1.1 ShapeMode "realistic" ... 25
3.8.1.2 ShapeMode "schematic" ... 25
3.8.1.3 ShapeMode "polygonal" ... 26
3.8.1.4 Automatische Ermittlung des ShapeMode und gemischte Darstellungen ... 27
3.8.2 ReinforcementType (Bewehrungslagen) ... 28
3.8.2.1 Definition der Bewehrungslagen ... 28
3.8.2.2 Festlegung der Bewehrungslagen ... 28
3.8.2.3 Obere Bewehrungslagen ... 28
3.8.3 SteelQuality ... 29
3.8.4 PieceCount, Diameter, X, Y, Z ... 29
3.8.5 RotZ ... 29
3.8.6 ArticleNo ... 30
3.8.7 NoAutoProd ... 30
3.8.8 ExtIronWeight ... 30
3.8.9 Bin ... 30
3.8.10 Pos ... 30
3.8.11 Note ... 30
3.8.12 Machine ... 30
3.8.13 BendingDevice ... 31
3.8.14 Spacer ... 31
3.8.14.1 Type ... 31
3.8.14.2 Position ... 31
3.8.15 WeldingPoint ... 31
3.8.15.1 WeldingOutput ... 31
3.8.15.2 Position ... 31
3.8.15.3 WeldingPointType, WeldingPrgNo ... 32
3.8.16 Segment ... 33
3.8.16.1 Segment-Orientation (RotX, BendY) ... 33
3.8.16.2 Segment-Length (L) ... 33
3.8.16.3 Bending-Radius (R) ... 33
3.8.16.4 Segment-Außenmaße ... 35
3.8.16.5 Außenlänge eines Segmentes ... 36
3.8.16.6 Höhe und Breite eines Segmentes ... 36
3.8.16.7 Rechenregeln für Außenmaße ... 36
3.8.16.8 Konventionelle Außenmaße ... 36
3.8.16.9 Allgemeine Berechnungen zum Biege-Radius ... 37
3.8.16.10 Bögen und Spiralen in klassischer Spiral-Form ... 37
3.8.16.11 Bögen normaler PXML-Form ... 38
3.8.16.12 Allgemeine Berechnungen zur Koordinaten-Drehung ... 40
3.8.16.13 Konvertierung von und zu UNICAM ... 42
3.8.16.14 Konvertierung von und zu BVBS-BF2D ... 44
3.8.16.15 Konvertierung von und zu BVBS-BF3D ... 44
3.9 Girder ... 45
3.9.1 PieceCount, X, Y, Z, GirderName, Length, AngleToX ... 45
3.9.2 NoAutoProd ... 46
3.9.3 Height, TopExcess, BottomExcess ... 46
3.9.4 Weight, TopFlangeDiameter, BottomFlangeDiameter ... 46
3.9.5 GirderType ... 46
3.9.6 MountingType ... 46
3.9.7 Machine ... 46
3.9.8 Period, PeriodOffset ... 47
3.9.9 Width ... 47
3.9.10 AnchorBar ... 47
3.9.11 GirderExt ... 47
3.9.11.1 Type = splicePos ... 47
3.9.11.2 Type = FixingPos ... 47
3.9.11.3 Type = GirderGripPos ... 48
3.9.11.4 Type = MeshGripPos ... 48
3.9.11.5 Type = SupportPos ... 48
3.9.11.6 Export nach UNICAM ... 48
3.9.12 Section ... 48
3.9.12.1 Felder der Section-Tabelle ... 48
3.9.12.2 Zuordnung der Section-Angaben ... 48
3.10 Alloc ... 50

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.10.1 GuidingBar (Leiteisen) ... 50
3.10.2 Richtungsbestimmung ohne GuidingBar ... 50
3.10.3 Region ... 51
3.11 SteelExt ... 52
3.12 Feedback ... 53
3.12.1 Production Test Service (PTS) ... 53
3.12.2 Machine Return ... 53
3.12.3 Felder des Feedback-Blocks ... 54
3.12.3.1 Feedback-Attribute ... 54
3.12.3.2 Feedback-Felder ... 54
3.12.4 Beispiele für PTS-Meldungen ... 55
3.12.5 Beispiele für Machine-Return-Meldungen ... 56
3.12.6 Kommunikations-Arten des PTS ... 57
3.12.6.1 Kommunikation auf File-Basis ... 58
3.12.6.2 Kommunikation über TCP/IP-Sockets ... 58
3.12.7 Kommunikations-Arten des Machine Return ... 58
3.12.7.1 Kommunikation auf File-Basis ... 58
3.12.7.2 Kommunikation über TCP/IP-Sockets ... 58
3.12.1 Parallel- und Reihenschaltung von PTS-Servers ... 58
3.12.2 Filtern, klassifizieren und sortieren von PTS-Meldungen ... 59
4 VORSCHLÄGE FÜR ZUKÜNFTIGE ERWEITERUNGEN ... 60
4.1 Erweiterung der Shapes auf 3D – Differenz-Methode ... 60
4.2 Drehung für Outline ... 60
5 VERSIONS-HISTORY ... 61

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

1 Allgemeines

1.1 Anwendungsbereich

Das Datenformat **ProgressXML** (im Folgenden auch **PXML** genannt) ist ein auf XML basierendes Datenformat für die Daten-Erstellung und Produktionssteuerung in Betonfertigteilwerken.

Im Einzelnen gibt es 2 unterschiedliche Anwendungsbereiche:

- Schnittstelle zwischen Systemen verschiedener Hersteller →*Standard-Tags*
- Interne (proprietäre) Daten-Speicherung von CAD/CAM-Systemen→*Interne Tags*

Im vorliegenden Dokument werden die **Standard-Tags** beschrieben: das sind jene Felder des PXML-Formates, die verbindlich definiert sind und für den Datenaustausch zwischen verschiedenen Systemen verwendet werden können. Es ist *nicht* notwendig, alle Standard-Tags zu verwenden; man sollte jedoch versuchen, beim Daten-Austausch ausschließlich die Standard-Tags zu verwenden.

Neben den Standard-Tags kann jeder Software-Hersteller für seine interne Daten-Repräsentation beliebige proprietäre **interne Tags** definieren und verwenden. Diese sollen beim Daten-Austausch zwischen verschiedenen Systemen einfach ignoriert werden.

Es wird empfohlen, internen Tags stets mit Präfix `I_` voranzustellen; dadurch werden Namens-Konflikte mit später eingeführten Standard-Tag vermieden (Standard-Tags beginnen niemals mit `I_`)[1].

Aus obigen Ausführungen ergeben sich für Systemübergreifende PXML-Import-Module folgende Forderungen:

a) Es werden nur Standard-Tags gelesen.
b) Das Fehlen von Standard-Tags muss toleriert werden (Default-Werte werden angenommen, siehe Abschnitt 1.2).
c) Die PXML-Datei kann beliebig viele zusätzliche Tags enthalten (interne Tags), die beim Import einfach ignoriert werden.

1.2 Default-Werte

Beim Fehlen eines Tags werden Default-Werte angenommen. Die Default-Werte sind für jeden Datentyp einheitlich vorgegeben:

- **string**: ""
- **double**: 0
- **bool**: false

Bei anderen Datentypen, als den oben angeführten, sollten niemals Default-Werte angenommen werden. Das gilt insbesondere auch für Integer-Werte: da dies häufig IDs beinhalten, die auch 0 sein könnten, gibt es für Integers keine Default-Wert-Regelung.

[1] Eine Variante dieser Konvention ist es die Präfixe **I_P_** und **I_V_** zu verwenden: ersterer wird für *persistente* interne Tags verwendet, das sind jene, die auch auf File geschrieben werden. Letzterer wird für *flüchtige* (*volatile*) interne Tags verwendet, die *nicht* auf File geschrieben werden und i.allg. eine redundante Information enthalten (Cache-Felder).

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

1.3 Character Encoding

Es wird empfohlen in PXML-Dateien das UTF-8-Encoding zu verwenden. Man vermeidet dadurch viele Probleme und Komplikationen.

Da es sich aber um eine XML-Datei handelt, ist die Wahl des UTF-8-Codes jedoch nicht zwingend, sondern es gelten allgemein die Encoding-Regeln für XML-Dokumente:

a) Wenn es keine Encoding-Declaration[2] gibt, und kein BOM (Byte Order Mark) vorhanden ist, handelt es sich um UTF-8.

b) Für UTF-16 und UTF-32 Encodings ist ein entsprechender BOM-Eintrag zwingend. Eine Encoding-Deklaration ist zwar nicht zwingen, aber empfohlen.

c) Falls ein altes (nicht-Unicode-fähiges) Encoding verwendet wird, muss eine explizite Encoding Deklaration angeführt werden. Typischerweise ist das "iso-8859-1" oder "windows-1252", für den üblichen sogenannten "ANSI"-Code.

1.4 Versioning

Es gibt 2 Versions-Kennungen: **Major-Version** und **Minor-Version**. Die Kombination *Major-Version /Minor-Version* stimmt mit der Versions-Kennung des vorliegenden Dokumentes überein.

Beachte: diese Versions-Kennungen beziehen sich nur auf die Standard-Tags; falls es ein Versioning für interne Tags gibt, muss dies über eine zusätzliche proprietäre Versions-Kennung erfolgen.

Major-Version (Schema-Version)

Die *Major-Version* wird im XML-Namespace festgelegt. Der Namespace hat folgendes Format:

`http://progress-m.com/ProgressXML/Version1`

Die *Major-Version* wird nur dann verändert, wenn sich die Schema-Syntax auf inkompatible Weise ändert.

Änderungen der *Major-Version* sollten, wenn möglich, vermieden werden. Ein Wechsel der *Major-Version* ist normalerweise nur dann nötig, wenn die vorhergehende Version (zumindest vorübergehend) weiterhin gepflegt werden muss.

Minor-Version

Die *Minor-Version* wird im DocInfo-Block eingetragen.

Die verschiedenen Minor-Versionen müssen syntaktisch kompatibel zueinander sein. D.h. eine Version kann Tags besitzen, die es in der anderen Version nicht gibt, oder ein- und derselbe Tag kann in verschiedenen Versionen eine unterschiedliche semantische Bedeutung haben; die Schema-Syntax muss jedoch kompatibel sein.

Beispiel 1: eine Winkel-Angabe, die ursprünglich in Grad war, wird in einer neuen Version als Zehntel-Grad-Wert interpretiert.

Beispiel 2: eine Integer-Winkel-Angabe einer alten Version wird in einer neuen Version nicht mehr verwendet; stattdessen wird ein neues Tag mit einer Fließkomma-Winkel-Angabe eingeführt.

Beachte: Ergänzungen, die voll abwärtskompatibel sind, können auch innerhalb einer Minor-Version hinzugefügt werden. Typischerweise handelt es sich hierbei um Felder, die Zusätzlich in den PXML-Standard aufgenommen wurden.

---

[2] Als **Encoding Declaration** ist hier eine Kopfzeile der Form
<?xml version="1.0" encoding="iso-8859-1" ?>
zu verstehen.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

1.5 Kompatibilität mit UNICAM

Zielsetzung

Bei der Definition des ProgressXML-Formates wurde versucht eine weitgehende Kompatibilität mit der UNICAM CAD-CAM-Schnittstelle zu erreichen. Insbesondere soll es möglich sein, bei Konvertierungen

UNICAM → ProgressXML → UNICAM

Wieder die ursprüngliche Datei zu erhalten (von wenigen Ausnahmen abgesehen).

Diese Forderung hat einige Elemente, Benennungen und Strukturen in die Spezifikation einfließen lassen, die an und für sich eher irrational erscheinen.

Wenn ohne Versions-Angabe auf die UNICAM-Dokumentation verwiesen wird, sind i. allg. die Versionen 5.2 oder 6.0 der UNICAM-Schnittstelle gemeint.

Nicht unterstützte UNICAM-Elemente

Einige Elemente der UNICAM CAD-CAM-Schnittstelle sind in PXML bewusst nicht integriert worden:

1) **Ist-Stückzahl:** wird beim Export nach UNICAM immer 0 gesetzt.
2) **Endhaken-Biegeformen:** es werden nur freie Biegeformen unterstützt; beim Import aus UNICAM werden Endhaken-Biegeformen in freie Biegeformen konvertiert.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

2 Struktur-Übersicht

Die folgende Darstellung beschreibt die Struktur des PXML-Files. Details zu den einzelnen Feldern sind weiter unten angeführt.

Die rechts angeführten roten Zahlen-Angaben legen fest, wie oft ein Eintrag vorkommen kann oder muss:

- **1** = genau einmal
- **0,1** = 0 oder 1 mal
- **>=1** = mindestens einmal
- **n** = beliebig oft (auch 0 mal)

Die Längen der Zeichenketten sind grundsätzlich beliebig. Beim Export nach UNICAM werden die Zeichenketten jedoch auf eine maximale Länge reduziert, bzw. auf eine bestimmte Länge mit Leerzeichen aufgefüllt.

Die angegebenen Texte (z.B. aaaaaaaa) sind Beispiele. Wo nicht anders erwähnt, sind alle Texte beliebige alphanumerische Zeichenfolgen; es ist aber zu beachten, dass einige dieser freien Texte beim UNICAM-Export zu numerischen Werten konvertiert werden müssen. Jene Werte, die bereits in der PXML-Definition numerisch sind, haben in der Tabelle die Anmerkung **Int**, **Bool**, oder **Double**.

<PXML Document>		**1**
<DocInfo GlobalID="aaa">		**1**
<MajorVersion>0</MajorVersion>	Int	1
<MinorVersion>0</MinorVersion>	Int	1
<Comment>aaaaaaaaaaaaa</Comment>		0,1
<ConvertConventions>aaa#bbb#ccc</ConvertConventions>		0,1
<Mode>		**n**
<ID>aaaaa</ID>		1
<Val>true</Val>		0,1
</Mode>		
</DocInfo>		
<Order GlobalID="aaa">		**n**
<OrderNo>aaaaaaaa</OrderNo>		0,1
<Component>bbbbb</Component>		0,1
<Storey>cccc</Storey>		0,1
<DrawingNo>dddddd</DrawingNo>		0,1
<DrawingDate>dd.mm.yyyy</DrawingDate>		0,1
<DrawingRevision>ee</DrawingRevision>		0,1
<DrawingAuthor>aaaaaaaaaaaaaaaaaaaa</DrawingAuthor>		0,1
<Comment>aaaaaaaaaaaaaaaaaaaa</Comment>		0,1
<ImportSource>aaaaaaaaaaaaaaaaaaaa</ImportSource>		0,1
<ImportSourceType>aaaaaaaaaaaaaaaaaaaa</ImportSourceType>		0,1
<DeliveryDate>2010-04-16T11:46:48.933+02:00</DeliveryDate>		0,1
<GenericOrderInfo01>aaaaaaaaaaaaaaaaaaaa</GenericOrderInfo01>		0,1
: : : :		0,1
<GenericOrderInfo20>aaaaaaaaaaaaaaaaaaaa</GenericOrderInfo20>		0,1
<ApplicationName>aaaaaaaaaaaaaaaaaaaa</ApplicationName>		0,1
<ApplicationGUID>aaaaaaaaaaaaaaaaaaaa</ApplicationGUID>		0,1
<ApplicationVersion>aaaaaaaaaaaaaaaaaaaa</ApplicationVersion>		0,1

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

<Product GlobalID="aaa">		**n**
<ProductType>11</ProductType>		0,1
<TotalThickness>6666</TotalThickness>	Double	0,1
<DoubleWallsGap>777</DoubleWallsGap>	Double	0,1
<PieceCount>1111</PieceCount>	Int	0,1
<TurnWidth>3500</TurnWidth>	Double	0,1
<Comment>aaaaaaaaaa</Comment>		0,1
<RotationPosition>111</RotationPosition>	Double	0,1
<Slab GlobalID="aaa">		**n**
<SlabNo>111</SlabNo>		0,1
<PartType>11</PartType>		0,1
<ProductAddition>44</ProductAddition>		0,1
<ProductionWay>aa</ProductionWay>		0,1
<NumberOfMeansOfTransport>666</NumberOfMeansOfTransport>		0,1
<TransportSequence>777</TransportSequence>		0,1
<PileLevel>888</PileLevel>		0,1
<TypeOfUnloading>99</TypeOfUnloading>		0,1
<MeansOfTransport>00</MeansOfTransport>		0,1
<ExpositionClass>aaaaaaa</ExpositionClass>		0,1
<SlabArea>11.111</SlabArea>	Double	0,1
<SlabWeight>55555.5</SlabWeight>	Double	0,1
<ProductionThickness>2222</ProductionThickness>	Double	0,1
<MaxLength>11111</MaxLength>	Double	0,1
<MaxWidth>22222</MaxWidth>	Double	0,1
<IronProjectionLeft>±3333</IronProjectionLeft>	Double	0,1
<IronProjectionRight>±4444</IronProjectionRight>	Double	0,1
<IronProjectionBottom>±5555</IronProjectionBottom>	Double	0,1
<IronProjectionTop>±6666</IronProjectionTop>	Double	0,1
<X>111111</X>	Double	0,1
<Y>222222</Y>	Double	0,1
<Z>222222</Z>	Double	0,1
<OrderPosition>aaaaaaaa</OrderPosition>		0,1
<ProductGroup>bbbb</ProductGroup>		0,1
<SlabType>33</SlabType>		0,1
<ItemDesignation>c...c</ItemDesignation>		0,1
<ProjectCoordinates>111111 222222…</ProjectCoordinates>		0,1
<PositionInPileX>111111</PositionInPileX>	Double	0,1
<PositionInPileY>111111</PositionInPileY>	Double	0,1
<PositionInPileZ>111111</PositionInPileZ>	Double	0,1
<AngleInPile>111111</AngleInPile>	Double	0,1
<GenericInfo01>aaaaaaa…aaaaaa</GenericInfo01>		0,1
<GenericInfo02>aaaaaaa…aaaaaa</GenericInfo02>		0,1
<GenericInfo03>aaaaaaa…aaaaaa</GenericInfo03>		0,1
<GenericInfo04>aaaaaaa…aaaaaa</GenericInfo04>		0,1
<ReforcemInfo></ReforcemInfo>		0,1
<Outline Type="lot" GlobalID="aaa">		**n**
<X>22222</X>	Double	0,1
<Y>22222</Y>	Double	0,1
<Z>22222</Z>	Double	0,1

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

<Height>22222</Height>	Double	0,1
<Name>bbbb...bbbb</Name>		0,1
<GenericInfo01>bbbb...bbbb</GenericInfo01>		0,1
<GenericInfo02>bbbb...bbbb</GenericInfo02>		0,1
<MountingInstruction>2</MountingInstruction>		0,1
<MountPartType>33</MountPartType>		0,1
<MountPartArticle>aaaaaaaaaa</MountPartArticle>		0,1
<MountPartIronProjection>333</MountPartIronProjection>	Double	0,1
<MountPartDirection>±555</MountPartDirection>	Double	0,1
<MountPartLength>66666</MountPartLength>	Double	0,1
<MountPartWidth>77777</MountPartWidth>	Double	0,1
<ConcretingMode>aa</ConcretingMode>		0,1
<ConcreteQuality>aaaaaaaa</ConcreteQuality>		0,1
<UnitWeight>4.444</UnitWeight>	Double	0,1
<Volume>33.333</Volume>	Double	0,1
<Layer>aaa</Layer>		0,1
<Shape GlobalID="aaa">		**n**
<Cutout>false</Cutout>	Bool	0,1
<SVertex GlobalID="aaa">		**n**
<X>11111</X>	Double	0,1
<Y>22222</Y>	Double	0,1
<Bulge>33333</Bulge>	Double	0,1
<LineAttribute>aaaa</LineAttribute>		0,1
</SVertex>		
</Shape>		
</Outline>		
<Steel Type="mesh" GlobalID="aaa">		**n**
<X>22222</X>	Double	0,1
<Y>22222</Y>	Double	0,1
<Z>22222</Z>	Double	0,1
<ToTurn>true</ToTurn>	Bool	0,1
<StopOnTurningSide>true</StopOnTurningSide>	Bool	0,1
<Name>2</Name>		0,1
<GenericInfo01>aaa…aaa</GenericInfo01>		0,1
: : : :		0,1
<GenericInfo06>aaa…aaa</GenericInfo06>		0,1
<MeshType>2</MeshType>		0,1
<WeldingDensity>100</WeldingDensity>	Int	0,1
<BorderStrength>2</BorderStrength>	Int	0,1
<ProdRotX>±123</ProdRotX>	Double	0,1
<ProdRotY>±123</ProdRotY>	Double	0,1
<ProdRotZ>±123</ProdRotZ>	Double	0,1
<Layer>aaa</Layer>		0,1
<Bar GlobalID="aaa">		**n**
<ShapeMode>realistic</ShapeMode>		0,1
<ReinforcementType>3</ReinforcementType>		0,1
<SteelQuality>aaa</SteelQuality>		0,1
<PieceCount>55555</PieceCount>	Int	0,1
<Diameter>666</Diameter>	Double	0,1

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

	<X>±88888</X>	Double	0,1
	<Y>±99999</Y>	Double	0,1
	<Z>±77777</Z>	Double	0,1
	<RotZ>±123</RotZ>	Double	0,1
	<ArticleNo>bbbbbbbbbb</ArticleNo>		0,1
	<NoAutoProd>false</NoAutoProd>	Bool	0,1
	<ExtIronWeight>444.444</ExtIronWeight>	Double	0,1
	<Bin>123</Bin>		0,1
	<Pos>aaa</Pos>		0,1
	<Note>aaa</Note>		0,1
	<Machine>aaa</Machine>		0,1
	<BendingDevice>1</BendingDevice>		0,1
	<Spacer GlobalID="aaa">		**n**
	<Type>222</Type>	Int	0,1
	<Position>33333</Position>	Double	0,1
	</Spacer>		
	<WeldingPoint GlobalID="aaa">		**n**
	<WeldingOutput>77</WeldingOutput>	Double	0,1
	<Position>33333</Position>	Double	0,1
	<WeldingPointType>111</WeldingPointType>	Int	0,1
	<WeldingPrgNo>222</WeldingPrgNo>	Int	0,1
	</WeldingPoint>		
	****		**n**
	<RotX>±123</RotX>	Double	0,1
	<BendY>±123</BendY>	Double	0,1
	<L>33333</L>	Double	0,1
	<R>22</R>	Double	0,1
	****		
</Bar>			
<Girder GlobalID="aaa">			**n**
	<PieceCount>55555</PieceCount>	Int	0,1
	<X>±88888</X>	Double	0,1
	<Y>±99999</Y>	Double	0,1
	<Z>±77777</Z>	Double	0,1
	<GirderName>aaaaaaaaaa</GirderName>		0,1
	<Length>55555</Length>	Double	0,1
	<AngleToX>±999</AngleToX>	Double	0,1
	<NoAutoProd>true</NoAutoProd>	Bool	0,1
	<Height>222</Height>	Double	0,1
	<TopExcess>222</TopExcess>	Double	0,1
	<BottomExcess>222</BottomExcess>	Double	0,1
	<Weight>33.333</Weight>	Double	0,1
	<TopFlangeDiameter>44</TopFlangeDiameter>	Double	0,1
	<BottomFlangeDiameter>44</BottomFlangeDiameter>	Double	0,1
	<GirderType>2</GirderType>	Int	0,1
	<MountingType>1</MountingType>	Int	0,1
	<Machine>aaa</Machine>		0,1
	<Period>000</Period>	Double	0,1
	<PeriodOffset>111</PeriodOffset>	Double	0,1

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

<Width>80</Width>	Double	0,1
<AnchorBar GlobalID="aaa">		**n**
<Type>222</Type>	Int	0,1
<Length>111</Length>	Double	0,1
<Position>33333</Position>	Double	0,1
</AnchorBar>		
<GirderExt Type="SplicePos" GlobalID="aaa">		**n**
<Position>33333</Position>	Double	0,1
<Flags>0</Flags>	Int	0,1
<Val0>33333</Val0>	Double	0,1
<Val1>33333</Val1>	Double	0,1
<Val2>33333</Val2>	Double	0,1
<Val3>33333</Val3>	Double	0,1
</GirderExt>		
<Section GlobalID="aaa">		**n**
<L>195</L>	Double	0,1
<S>-100</S>	Double	0,1
</Section>		
</Girder>		
<Alloc Type="Bar" GlobalID="aaa">		**n**
<GuidingBar>2</GuidingBar>	Int	0,1
<Region>		**n**
<IntervalCount>5</IntervalCount>	Int	0,1
<Pitch>111</Pitch>	Double	0,1
<IncludeBegin>true</IncludeBegin>	Bool	0,1
<IncludeEnd>true</IncludeEnd>	Bool	0,1
<RefIndex>4</RefIndex>	Int	0,1
</Region>		
</Alloc>		
<SteelExt Type="Xyz" GlobalID="aaa">		**n**
<Info>aaaaaaaaaa</Info>		0,1
</SteelExt>		
</Steel>		
</Slab>		
</Product>		
</Order>		
<Feedback ItemType="Bar" GlobalID="aaa">		**n**
<MessageType>error</MessageType>		0,1
<Code>123ABC</Code>		0,1
<InfoValue>123XYZ</InfoValue>		0,1
<PieceCount>3</PieceCount>	Int	0,1
<MaterialType>16A</MaterialType>		0,1
<MaterialBatch>12345@AR177228C</MaterialBatch>		0,1
<MaterialWeight>12345</MaterialWeight>	Double	0,1
<ProdDate>2010-07-30T09:06:05+02:00</ProdDate>		0,1
<Machine>aaa</Machine>		0,1
<Description Culture="en" Text="aaaaaaaaaa"/>		n
</Feedback>		
</PXML_Document>		

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3 Detail-Spezifikationen

3.1 Global ID

Jede PXML-Tabelle kann ein Attribut namens **GlobalID** haben, das eine globale Identifikation des entsprechenden Items ermöglicht. "Global" ist hierbei als systemübergreifend zu verstehen, im Gegensatz also, zu systeminternen IDs, die nur innerhalb eines Subsystems bekannt sind.

Die *GlobalID* wird typischerweise vom datenerzeugenden System vergeben (also im CAD oder im Leitrechner), und von den Subsystemen unverändert übernommen, um dann in Rückmeldungen an das übergeordnete System verwendet zu werden.

Manche Systeme verwenden eine numerische *GlobalID*, andere verwenden einen String (z.B. einen GUID-String). Wie für fast alle PXML-Felder, gilt aber auch für die *GlobalID*, dass ihre Verwendung optional ist; so ist es durchaus möglich, ganz ohne *GlobalID* zu arbeiten, bzw. die *GlobalID* nur in einigen PXML-Tabellen zu verwenden.

3.1.1 Eindeutigkeit der GlobalID

Die *GlobalID* soll innerhalb eines Item-Typs eindeutig sein, d.h. beispielsweise, zwei unterschiedliche *Bar*-Items sollen unterschiedliche *GlobalID*s haben (das betrifft nicht nur 2 *Bar*-Items, die zum selben *Steel*-Block gehören, sondern auch *Bar*-Items aus unterschiedlichen *Steel*-Blocks sollen unterschiedliche *GlobalID* haben). Die *GlobalID*s verschiedener Item-Typen (z.B. *Bar* und *Girder*) können dagegen Überschneidungen haben.

Die Eindeutigkeit der *GlobalID* ist aber keine feste PXML-Vorschrift, sondern lediglich ein Erfordernis der Anwendung, das – je nach konkreter Applikation – unterschiedlich ausgeprägt sein kann. So reicht es für eine *PTS*-Anfrage beispielsweise aus, wenn die *GlobalID*s innerhalb eines Anfrage-Dokuments eindeutig sind. Für Produktionsrückmeldungen von Maschinen wird man dagegen typischerweise eine universellere Eindeutigkeit der *GlobalID*s benötigen. (*PTS*-Anfragen und Produktionsrückmeldungen werden weiter unten näher erläutert).

3.1.2 Sonderfall: GlobalID der DocInfo-Tabelle

Für die *DocInfo*-Tabelle hat die *GlobalID* eine etwas abgewandelte Bedeutung: sie identifiziert dort nicht den Tabelleneintrag (denn es gibt höchstens einen *DocInfo*-Tabelleneintrag), sondern das gesamte Dokument.

3.1.3 Automatisches generieren von GlobalIDs

Falls vom datenerzeugenden System keine *GlobalID* übergeben wird, kann das Subsystem diese Identifikations-Kennung eigenständig generieren, um dann gezielte Rückmeldungen machen zu können. Es soll dabei folgendes Muster verwendet werden:

ParentItemGlobalID.ItemIndex

Hierbei ist "ParentItemGlobalID" die *GlobalID* des unmittelbar darüber liegenden Parent-Items, und "ItemIndex" ist der nullbasierte Element-Index des betreffenden Items innerhalb seiner unmittelbaren Parent-Umgebung. Da *Order*-Items keinen Parent haben, wird dort nur der ItemIndex genommen (ohne Präfix).

<u>Beispiel 1</u>: Ein *Bar*-Item habe die *GlobalID* "ABC"; für die Segmente dieses Eisens können dann die *GlobalID*s wie folgt generiert werden:

"ABC.0"
"ABC.1"
"ABC.2"
usw.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Beispiel 2: Alle PXML-Items seien ohne *GlobalID*. für das fünfte Eisen im dritten *Order*-Block kann man dann folgende *GlobalID* automatisch generieren:
"2.0.0.0.4"
(*Order*-Index = 2, *Product*-Index = 0; *Slab*-Index = 0, *Steel*-Index = 0; *Bar*-Index = 4).

Eine eventuelle automatische Zuordnung von *GlobalID*s muss sofort nach dem Datenimport gemacht werden, also noch bevor über Filter-Funktionalitäten einzelne Items weggefiltert oder zusammengefasst werden.

Generell ist von der Verwendung automatisch generierter *GlobalID*s abzuraten, da sie als eher unzuverlässig anzusehen sind. Das gilt insbesondere dann, wenn in Datenexport-Modulen Filter- oder Gruppier-Funktionen vorhanden sind, die eine Änderung der Anzahl einiger Items verursachen könnten.

3.2 DocInfo

Der *DocInfo*-Block muss genau einmal vorhanden sein.

3.2.1 GlobalID

Der *GlobalID*-Eintrag ist optional und dient der Identifikation des gesamten Dokumentes, insbesondere in Hinblick auf *PTS*-Rückmeldungen.

3.2.2 Document Version

Die *MajorVersion* und die *MinorVersion* bezeichnen die Haupt-Version der zugrundeliegenden PXML-Spezifikation. Siehe hierzu Abschnitt 1.4.

3.2.3 Comment

Im *Comment*-Feld kann ein beliebiger Kommentar zum Dokument eingetragen werden. Die *Comment*-Angabe ist optional, sollte aber vor allem bei *PTS*-Rückmeldungen gesetzt werden, um den Zustand des sendenden Systems zu beschreiben (Identifikation des Systems, Version, Parameter-Version).

3.2.4 ConvertConventions

Wenn eine PXML-Datei aus irgendwelchen Gründen nicht PXML-konform ist, kann man das in den *ConvertConventions* vermerken. Beim Importieren einer solchen Datei können die Daten dann geeignet konvertiert werden, um einen PXML-konformen Datenbestand zu erhalten.

Falls mehrere Konventionen angegeben werden sollen, sind diese durch ein #-Zeichen zu trennen.

PXML-basierte Systeme arbeiten intern ohne Berücksichtigung der *ConvertConventions*. Die *ConvertConventions* werden sofort beim Importieren der Daten abgearbeitet und dann gelöscht.

Es gibt grundsätzlich keine *ConvertConventions*, die zwingend akzeptiert werden müssen. Folgende Konventionen werden aber häufig akzeptiert:

- **SegmentsHaveOuterLen**: Die Segment.L-Angaben geben die Außenlänge des Segmentes an, und nicht (wie in PXML gefordert) die Kernmaßlänge.

3.2.5 Mode

Über *Mode*-Einträge können Begleitinformationen angegeben werden, die typischerweise Verarbeitungs-Anweisungen für den Daten-Empfänger enthalten.

Jeder *Mode*-Eintrag besteht aus einem *ID*-Feld, das die Bedeutung des Eintrags festlegt, und einem *Val*-Feld, das den Wert enthält.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Folgende *Mode*-IDs sind im Standard definiert[3]:

- **ProdLayout**: Mögliche Werte sind "true" und "false".
  Diese *Mode*-Anweisung gibt an, ob die Elemente bereits für die Produktion angeordnet sind. Für Fertigteilumlaufanlagen bedeutet das konkret: bei *ProdLayout=true* handelt es sich um Daten die bereits von der Palettenbelegung auf Produktions-Paletten angeordnet wurden, und bei *ProdLayout=false* handelt es sich um CAD-Daten, deren Elementkoordinaten noch nicht auf Produktionspaletten ausgerichtet sind.
  Beachte: bei *ProdLayout=true* entspricht das PXML-Dokument (die Datei) genau einer Fertigungseinheit (also z.B. einer Umlaufpalette). D.h. die Daten werden in Fertigungseinheiten "portioniert", so dass das empfangende System weiß, welche Elemente zusammen in einer Fertigungseinheit liegen.
  Anmerkung zur PTS-Prüfung: Bei *ProdLayout=false* überprüft der PTS-Server die Elemente nur einzeln und ignoriert hierbei ihre Absolutposition (der PTS-Server geht davon aus, dass die Elemente später bei der Palettenbelegung noch ideal positioniert werden). Die Drehlage der Elemente wird vom PTS-Server aber als fest vorgegeben betrachtet, d.h. er versucht *nicht* eigenständig, eine optimale Drehlage anzunehmen[4].
- **RequestedCulture:** Mit einem oder mehreren solcher Einträge kann angegeben werden, welche Landessprachen interessieren. So kann beispielsweise einem PTS-Server mitgeteilt werden, welches die Sprachen sind, in denen die Meldungstexte geliefert werden sollen.
  Die Einträge sind mittels ISO 639 Language Codes anzugeben, optional erweitert durch die ISO 3166 Country Codes (Beispiele: "en" oder "en-US"). Wenn mehrere Sprachen angefordert werden, sind dementsprechend mehrere *Mode*-Einträge anzuführen.
- **EstimateProdTime:** Mögliche Werte sind "true" und "false".
  Diese *Mode*-Direktive weist einen PTS-Server an, eine Produktionszeitenberechnung vorzunehmen oder nicht.
  Diese Option soll helfen, die (eventuell rechenaufwändige) Produktionszeiten-Simulation nur dann auszuführen, wenn sie tatsächlich benötigt wird.

Beispiel für *Mode*-Angaben:

```
<?xml version="1.0" encoding="utf-8"?>
<PXML_Document xmlns="http://progress-m.com/ProgressXML/Version1">
  <DocInfo GlobalID="7C7E1FC5-0A46-48c5-A3B2-249D75B70BCF">
    <MajorVersion>1</MajorVersion>
    <MinorVersion>2</MinorVersion>
    <Mode>
      <ID>ProdLayout</ID>
      <Val>true</Val>
    </Mode>
    <Mode>
      <ID>RequestedCulture</ID>
      <Val>en</Val>
    </Mode>
    <Mode>
      <ID>RequestedCulture</ID>
      <Val>fr-BE</Val>
    </Mode>
```

[3] Darüber hinaus steht es jeder Applikation frei, für interne Zwecke eigene zusätzliche *Mode-IDs* zu definieren. Diese sollten dann mit "I_" beginnen, um Konflikte mit eventuellen späteren Erweiterungen des Standards auszuschließen.

[4] Es wird bewusst eine unterschiedliche Behandlung von Positionsoffset und Drehlage festgelegt: ersterer wird bei *ProdLayout=false* nicht geprüft, letztere dagegen immer. Es wäre praxisfern zu verlangen, dass ein CAD-System die Platten für die PTS-Anfrage produktionstechnisch optimal positioniert – denn das ist wesentliche Aufgabe des Palettenbelegungsprozesses. Andererseits würde es nicht funktionieren, den PTS-Server alle möglichen Drehlagen eigenständig durchprobieren zu lassen. Denn einerseits weiß der PTS-Server zu wenig über die Praktikabilität einer Drehung der Elemente, und andererseits kann es bei Verkettung von Systemen zu falschen PTS-Test-Ergebnissen kommen, wenn unterschiedliche Teilsysteme unterschiedliche Drehlagen annehmen. (Dieses Argument könnte im Prinzip zwar auch für die Positionierungsoffsets angeführt werden, doch hat es dort geringe Praxisrelevanz).

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

```
    <Mode>
      <ID>EstimateProdTime</ID>
      <Val>false</Val>
    </Mode>
  </DocInfo>
</PXML_Document>
```

3.3 Order

Der *Order*-Abschnitt kann beliebig oft vorkommen (auch 0 mal). Er enthält Fertigungseinheiten mit übereinstimmenden Auftrags-Daten.

Beachte: ein *Order*-Abschnitt muss *nicht alle* Fertigungseinheiten eines Auftrages enthalten. Er darf jedoch nicht Fertigungseinheiten zu verschiedenen Aufträgen enthalten.

3.3.1 Specific Order Info

OrderNo: Auftrags-Kennung (Beliebiger Text).

Component: Bauteil (Beliebiger Text).

Storey: Geschossbezeichnung (Beliebiger Text).

DrawingNo: Kennung der Zeichnung (Beliebiger Text).

DrawingDate: Datum an dem die Zeichnung erstellt oder geändert wurde (Beliebiger Text).

DrawingRevision: Revisionsnummer der Zeichnung (Beliebiger Text).

DrawingAuthor: Ersteller der Zeichnung (Beliebiger Text).

3.3.2 Delivery Date

Lieferdatum.

3.3.3 GenericOrderInfo

Die Einträge *GenericOrderInfo01* bis *GenericOrderInfo20* sind Informationseinträge mit frei wählbarer Bedeutung.

Beim UNICAM-Import und -Export werden die ersten 12 GenericOrderInfo-Einträge (sofern vorhanden) auf die 12 Zeilen von Bauvorhaben/Bauort/Bauherr abgebildet.

3.3.4 Comment

Beliebiger Kommentar zum Auftrag.

3.3.5 ImportSource, ImportSourceType

Falls der Auftrag aus einer anderen Datei importiert wurde, kann in *ImportSource* der Dateiname angegeben werden.

In *ImportSourceType* kann bei importierten Daten der Typ der Ursprungs-Datei angegeben werden (z.B. BVBS oder Unitechnik 5.2).

3.3.6 ApplicationName, ApplicationGUID, ApplicationVersion

Diese Angaben identifizieren die Applikation, mit der die Daten erstellt oder verändert wurden. Im Zweifelsfall bestimmt dies, wie die Daten zu interpretieren sind.

Typischerweise werden diese Angaben beim Laden aus Datei oder Zwischenablage ausgewertet, und dann (nach einer eventuellen Daten-Konvertierung) auf null gesetzt. Applikationsintern sollten diese Werte nicht verwendet werden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Der **ApplicationName** ist typischerweise der Titel der Applikation, ohne genaue Versions-Angabe und ohne Anspruch auf Eindeutigkeit.

Die **ApplicationGUID** sollte die Applikation eindeutig identifizieren, jedoch ebenfalls ohne genauere Versions-Angabe (eventuell könnte man für jede Hauptversion eine eigene GUID vergeben). Die GUID sollte das Format "59303B9F-B7E1-42bf-857A-9F6574A37433" haben.

Die **ApplicationVersion** soll eine möglichst genaue Versionsangabe beinhalten; typischerweise ist dies eine Zeichenkette der Form "4.21.2471.40371".

3.4 Product (Element)

Bei Doppelwänden entspricht das **Product** der vollständigen Wand; es enthält beide Wandschalen.

Die Informationen der UNICAM-SLABDATE sind teilweise im Product-Abschnitt und teilweise im Slab-Abschnitt untergebracht.

3.4.1 ProductType

Entspricht in etwa der Produkt-Spezifikation des UNICAM-HEADER.

Man kann beliebige Typangaben verwenden. Folgende Typen sind fest definiert:

- **00:** element and roof slab
- **DW:** double wall
- **03:** prestressed slab
- **04:** isolating slab
- **05:** facade element
- **06:** solid floor
- **07:** silo slab
- **08:** constructional part
- **09:** solid wall
- **10:** hollow floor
- **11:** sandwich element
- **16:** brick floor
- **19:** brick wall
- **NW:** zero wall
- **TW:** thermo wall
- **36:** light-weight concrete full-thickness floor
- **39:** light-weight concrete solid wall
- **GML:** Generic Multi Layer element

Beachte: falls in einem Order-Block verschiedene ProductType-Werte vorkommen, wird der Order-Block beim UNICAM-Export auf mehrere HEADER-Blöcke aufgeteilt (In UNICAM steht der Product-Type im HEADER-Block).

Beachte: für die Doppelwand werden hier *nicht* die Typen 01 und 02 verwendet, sondern der Typ **DW**; Im PartType des Slabs wird dann zwischen Wandhälfte 1 und Wandhälfte 2 unterschieden.

3.4.2 TotalThickness, DoubleWallsGap

Sind identisch mit den entsprechenden UNICAM-SLABDATE-Feldern. Beide Werte in mm.

Die DoubleWallsGap wird natürlich nur bei Doppelwänden benötigt.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.4.3 PieceCount

Soll-Stückzahl.

3.4.4 TurnWidth, Datenübergabe für Doppelwände

Für Doppelwände (und andere 2-Platten-Elemente) gilt folgende Festlegung:

Die Einzelplatten werden so dargestellt, wie sie vor dem Einwenden auf den Einzelpaletten liegen; die erste (= die einzuwendende) Wandhälfte ist also gegenüber dem fertigen Produkt gedreht darzustellen.

Die Darstellung der Einzelplatten entspricht also der produktionstechnischen Situation, und nicht jener des fertigen Produktes.Die Doppelwand wird also in *aufgeklappter* Form beschrieben.

Diese Festlegung hat aber auch einen Nachteil: um zu verstehen, wie das fertige Element aussieht, reicht es nicht aus, die Einzelplatten zu betrachten, sondern man muss zusätzlich wissen, wie die beiden Wandhälften zusammengeführt werden.

Innerhalb einer Produktionsanlage ist diese Information üblicherweise implizit gegeben: es gibt eine feste Palettenbreite und dadurch eine feste Kippachse, die auf halber Palettenbreite liegt. Diese Kippachse (= Achse des Kipptisches) ist es also, die bestimmt, wie die beiden Wandhälften zusammengeführt werden. Wenn jedoch CAD-Daten vorliegen, die nicht für eine feste Palettenbreite angefertigt wurden, hat man ein Problem; das zu produzierende Element ist in seiner Geometrie dann nicht vollständig bestimmt.

Diese Unbestimmtheit kann durch explizite Angabe der **TurnWidth** (=Kippbreite) behoben werden. Diese Größe beschreibt die angenommene Palettenbreite. Die beiden Wandhälften sind zueinander so verschoben, dass durch Einwenden an der Achse *Y=TurnWidth*/2 das gewünschte Doppelwand-Produkt entsteht. Wenn die Anlage eine andere Palettenbreite hat, als die angenommene, so müssen die Koordinaten bei der Palettenbelegung entsprechend angepasst werden.

Hinweis zum Import von UNICAM-Daten aus CAD-Systemen:

Wenn UNICAM-Daten von einem CAD-System übernommen werden, muss man eine Annahme über die verwendeten Doppelwand-Koordinaten treffen. Je nach CAD-System, und je nach CAD-System-Einstellung, kann es hier ganz unterschiedliche Festlegungen geben, die aber aus dem UNICAM-File selbst nicht ersichtlich sind. Die häufigsten Konventionen sind:

a) Die Doppelwandhälften werden in **aufgeklappter Darstellung** übertragen, also so wie im PXML. Da es in UNICAM kein *TurnWidth*-Feld gibt, muss man hierbei wissen, welche Palettenbreite im CAD angenommen wird; andernfalls sind die Daten nicht korrekt interpretierbar.

b) Die zweite Wandhälfte wird so wie im Fall a) übergeben, und damit gleich wie im PXML. Die erste Wandhälfte wird dagegen in **pseudo-gewendeter Form** übergeben: die Daten entsprechen teilweise der eingewendeten Platte; die Z-Koordinaten werden aber nicht transformiert, Form und Ausrichtung der Eisen und Gitterträger werden ebenfalls nicht transformiert. Für die Eisen und Gitterträger hat man also genau genommen keine Drehung, sondern eine Y-Verschiebung um *Palettenbreite-2*Ym*, wobei *Ym* die Y-Mitte des Eisens oder des Gitterträgers ist (Wer denkt sich bloß eine solche Transformation aus?!). Die erste Wandhälfte wird hier also geometrisch *falsch* dargestellt, da die Daten weder der ungewendeten noch der gewendeten Platte entsprechen; im Gegensatz zur Variante a) hat man aber den Vorteil, dass die Lage der Wandhälften zueinander eindeutig bestimmt ist (Diesen Vorteil könnte man aber auch erzielen, wenn man die erste Wandhälfte korrekt gewendet darstellt; es ist also kaum nachvollziehbar, warum gerade das so verwirrende *pseudo-wenden* so oft verwendet wird.).

c) Die **invers pseudo-gewendete Form** ist ähnlich dem Fall b), wobei aber die Doppelwand von unten betrachtet wird, d.h. mit Blick auf die Außensete der zweiten Wandhälfte. Beim Importieren dieser Daten muss man die gesamte Wand im 180° um die Z-Achse drehen, und dann noch die zweite Wandhälfte pseudo-wenden.

3.4.5 Comment

Beliebiger Kommentar zum Produkt.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.4.6 RotationPosition

Ein Winkel in Grad, der angibt, wie das Element gegenüber der ursprünglichen CAD-Zeichnung gedreht wurde (Die Drehung erfolgt typischerweise während des Palettenbelegungsprozesses aus produktionstechnischen Gründen).

Die Drehung des Elementes erfolgt um die Z-Achse im positiven Umlaufsinn (also gegen Uhrzeigersinn). Da die erste Wandhälfte von Doppelwänden vor dem Einwenden in geklappter Form vorliegt, muss der Winkel dort in negativem Umlaufsinn gerechnet werden.

3.5 Slab (Elementteil, Platte)

3.5.1 SlabNo

Dient zur Identifikation der Platte. Entspricht in UNICAM 5.2 der *Element Part Number* und in späteren Versionen dem *Element Part Name*.

3.5.2 PartType

Man kann beliebige Typangaben verwenden. Die Bedeutung des PartType hängt vom übergeordneten ProductType ab.

Für den Produkt-Typ DW sind folgende PartType-Werte fest definiert:

- 01: double wall 1st stage
- 02: double wall 2nd stage

3.5.3 Various Slab Information

Sammlung diverser Slab-Informationen. Entspricht im Einzelnen den entsprechenden Feldern im Info-Block der UNICAM-SLABDATE.

Einige dieser Informationen sind redundant, da sie sich aus anderen Größen berechnen lassen; die Werte sind aber dennoch explizit aufgelistet, um einer CAD-Applikation die Möglichkeit zu geben, diese Werte frei zu setzen.

Im Gegensatz zu UNICAM gibt es in PXML auch eine Z-Koordinate für die Platte.

Die **Transport- und Stapeleigenschaften** sind der Slab (=Elementteil) zugeordnet, obwohl es normalerweise Eigenschaften des Produkts (=Element) sind. Diese Wahl verbessert die Kompatibilität mit UNICAM und erhöht die Flexibilität, da es auch denkbar wäre Elementteile von einer Produktionsstelle zu einer anderen zu transportieren.

3.5.4 Slab-Type

Bei der Konvertierung von und zu UNICAM wird der SlabType auf den UNICAM-Elementtyp abgebildet. Genaugenommen ist in PXML jedoch der SlabType eine Eigenschaft des Element**teils**.

Wenn möglich, sollte dieser Wert überhaupt nicht verwendet werden.

3.5.5 ProjectCoordinates

Enthält die 9 Projektkoordinaten (durch Leerzechen getrennt), wie sie in der UNICAM-Spezifikation angeführt sind.

3.5.6 Coordinates of Element in Pile

Koordinaten des Elementes im Stapel. Entspricht der UNICAM-Spezifikation (ab Version 6.0).

Im Gegensatz zu UNICAM ist der Winkel nicht auf [0, 359°] beschränkt, und es wird sogar empfohlen den Bereich]-180°, 180°] zu verwenden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.5.7 ReforcemInfo

Nur für Kompatibilität mit UNICAM; Enthält die Angaben des UNICAM-REFORCEM Info-Blocks.

3.5.8 Generic Slab Info

Frei definierbare Zusatz-Informationen.

3.5.9 Multi-Layer Elements

In einer Slab können mehrere Betonschichten angegeben werden (Lots).Ein Elementteil könnte aber auch aus mehreren Schichten bestehen, von denen jeder ihre eigene Bewehrung und eigene Einbauteile hat (Entsprechend den UNICAM-LAYERS).

In PXML kann man das realisieren, indem in den *Outline*- und in den *Steel*-Blöcken die Layer-Werte gesetzt werden.

3.6 Outline

Unter **Outline** wird hier eine allgemeine geometrische Umrandung verstanden. Das Attribut **Type** bestimmt die Art der Umrandung:

- **lot**
- **mountpart**

Einige Tags des Outline-Abschnittes sind nur für bestimmte Outline-Typen vorgesehen, und werden bei den anderen Outline-Typen ignoriert.

Die **lot**-Outlines beschreiben ein **Beton-Parzelle** mit Konturen, Aussparungen und Beton-Daten.

Die **mountpart**-Outlines beschreiben Einbauteilen.

3.6.1 X/Y/Z

Offset aller Daten bezogen auf den Slab-Nullpunkt.

Bei Einbauteilen ist die Z-Koordinate die Einbauhöhe.

In UNICAMgibt es (ab Version 6.0)für Einbauteile die Angabe einer **Einbau-Position**, bezogen auf den Slab-Nullpunkt. Dieser Wert ist bis zu einem gewissen Grad redundant, da die Einbauposition aus der Position und Geometrie des Einbauteils folgt. (Das gilt zwar nur für bekannte Einbauteile, doch können auch nur solche automatisch verlegt werden). Aus diesem Grund gibt es in PXML diese Positions-Angabe nicht.

Man kann aber den X/Y-Offset des Einbauteils so setzen, dass er mit der Einbau-Position übereinstimmt (die Vertex-Koordinaten müssen entsprechend kompensiert werden). Es wird empfohlen, dies beim Datenaustausch mit UNICAM so zu handhaben.

3.6.2 Height

Höhe in mm (Dicke der Betonschicht, Höhe des Einbauteils).

3.6.3 Name

Bezeichnung. In UNICAM 5.2 gibt es die Bezeichnung nur für Mountparts; ab UNICAM 6.0 gibt es sie auch für Cutouts. In PXML gibt es sie auch für Contours, wobei aber empfohlen wird, dies für Contours nicht zu verwenden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.6.4 GenericInfo

Frei verwendbare Informationszeilen.

3.6.5 MountingInstruction (only for mountparts)

Einbau-Anweisung. In Anlehnung an das UNICAM-Einbau-Kennzeichen gilt:

- 0 = Teil wird eingebaut
- 1 = Teil wird nur gezeichnet
- 2 = Teil wird nur eingebaut
- 3 = Teil wird weder gezeichnet noch eingebaut
- 4 = Teil wird in Bewehrung eingebaut
- 5 = Teil wird automatisch eingebaut

3.6.6 MountPartType, MountPartArticle(only for mountparts)

Entspricht den entsprechenden UNICAM-Definitionen.

3.6.7 MountPart Properties (only for mountparts)

Entspricht den entsprechenden UNICAM-Definitionen:

- **MountPartIronProjection** = Eisenüberstand in mm. Es handelt sich hierbei um Eisen, die aus dem Einbauteil herausragen.
- **MountPartDirection** = Richtung des Eisenüberstandes oder Ausrichtung des Einbauteils. Die Winkel-Angabe ist im Bereich]-180°, 180°]. Beim Export nach UNICAM wird dies auf [0°, 360°] konvertiert.
- **MountPartLength/MountPartWidth** = Länge/Breite in mm.
  Die Wertesind optional, und werden im allgemeinen gesetzt, um das automatische Zuschneiden des Teils zu ermöglichen. In diesem Sinne handelt es sich um Produktionsmaße des Einbauteils.
  Benötigt werden die Werte grundsätzlich nur für Einbauteile, deren Größe variabel ist, und nicht durch die Artikelnummer vollständig bestimmt ist.
  Die genaue Bedeutung der beiden Maße hängt vom Typ des Einbauteils ab. Es kann Typen geben, die nur eines der beiden Maße benötigen, und es ist auch möglich, den beiden Werten eine andere Bedeutung, als jene der Länge und Breite, zuzuordnen.

3.6.8 Concrete Properties (only for lots)

- **ConcretingMode** = Betonierkennzeichen.
- **ConcreteQuality** = Betongüte (z.B. B25).
- **UnitWeight** =Rohdichte in kg/dm³.
- **Volume** =Soll-Betonvolumenm³.

3.6.9 Layer

Definiert bei mehr-schichtigen Elementen die Layer-Zugehörigkeit.

3.6.10 Shape, SVertex

Ein **Shape** enthält eine Folge von Punkten (**SVertex**), die ein Polygon bilden. Das Polygon wird geschlossen, indem der letzte Punkt wieder mit dem ersten Punkt verbunden wird (*ohne* dass der erste Punkt deswegen zweimal angeführt würde).

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Cutout:

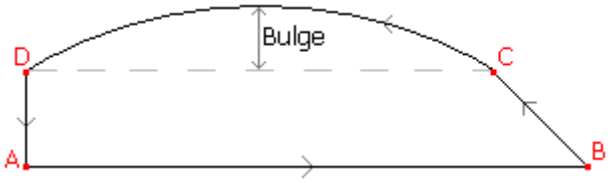
Wenn dieses Feld auf *true* steht, beschreibt das *Shape*-Objekt eine Aussparung.

Das *Cutout*-Feld wird nur für *lot*-Outlines benötigt.

SVertex:

Ein SVertex beschreibt einen Eckpunkt des Polygons (Punkt in der Ebene).

Jedem *SVertex* kann eine Ausbuchtungs-Höhe **Bulge** zugeordnet werden: falls diese ungleich 0 ist, ist die Verbindung des Vertex zu seinem Folge-Vertex ein Bogen mit der angegebenen Höhe (in mm).



Die Abbildung zeigt ein *Shape* mit 4 *Vertices*. A, B und D habe *Bulge=0*, während für C der Wert von *Bulge* positiv ist; bei einem negativen Wert von *Bulge* wäre der Bogen nach innen gewölbt, d.h. das Vorzeichen von *Bulge* gibt den Bogen-Umlaufsinn an.

Sonderfall von nur einem SVertex:

Falls der *Shape* aus einem einzigen *Vertex* besteht, wird dies als *Kreis* interpretiert. Der *Vertex* gibt den Kreismittelpunkt an und der Betrag von *Bulge* entspricht dem Kreisdurchmesser (dieser kann auch 0 sein); das Vorzeichen von *Bulge* gibt den Umlaufsinn an.

Line-Attribute:

Für jeden *SVertex* kann ein **LineAttribute** angegeben werden, dem man verschiedenste Applikationsspezifische Bedeutungen zuordnen kann. So könnte beispielsweise über das *LineAttribute* einer Element-Kontur die Information über den Schalungs-Typ angegeben werden.

Meistenswird für das *LineAttribut* eine 4-stellige Hex-Zahl angegeben, die als Bit-Feld interpretiert wird. Die einzelnen Bits haben dann folgende Bedeutung:

Bit 00 [0001]: Keine Fase unten.
Bit 01 [0002]: Sonderschalung.
Bit 02 [0004]: Vergussfuge.
Bit 03 [0008]: Keine Fase oben.
Bit 04 [0010]: Rundung unten.
Bit 05 [0020]: Feder (Schalung mit Nut).
Bit 06 [0040]: Nut (Schalung mit Feder).
Bit 07 [0080]: Rundung oben.
Bit 08 [0100]: Saubere Kante.
Bit 09 [0200]:
Bit 10 [0400]:
Bit 11 [0800]: Stützschalung (Schalung stützt ein Einbauteil).
Bit 12 [1000]: Fensterschalung.
Bit 13 [2000]: Konturschalung.
Bit 14 [4000]: Schalung nicht fixieren.
Bit 15 [8000]:

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Offener Polygonzug:

Nach obiger Definition bilden die *Vertices* eines *Shapes* immereinen geschlossenen Polygonzug. Ein offener Polygonzug kann nur dadurch realisiert werden, dass man vom letzten Punkt wieder den gesamten Weg bis zum ersten Punkt zurückläuft. Insbesondere wird ein *offenerKreisbogen* durch ein Shape mit 2 *Vertices A* und *B* realisiert, die $A_{Bulge}=-B_{Bulge}$ erfüllen. Falls die beiden *Bulge*-Werte zudem gleich *0* sind, hat man den Spezialfall einer einfachen *Strecke*.

Beachte: offene Polygonzüge (vereinzelte Strecken oder Bögen) haben immer einen Flächeninhalt von 0. Sie sind daher nur beim *mountpart*-Outlines sinnvoll;bei *lot*-Outlines würden offene Polygonzüge als Konturen und Aussparungen mit Fläche 0 interpretiert, was offenbar sinnlos ist. Eine Kontur oder Aussparung darf daher nie als Folge von Einzelstrecken dargestellt werden, sondern muss stets über Polygone dargestellt werden.

3.7 Steel

Der *Steel*-Abschnitt ist eine Gruppierung von Rundeisen und Gitterträgern.

Type-Attribut

Es gibt folgende Typen:

- **none:** Lose Eisen und Gitterträger.
- **mesh:** Stahl-Matte (eventuell auch mit Gitterträgern). Auch Körbe, die aus einer Stahlmatte produziert werden.
- **cage:** Bewehrungskorb.
  Typischerweise werden Bügel und Längseisen angegeben (vorzugsweise die Längseisen mit RotZ=0 und die Bügel mit RotZ=90).
- **extiron:** Getrennt zugeführte Bewehrung, die nicht im Detail spezifiziert wird. Das können neben losen Eisen auch komplexere Einheiten wie Matten oder Körbe sein. Der Gitterträger-Teil wird bei Extiron-Steel-Blöcken nicht verwendet.

3.7.1 X/Y/Z

Offset aller Daten bezogen auf den Slab-Offset.

3.7.2 ToTurn (Nur für Matten)

Mögliche werte: **true**, **false**.

Gibt an ob die Matte gewendet ausgeliefert werden soll (gewendet wird entlang der Längs-Achse).

3.7.3 StopOnTurningSide (Nur für Matten)

Mögliche Werte: **true**, **false**.

Gibt an ob die Anschlagsseite der Matte zugleich auch die Wendeseite ist.

3.7.4 Name

Bezeichnung des Steel-Blocks (z.B. Mattenbezeichnung).

3.7.5 MeshType

Matten-Typ:

- 0:Standardmatte.
- 1: Gebogene Matte 2D; Matte ist einer Balkenbiegemaschine zuzuführen.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

- 2:Bewehrungsmodul.
- 3:Gebogene Matte 3D; Matte ist einer Einzelkopfbiegemaschine zuzuführen.
- 4:Deckelmatte (z.B. Deckel eines Korbes, oder einer Massivwand-Bewehrung).
- 5:Deckelmatte 2D; wie Typ '1', wird aber zusammen mit Typ '4' ausgeliefert.
- 6: Deckelmatte 3D; wie Typ '3', wird aber zusammen mit Typ '4' ausgeliefert.
- 7: (Frei für Erweiterungen; noch nicht festgelegt; sollte derzeit nicht verwendet werden).
- 8: Lose Matte: wird nicht von der Stahlmaschine produziert, sondern manuell hinzugefügt (üblicherweise wird eine solche Matte einem Matten-Lager entnommen).
- 9: Applikationsspezifischer Typ.

Der MeshType kann neben der oben genannten Typ-Kennung noch weitere typspezifische Detail-Informationen beinhalten; Folgende Verwendung wird empfohlen:

- 8#StandardSheet:R188A
  Lose Matte, zu realisieren mittels Lagermatte vom Typ R188A.
- 9#ProgressInfo8080:123#ProgressInfo8090:abc
  Hier wird ein applikationsspezifischer Typ angegeben, der über 2 applikationsspezifische Parameter näher beschrieben wird (die Parameter haben in diesem Beispiel die Bezeichnung "ProgressInfo8080" und "ProgressInfo8090"; diesen Parametern werden die Werte "123" bzw. "abc" zugewiesen).

Es können also mehrere Parameter angegeben werden, die durch #-Zeichen getrennt sind; Parameter-Name und der Parameter-Wert sind durch einen Doppelpunkt getrennt.

3.7.6 WeldingDensity (Nur für Matten)

Schweissdichte in % (Integer-Wert).

Anzugeben sind Werte zwischen 0 und 100. Zusätzlich können höherwertige Stellen für die Kodierung von Zusatz-Informationen verwendet werden:

$a = WeldingDensity \textbf{ mod } 1000$

$b = WeldingDensity \textbf{ div } 1000.$

Der Wert von *a* bestimmt die Schweißdichte; der Wert von *b* steht für zusätzliche anlagenspezifische Informationen zur Verfügung.

Falls *WeldingDensity* einen Wert von *0* oder *DBNull* hat, so steht dies für "Default"; je nach Anlage kann dies 0%, 100% oder irgend ein anderer Wert sein.

Beachte: die angegebene Schweißdichte ist als *innere* Schweißdichte zu verstehen; am Matten-Rand werden *zusätzliche* Schweißpunkte eingefügt (siehe *BorderStrenght*).

3.7.7 BorderStrength

Verstärkung des Mattenrandes. Ein Wert von *0* oder *DBNull* steht für den anlagenspezifischen Default-Wert. Ein Wert von 1 bedeutet, dass die äußerste Reihe zu 100% geschweißt wird; ein Wert von 2 bedeutet, dass die 2 äußersten Reihen zu 100% geschweißt werden.

3.7.8 Generic Steel Info

Frei verwendbare Info-Zeilen.

3.7.9 ProdRotX/Y/Z

Die **ProdRot**-Felder beschreiben eine Drehung des Bewehrungs-Blocks (typischerweise eines Korbes), der aber nicht das fertige Produkt betrifft, sondern nur als Empfehlung für die Bewehrungsproduktionsmaschine zu verstehen ist. Der Bewehrungskorb ist also *nicht* in gedrehter Form

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

in das Betonelement einzubauen, sondern nur vorübergehend zu drehen, um leichter gefertigt werden zu können. Der Bewehrungsproduktionsmaschine steht es natürlich frei, diese Empfehlung zu beachten, oder eigenständig zu entscheiden wie herum der Korb gedreht werden soll, um gut gefertigt werden zu können.

Die 3 ProdRot-Winkel sind in DEG-Einheit anzugeben, und beschreiben die Drehung wie folgt:

- Zunächst wird der gesamte Korb um die X-Achse gedreht.
- Dann wird um die Y-Achse des gedrehten Systems gedreht.
- Dann wird um die Z-Achse des gedrehten Systems gedreht.

Beachte: üblicherweise betrifft die Drehung nur die Rundstahl-Eisen, nicht aber die Gitterträger.

3.7.10 Layer

Definiert bei mehr-schichtigen Elementen die Layer-Zugehörigkeit.

3.8 Bar

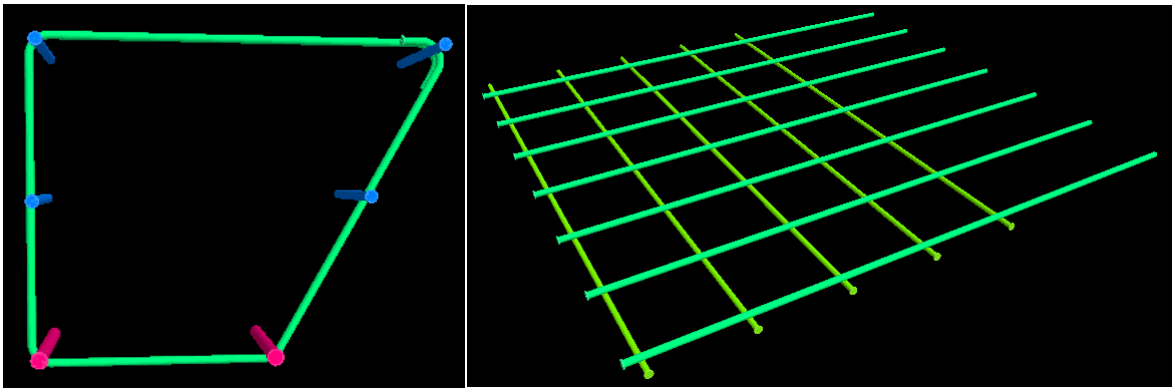
Ein *Bar*-Eintrag entspricht einem Rundstahl-Element, also einem geraden oder gebogenen Bewehrungseisen.

3.8.1 ShapeMode

Über den *ShapeMode* hat man die Möglichkeit eine Darstellungsart für die Geometrie des Eisens zu wählen. Für gerade Eisen spielt diese Option keine große Rolle; für komplexe Biegeformen hat man durch geeignete Wahl des *ShapeMode* aber die Möglichkeit, die Daten-Darstellung möglichst stark an die interne Darstellung im CAD anzulehnen. Der Aufwand für die Implementierung des Datenexports vom CAD kann dadurch entscheidend verringert werden.

3.8.1.1 ShapeMode "realistic"

Alle Eisen werden mit korrekten Biegeradien und korrekten Raum-Koordinaten angegeben (die Biegeradien können auch über die *BendingDevice* gegeben werden). Die Relativ-Abstände der Eisen sind ebenfalls korrekt zu berücksichtigen. Diese Darstellung ist uneingeschränkt verwendbar, da sie das zu erzeugende Produkt direkt wiedergibt, uns so keine Interpretationsspielräume lässt.



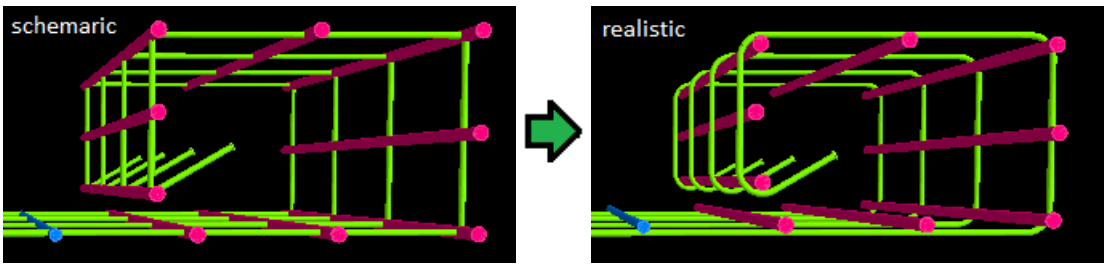
Die Bewehrungsart der Eisen hat in dieser Darstellung keine geometrische Bedeutung, da bereits ohne Kenntnis der Bewehrungsartdie Geometrie eindeutig definiert ist.

Wegen ihrer Universalität und Eindeutigkeit ist das die empfohlene Darstellungsart.

3.8.1.2 ShapeMode "schematic"

Der *schematic*-Mode ist eine vereinfachte Darstellung in der die Eisen mit Biegeradius 0 gezeichnet werden:

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.



Bei Biegewinkeln bis zu 90° ist die Biegeform einfach nur "eckiger" als in der Realität. Für Biegewinkel über 90° wird aber die äußere Form verfälscht dargestellt.

Die räumliche Absolut-Lage der Eisen ist hierbei wie folgt zu setzen:

- Unabhängige Eisen (*Master*-Eisen): die Eisen-Koordinaten werden so gesetzt, dassdas L0-Segment (=Hauptsegment) die reale Lage einnimmt.
- Abhängige Eisen (*Slave*-Eisen): die *Slave*-Eisen werden entlang der *Master*-Eisen aufgefädelt, typischerweise so, dass sich die Eisen im Kern schneiden.

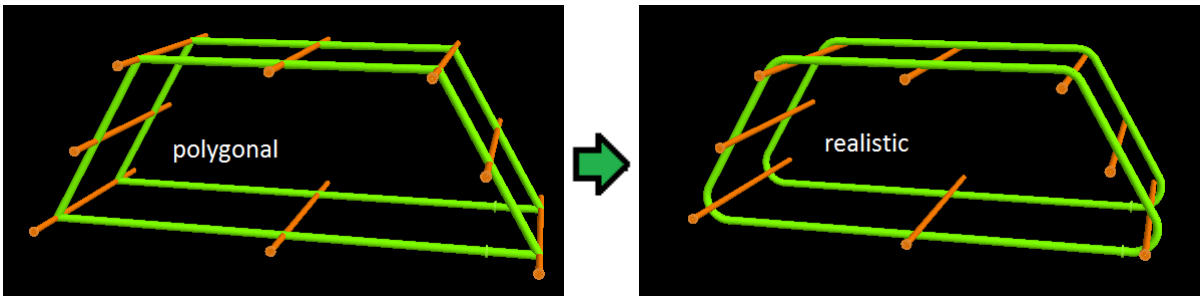
Master/*Slave*-Beziehungen können entweder explizit durch Leiteisen-Staffelmuster definiert werden[5], oder implizit über die Bewehrungsart: zwei Eisen mit Bewehrungsart 1 und 2 stehen in einer *Master*/*Slave*-Beziehung, wenn sie sich im Eisenkern schneiden (Das Bewehrungsart-1-Eisen ist hierbei der *Master* und das Bewehrungsart-2-Eisen ist der *Slave*)[6].

Wenn man die Eisen so angibt, dass sie sich im Kern schneiden, benötigt man eine formale Zusatz-Regel, die festlegt, wie die Eisen-Lagen real zueinander versetzt sind. Für *Master*/*Slave*-Eisen die sich im Kern schneiden, soll daher folgendes gelten:

Bei geraden *Master*-Eisen liegen die *Slave*-Eisen oben (also Bewehrungsart 1 unten, Bewehrungsart 2 oben). Bei gebogenen *Master*-Eisen liegen die *Slave*-Eisen *innen*[7].

3.8.1.3 ShapeMode "polygonal"

Die *polygonal*-Darstellungist der *schematic*-Darstellung ähnlich, und die meisten oben beschriebenen Regeln können unverändert übernommen werden. Im Gegensatz zur *schematic*-Darstellung werden aber nicht die üblichen Eisenlängen angegeben, sondern die Kantenlängen des Polygons, das die reale Biegeform abdeckt.



Für Biegungen bis zu 90° stimmen die *schematic* und die *polygonal*-Darstellungen überein. Die Darstellungen unterscheiden sich aber bei spitzen Biegungen: die *polygonal*-Darstellung bleibt auch bei spitzen Biegewinkeln formgetreu zum realen Eisen, während die *schematic*-Darstellung die äußeren Proportionen verändert[8].

---

[5] Leiteisen-Staffelmuster: siehe Abschnitt 3.10.1.

[6] Analog dazu kann ein Bewehrungsart-4-Eisen ein *Slave* zu einem Bewehrungsart-2-Eisen sein. Diese mehrstufige Abhängigkeit wird aber nur bei sehr komplizierten Körben verwendet. (Für eine Definition der Bewehrungsarten siehe Abschnitt 3.8.2).

[7] In praktischen Fällen ist bei Biegeformen klar ersichtlich, wo *innen* und *außen* ist. Eine allgemeine mathematisch präzise Definition von *außen* und *innen* kann wie folgt festgelegt werden:
Seien $\vec{v_i}$ die Vektoren der Eisensegmente, und $\vec{a} := \sum_{i=0}^{n-2} \vec{v_i} \times \overrightarrow{v_{i+1}}$. Dann zeigt $\vec{a} \times \vec{v_i}$ nach *innen*.

[8] In obigem Beispiel hat der Korb in der *polygonal*- und in der *realistic*-Darstellung die selbe Höhe; in der *schematic*-Darstellung hätte er eine geringere Höhe.

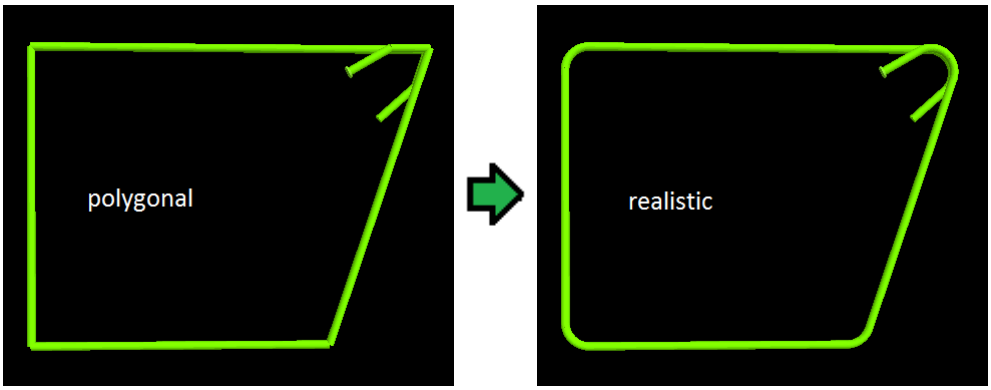
Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Für "Abrundungsbiegungen" an den Eisenenden, ist die *polygonal*-Darstellung aber oft unnatürlich und für 180°-Biegungen ist sie sogar unmöglich (die Polygonschenkel wären dann unendlich lang). In diesem Fall kann man einen spitzen Biegewinkel α in zwei Winkel α_1 und α_2 aufteilen, und zwischen diesen Biegungen eine Zusatzkante einfügen, die folgende Länge hat:

$$L_k = R \cdot \left(\tan\frac{\alpha_1}{2} + \tan\frac{\alpha_2}{2}\right), \qquad \alpha_1 + \alpha_2 = \alpha$$

Hierbei ist R der Biegeradius im Eisenkern.

Typischerweise ist α_1 durch eine äußere Formgebung des Polygons gegeben, und α_2 vervollständigt die Biegung, um auf den gewünschten Gesamtwinkel α zu kommen. Die dazwischen künstlich einzuführende Polygonkante L_k kann wie oben beschrieben ermittelt werden. Nach der Umrechnung in die *realistic* Darstellung reduziert sich die Zusatzkante auf ein Segment der Länge 0, das dann sinnvollerweise weggelassen wird – die beiden Biegewinkel α_1 und α_2 verschmelzen dann wieder zu einem Gesamtwinkel α.



Unter Einbeziehung dieser Winkelaufteilung wird die *polygonal*-Darstellung zu einer sehr universellen Form, die der *schematic*-Darstellung überlegen ist[9].

Die Abhängigkeit des L_k vom R bringt eine gewisse Redundanz in die Daten, da L_k aus R in eindeutiger Weise hervorgeht. Tatsächlich kann man L_k auch kleiner wählen, oder sogar auf 0 setzen, denn das datenempfangende System weiß, dass jeder Polygonschenkel mindestens die oben angeführte Länge haben muss. Man hat dann aber, so wie in der *schematic*-Darstellung, für manche Eisensegmente eine Abweichung des Polygons vom realen Eisen. Das betrifft typischerweise aber nur Endabrundungen, so dass die Gesamtform des Eisens unverfälscht wiedergegeben wird.

3.8.1.4 Automatische Ermittlung des ShapeMode und gemischte Darstellungen

Oft wird kein *ShapeMode* angegeben, insbesondere dann, wenn die Daten aus UNICAM- oder BVBS-Dateien importiert werden. Da unterschiedliche CAD-Systeme mit unterschiedlichen Darstellungsformen arbeiten, müssen Daten ohne *ShapeMode*-Angabe als unvollständig angesehen werden[10].

In UNICAM-Dateien wird der *ShapeMode* manchmal (in Erweiterung der offiziellen Definition) in das Reserve-Feld RODSTOCK Zeile 5, Spalten 9-11 abgebildet:

- 000 = *undefined*
- 001 = *schematic*

[9] Die genannte Winkelaufteilung kann in der *schematic*-Darstellung ebenso durchgeführt werden; man kann dadurch vermeiden, dass Biegewinkel über 90° angegeben werden müssen, und hätte dann vollkommene Übereinstimmung zwischen der *schematic*- und der der *polygonal*-Darstellung. Die Überlegenheit der *polygonal*-Darstellung entsteht aber durch die Möglichkeit spitze Biegewinkel durch spitze Polygon-Winkel formgetreu anzugeben – das ist in der *schematic*-Darstellung nicht möglich.

[10] Es gibt sinnvolle Ansätze, den beabsichtigten *ShapeMode* systematisch zu erraten: Daten mit expliziter Angabe von Biegeradien oder Biegerollendurchmessern verwenden typischerweise die *realistic* Form; Daten ohne diese Angaben sind meistens in *schematic* Form. Diese Regel hat aber keine uneingeschränkte Gültigkeit.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

- 002 = *polygonal*

Einige CAD-Systeme verwenden auch eine **gemischte Darstellung**: die gebogenen Eisen werden im *schematic*-Mode dargestellt, aber die geraden Eisen werden an ihre reale Raumposition gesetzt. Die Eisen schneiden sich also nicht im Eisenkern und werden in ihrer Position daher nicht über eine *Master/Slave*-Beziehung korrigiert.

3.8.2 ReinforcementType (Bewehrungslagen)

3.8.2.1 Definition der Bewehrungslagen

Die Bewehrungs-Typen folgen weitgehend der UNICAM-Definition; es wurden jedoch Zusatzdefinitionen für die Korbfertigung eingeführt:

0 = ohne Definition.

1 = erste Eisenlage; **bei Korbfertigung: Bügel**.

2 = zweite Eisenlage; **bei Korbfertigung: Längseisen**.

3 = Spieße.

4 = übrige Bewehrung (dritte Eisenlage).

5 = obere Bewehrung erste Eisenlage.

6 = obere Bewehrung zweite Eisenlage.

7 = obere Bewehrung übrige Eisen (obere Bewehrung dritte Eisenlage).

8 = lose Eisen (nicht verschweißt, nicht an andere Eisen gebunden).

Für Extirons entspricht der ReinforcementType dem UNICAM Extiron-Type.

3.8.2.2 Festlegung der Bewehrungslagen

Für flache Matten wählt man die Bewehrungslage direkt über die Solllage der Eisen: jene Eisen, die in der Produktionspalette unten liegen (= außen in der Doppelwand), bilden die Lage 1; die darüber liegenden Eisen bilden die Lage 2.

Für gebogene Matten und Körbe wird folgende Festlegung empfohlen:

a) Falls nur in einer Richtung gebogen wird, so sollte es die 1. Lage sein, die gebogen wird.

b) Falls in beide Richtungen gebogen wird, soll es möglich sein, bei der ausgerollten Matte die 1. Lage als letzte zu biegen.

Der Punkt b) ist so zu verstehen, dass beim Ausrollen der Matte zunächst die erste Lage auszurollen ist, dann die zweite. Beim Ausrollen der ersten Lage werden die anliegenden Lage-2-Eisen mitgenommen; beim darauffolgenden Ausrollen der zweiten Lage werden *keine* Lage-1-Eisen mehr mitgenommen[11].

Die Lage 1 kann auch als *Hauptlage* oder *stabile Lage* gesehen werden: die Koordinaten-Korrekturen für Drahtdurchmesser und Biegeradien werden von der Lage 1 dominiert; die Lage 2 passt sich an die Lage 1 an, nicht aber umgekehrt.

3.8.2.3 Obere Bewehrungslagen

Die Bewehrungstypen 1, 2 und 4 bilden einen zusammenhängenden Satz von "Lagen" wie er oben beschrieben wurde. Die Bewehrungstypen 5, 6 und 7 bilden einen zweiten unabhängigen Satz, für den wieder die gleichen Regeln gelten, wie für den ersten Lagen-Satz.

---

[11] Sollte es Eisen geben, die beim Ausrollen der 2. Lage mitgenommen werden müssen, so sind diese einer 3. Lage zuzuordnen (wähle hierfür ReinforgementType = 4, "Übrige Bewehrung"). Um solche Eisen als Matteneisen Produzieren zu können, benötigt man in Längs- und Quer-Richtung eine Balkenbiegemaschine; andernfalls sind diese Eisen als lose Eisen zu produzieren und müssen manuell verschweißt werden. Eisen, die ungebunden sind, also nicht mit anderen Eisen auszurollen oder zu biegen sind, werden dem Bewehrungstyp 8 zugeordnet.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Der zweite ("oberen") Lagen-Satz wird nur dann zwingend benötigt, wenn man innerhalb eines *Steel*-Blocks 2 unabhängige Lagen-Sätze braucht (Lagen in verschiedenen *Steel*-Blöcken sind ohnedies unabhängig voneinander). In der Praxis dürfte das sehr selten vorkommen, da komplexe Bewehrungskörbe üblicherweise in mehrere *Steel*-Blöcke getrennt werden, die üblicherweise sogar zu unterschiedlichen Zeitpunkten gefertigt werden (Siehe z.B. "Deckelmatte" bei Massivwänden).

Die oberen Lagen werden somit hauptsächlich für die Kompatibilität zu UNICAM benötigt. PXML-Implementierungen sollten die oberen und die unteren Lagen gleich behandeln, denn es ist durchaus möglich, einen *Steel*-Block vorzufinden, der nur obere Lagen enthält[12].

3.8.3 SteelQuality

Stahl-Güte.

3.8.4 PieceCount, Diameter, X, Y, Z

Stückzahl, Durchmesser und Offsets.

3.8.5 RotZ

Für die Verlegung eines Eisens wird zunächst das Koordinatensystem um den Winkel ***RotZ*** um die z-Achse gedreht. Alle weiteren Richtungsangaben beziehen sich auf das so gedrehte Koordinatensystem (Siehe auch Segmentorientierung, Abschnitt 3.8.16.1).

Der zulässige Bereich für *RotZ* ist

$$RotZ \in]-180°, 180°].$$

RotZ legt die Verlege-Richtung des Eisens fest und entspricht in UNICAM in etwa dem Winkel zur x-Achse (letztere liegt jedoch im Bereich [-360, 360°[). Wenn also das erste Teilstück des Eisens (oder das gesamte Eisen) in der xy-Ebene liegt, ist *RotZ* der Winkel, den das Eisen mit der x-Achse einschließt.

Für den allgemeinen Fall ist die Festlegung von *RotZ* etwas komplizierter:

Angenommen, die Biege-Ebene der ersten Biegung schneide die xy-Ebene; RotZ ist dann der Winkel, den die Schnittlinie mit der x-Achse einschließt.

Genauer heißt das: wenn $\vec{v}_0$ und $\vec{v}_1$ die ersten beiden Teilstücke des Eisens sind, dann ist

$$\boxed{RotZ = \arctan\left(\frac{v_{0y}v_{1z} - v_{0z}v_{1y}}{v_{0x}v_{1z} - v_{0z}v_{1x}}\right) + k\pi\,.}$$

(Es kann durchaus vorkommen, dass der Nenner 0 ist; das Ergebnis ist dann natürlich $\pm\pi$. Wirklich undefiniert ist das Ergebnis nur wenn Nenner und Zähler 0 sind; dieser Fall tritt aber dann ein, wenn $\vec{v}_0$ und $\vec{v}_1$ parallel oder antiparallel sind, oder beide in der XY-Ebene liegen, oder nur ein Segment existiert).

Obige Gleichung enthält noch den unbestimmten Summanden $k\pi$, wobei k gleich 0 oder ±1 ist. Tatsächlich bestimmt die angeführte Überlegung dieses Detail nicht, denn eine Richtung ist zunächst gleich gut, wie die um 180° entgegengesetzte. Um *k* festzulegen kann man fordern, dass die Projektion von $\vec{v}_0$ die Richtung festlegt, dass also[13]:

$$\boxed{\vec{v}_r \cdot \vec{v}_0 \geq 0 \quad \text{bei } \vec{v}_r := (\cos(RotZ), \sin(RotZ), 0)}$$

Herleitung der Gleichung für *RotZ*:

[12] Die obere Matte ("Deckelmatte") von Massivwänden könnte beispielsweise ausschließlich obere Lagen enthalten. Für eine PXML-Implementierung soll dies jedoch irrelevant sein.

[13] Diese Einschränkung führt nicht nur zu einem intuitiv sinnvollen *RotZ*, sondern vereinfacht auch andere Berechnungen, wie man im Abschnitt 3.8.16.12 sehen wird.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Sei λ derart, dass

$$\vec{v}_0 - \lambda\,\vec{v}_1 = \begin{pmatrix} x_p \\ y_p \\ 0 \end{pmatrix}.$$

Der Punkt (x_p, y_p) ist die Projektion von $\vec{v}_0$ in der XY-Ebene, wenn entlang des Vektors $\vec{v}_1$ projiziert wird. Aus der Z-Komponente der obigen Gleichung ergibt sich $\lambda = v_{0z} / v_{1z}$.

Wegen $RotZ = \arctan(y_p / x_p)$ ergibt sich dann obige Beziehung für *RotZ*.

Ein Problem hat man, wie gesagt, bei $x_p = y_p = 0$. Im diesem Fall ist es sinnvoll einfach nur die Projektion des ersten Teilstückes zu betrachten:

$$RotZ = \arctan2(v_{0y}, v_{0x}), \qquad (\text{bzw. } RotZ = 0 \quad \text{bei} \quad v_{0x} = v_{0y} = 0).$$

Anmerkung: Grundsätzlich ist die Angabe von *RotZ* überflüssig, denn bei beliebigem *RotZ* kann man über geeignete Segmentwinkel *RotX* und *BendY* (Siehe Abschnitt 3.8.16.1) jeden beliebigen Richtungswechsel durchführen. *RotZ* wurde nur deshalb eingeführt, um den wichtigen Spezialfall der konstanten Biegeebene einfach und anschaulich behandeln zu können. Aus diesem Grund ist obige Festlegung von *RotZ* nur als Empfehlung zu verstehen. Im Prinzip steht es jedem frei *RotZ* nach Belieben zu setzen, oder auch überhaupt nicht zu verwenden (d.h. *RotZ=0* zu lassen).

3.8.6 ArticleNo

Artikel-Bezeichnung des Rundstahl-Eisens.

3.8.7 NoAutoProd

Gesetzt, wenn das Eisen*nicht* automatisch von der Stahlmaschine produziert werden soll.

Wird typischerweise für Lagerware genutzt, die in Standardlängen vorproduziert wurde, und folglich von der Just-in Time-Produktion ausgenommen wird.

3.8.8 ExtIronWeight

Gewicht eines *Exiron*-Eisens in *Kg*. Wird nur für *ExtIron*-Eisen verwendet (denn diese haben i. all. Keine Dimensions-Angaben).

3.8.9 Bin

Fachzuweisung (z.B. für die Zuweisung eines Wagenfachs der Stahlmaschine).

3.8.10 Pos

Text-Feld für Positionsangabe des Einzelstabes.

3.8.11 Note

Text-Feld für Notiz zum Eisen.

3.8.12 Machine

Text-Feld für Angabe der Produktions-Maschine.

Eine maschineninterne Produktionsliste kann optional auch als eigene Maschine behandelt werden. In diesem Fall wird empfohlen, folgendes Format zu wählen:

MSR:2

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

In diesem Beispiel ist "MSR" die eigentliche Maschinenangabe, und "2" die Nummer der Produktionsliste.

3.8.13 BendingDevice

Biegevorrichtung. Ist ein Textfeld, in dem normalerweise der Durchmesser (in mm) der zu verwendenden Biege-Matrize angegeben werden kann. Da es ein Textfeld ist, kann die Angabe jedoch auch allgemeiner sein (z.B. Matrizenbezeichnung). Das Feld wird in BVBS auf das Header-Feld 's' abgebildet[14].

Die Angabe der *BendingDevice* impliziert oft einen Mindest-Biegeradius. Bei einer Biegematrize mit Durchmesser *D* hat ein Draht mit Durchmesser *d* einen Mindest-Biegeradius von

$$R_{\min} = D/2 + d/2 + k_r \,.$$

(k_r ist hierbei ein Radius-Korrektur-Wert für die Rückfederung, also die Differenz zwischen Radius am Draht und an der Biegematrize. Vereinfachend wird dieser Wert häufig zu 0 angenommen)[15].

Daraus ergibt sich natürlich ein potentieller Konflikt mit der Biegeradius-Angabe des Segmentes (siehe Abschnitt 3.8.16.3). In diesem Fall ist immer der größere Biegeradius relevant.

3.8.14 Spacer

Ein Spacer-Eintrag beschreibt einen einzelnen Abstandhalter. Im Gegensatz zu einigen anderen Daten-Formaten werden in PXML die Abstandhalter stets einzeln gelistet (es gibt nicht die Möglichkeit eine feste Teilung einzugeben).

3.8.14.1 Type

Abstandhalter-Typ; entspricht typischerweise der Betondeckung in mm, dividiert durch 5.

3.8.14.2 Position

Position des Abstandhalters entlang des Eisens. Die Positionsangaben beziehen sich auf die *theoretischen* Längen.

3.8.15 WeldingPoint

3.8.15.1 WeldingOutput

Schweißleistung in %.

3.8.15.2 Position

Position des Schweißpunktes entlang des Eisens.

---

[14] In UNICAM wird das Feld (in Erweiterung der offiziellen Definition) auf das Reserve-Feld RODSTOCK Zeile 5, Spalten 5-7 abgebildet, wobei hier aber nur numerische Werte eingetragen werden, und auch das auf 3 Zeichen limitiert.

[15] Die Berücksichtigung von k_r ist in nur dann nötig, wenn große Genauigkeitsansprüche vorliegen. In der Praxis kann das vorkommen, wenn gebogene Eisen oder Körbe maschinell weiterverarbeitet werden. Dann muss zum einen die Gesamtgeometrie des Eisens gut bekannt sein, und diese hängt bei Biegungen über 90° eben auch von k_r ab. Vor allem müssen aber die realen Segmentlängen am ausgerollten Eisen mit hoher Genauigkeit bekannt sein. Es wird daher empfohlen, die messtechnische Ermittlung von k_r über die realen Eisenlängen zu machen. Dadurch wird nicht nur die Rückfederung berücksichtigt, sondern auch die Tatsache, dass die längenneutrale Fase der Biegung eventuell nicht exakt in der Eisenmitte verläuft (wenn sie weiter innen verläuft, wird die Eisenmittellinie beim Biegen gelängt, was über eine Erhöhung von k_r rechnerisch berücksichtigt werden kann).
Man beachte, dass k_r keine Konstante ist, sondern abhängig vom Draht. für dünne Drähte ist die Rückfederung deutlich größer als für dicke Drähte.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.8.15.3 WeldingPointType, WeldingPrgNo

Der *WeldingPointType* enthält eine Typ-Kennung des Schweißpunktes. Die *WeldingPrgNo* bestimmt direkt das Schweißprogramm das verwendet werden soll. Normalerweise wird man höchstens ein der beiden Angaben verwenden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.8.16 Segment

Ein *Segment* ist ein Teilstück eines Rundstahl-Eisens. Ein gerades Eisen besteht aus genau einem Segment; gebogene Eisen haben für jedes Teilstück ein Segment (d.h. bei *n* Biegungen hat man *n+1* Segmente).

Das erste Teilstück beginnt an den für das Eisen angegebenen X/Y/Z-Koordinaten; jedes weitere Teilstück beginnt am Endpunkt des vorhergehenden Teilstückes.

Die Angabe des **Type**-Attributes ist optional. Das *Type*-Attribut kann folgende Werte annehmen:

- Type = „normal“: das Segment beschreibt eine normale Strecke (das ist der Default-Typ).
- Type = „spiral“: das Segment beschreibt eine Spirale. Siehe Abschnitt 3.8.16.10

3.8.16.1 Segment-Orientation (RotX, BendY)

Für jedes Teilstück werden zwei Winkel **RotX** und **BendY** angegeben, die eine Drehung des Koordinatensystems beschreiben: das Koordinatensystem wird zunächst um die x-Achse um den Winkel *RotX* gedreht; anschließend wird um die neue negative y-Achse um *BendY* gedreht[16].

$$RotX \in]-90°, 90°] \text{ (nur bedingt[17])}.$$

$$BendY \in]-180°, 180°] \text{ (nur bedingt[18])}.$$

Die x-Achse des gedrehten Koordinatensystems gibt nun die Richtung des betreffenden Segmentes an. Die Drehungen der einzelnen Teilstücke summieren sich, d.h. beim 2. Teilstück wird (das bereit zuvor gedrehte) Koordinatensystem des 1. Teilstückes weitergedreht.

RotX dreht die Biege-Ebene und *BendY* entspricht dem Biegewinkel (außer für das erste Segment).

3.8.16.2 Segment-Length (L)

Die **Segmentlänge L** (Einheit mm) entspricht der sogenannten **theoretischen Länge** des Teilstückes (gemessen in der Mitte des Eisens). Falls der Biegeradius *R* größer als *0* ist, weicht die Länge *L* von der realen Eisen-Länge ab.

Bei der Darstellung von *Spiralen* (Siehe Abschnitt 3.8.16.10) gibt L den *Bogen-Radius* an.

3.8.16.3 Bending-Radius (R)

Der **Biegeradius R** beschreibt die Krümmung der Biegungen (Einheit mm).

Falls der Wert *0* ist, können die einzelnen Systeme autonom einen Biegeradius annehmen und entsprechende Längen-Korrekturen vornehmen.

Die Biegeradius-Angabe des *ersten Segmentes* ist zu ignorieren und wird stets als 0 angenommen. Dasselbe gilt für das erste Segment nach einer Spirale.

Bei Spiralen ist die wird der Wert von *R* ebenfalls ignoriert (siehe Abschnitt3.8.16.10).

Die Schenkellänge *L* eines Segmentes ist als *theoretische Länge* zu verstehen; bei *R>0* ist die *reale Länge* von *L* verschieden.

---

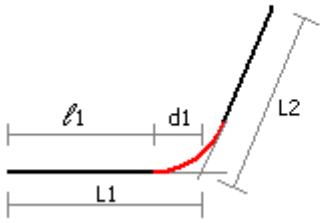
[16] Für *BendY* wird die *negative* Y-Achse verwendet, um die Biegewinkel kompatibel zu BVBS und UNICAM zu halten.

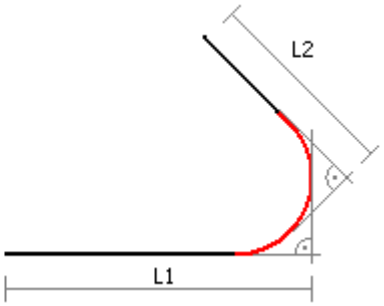
[17] Von der Definition her darf *RotX* beliebige Werte annehmen. Es ist aber empfehlenswert, Werte zwischen ±90° zu verwenden, da dieser Bereich ausreicht um jede beliebige Form darstellen zu können: *RotX*-Werte außerhalb dieses Bereiches können durch Addieren (oder Subtrahieren) von π in diesen Bereich hineintrasformiert werden, wenn man gleichzeitig alle folgenden *BendY* invertiert. Durch die Einschränkung auf den Bereich zwischen ±90° gewinnt die Darstellung an Eindeutigkeit, und man vermeidet den Fall *RotX*=±180°, der die Darstellung in unnatürlicher Weise verkomplizieren würde, da dieser Fall auch über *RotX*=0 beschrieben werden könnte.

[18] Die Beschränkung von *BendY* auf ±180° ist nur für Biegeradius *R=0* sinnvoll; bei *R>0* kann *BendY* im Bereich ±360° liegen. Um schließlich die Darstellung beliebig langer Spiralen auf einfache Weise zu ermöglichen, ist es in PXML erlaubt, für *BendY* beliebige Werte anzugeben, d.h. auch solche, die über ±360° hinausgehen.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Die Definition der theoretischen Länge ist folgenden Abbildungen zu entnehmen:





Für Winkel, die nicht größer als 90° sind, beeinflusst *R* die globale Struktur der Biegeform *nicht*, sondern führt lediglich zu einer Abrundung der Ecken. Die prinzipielle Form ist dann durch die theoretischen Längen gegeben.Bei Biegewinkel über 90° hat *R* direkten Einfluss auf die Gesamt-Abmessungen.

Radius-Definition über Angabe der ***BendingDevice***:

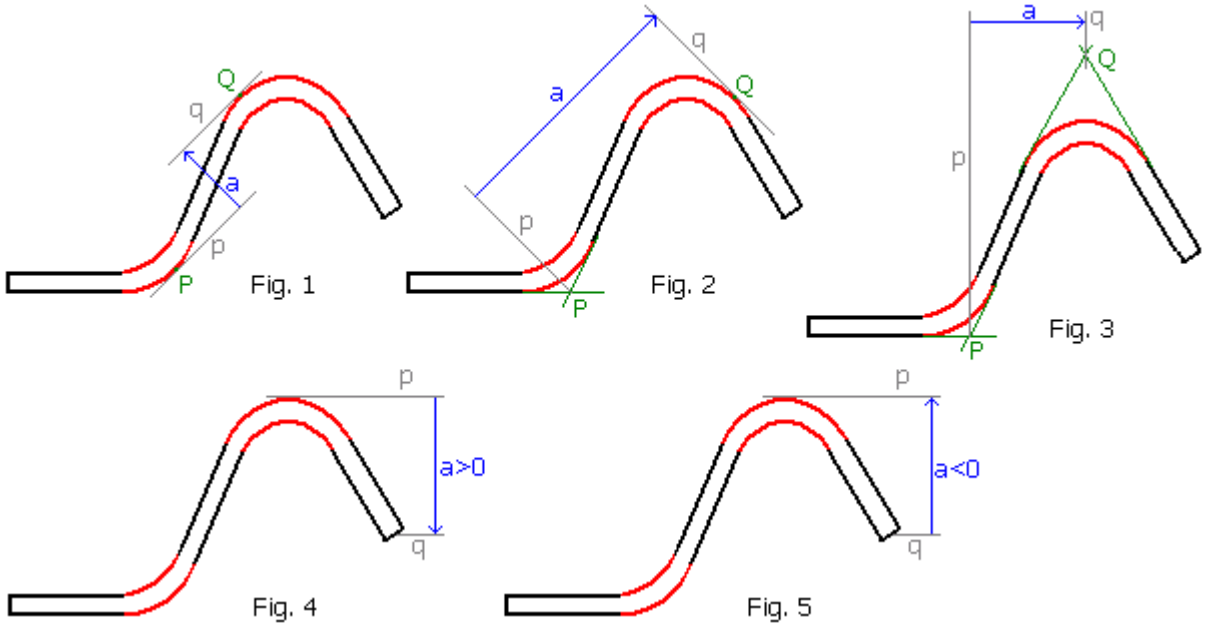
Alternativ zu direkten Radius-Angabe im Segment, kann es eine implizite Biegeradius-Bestimmung über die *BendingDevice* geben (Siehe Abschnitt 3.8.13). Falls beides gegeben ist (*BendingDevice* und *R*), ist der größere Radius relevant.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.8.16.4 Segment-Außenmaße

Normalerweise werden immer *Kernmaße* betrachtet, d.h. alle Abmessungen beziehen sich auf die Eisen-Mitte. In einigen Fällen (insbesondere beim Entwerfen komplexer Biegeformen) ist es sinnvoll Außenmaße zu betrachten.

Ein **Segment-Außenmaß** ist erst durch die Angabe einer **Bemaßungsrichtung** eindeutig festgelegt.



Die oben stehenden Abbildungen zeigen, wie die Segment-Außenmaße für verschiedene Bemaßungsrichtung definiert sind. Die *Bemaßungsrichtung* (Richtung des blauen Pfeiles) definiert 2 *Maßhilfslinien p* und *q* (graue Linien), die orthogonal zur Bemapungsrichtung sind. Das *Segment-Außenmaß* entspricht dem Abstand der Maßhilfslinien *p* und *q*. Das *Segment-Außenmaß* ist *positiv*, wenn die Normale von *p* nach *q* parallel zur Bemaßungsrichtung ist; das Maß ist *negativ*, wenn die Normale von *p* nach *q* antiparallel zur Bemaßungsrichtung ist (*p* ist die Hilfslinie am *Beginn* des Segments, *q* ist die Hilfslinie am *Ende* des Segments).

Fig. 1: Falls möglich, sind *p* und *q* Tangenten (der Bogen-Außenseite).

Fig. 2: Hier kann *p* nicht Tangente sein. Stattdessen gehr *p* durch den Schnittpunkt*P* der beiden "Rand"-Tangenten des Bogens.

Fig. 3: Hier ist weder *p*noch*q* eine Tangente; die Linien werden hier über die Schnittpunkte *P* und *Q*festgelegt.

Fig. 4: Falls ein Segment an einem Ende keine Biegung hat (erstes oder letztes Segment), bezieht sich die Bemaßung auf den Eisen-Kern (Dies gilt jedoch nicht für *konventionelle Außenmaße*; siehe hierzu Abschnitt 3.8.16.8).

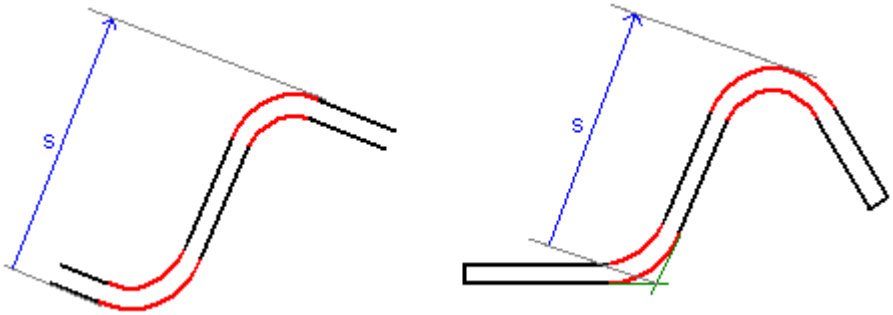
Fig. 5: Der positive Abstand wird stets von *p* nach *q* gezahlt (*p* an Segment-Beginn). Wenn dies entgegen der Bemaßungsrichtung ist, so ist der Abstand *negativ*.

Für Biegewinkel, die betragsmäßig größer oder gleich 180° sind, ist obige Definition mehrdeutig. In diesem Fall wird die Maßhilfslinie so gelegt, dass sie Tangente des ersten Teils der Biegung ist (so als ob der Biegewinkel unter 180° wäre).

Formale Definition: Die **Messpunkte***P* und *Q* sind entweder Tangenten-Punkte, oder Schnittpunkte der Rand-Tangenten (wobei immer nur Bögen betrachtet werden, die kleiner als 180° sind). Das **Segment-Außenmaß** ist das innere Produkt des Vektors PQ mit dem Einheitsvektor in Bemaßungsrichtung.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.8.16.5 Außenlänge eines Segmentes



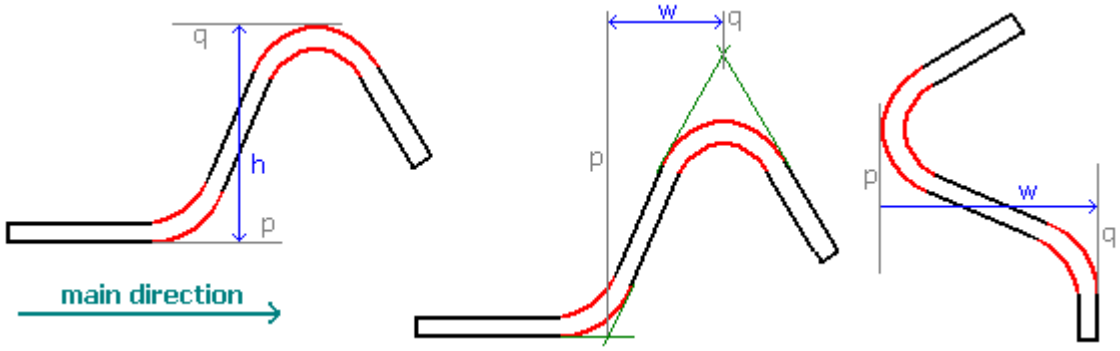
Die **Außenlänge** eines Segmentes ist jenes Segment-Außenmaß, das in Richtung des Segmentes verläuft. (Segment-Außenmaß definiert nach Abschn. 3.8.16.4.)

3.8.16.6 Höhe und Breite eines Segmentes

Um von **Höhe** und **Breite** eines Segmentes sprechen zu können, muss zunächst eine **Hauptrichtung** festgelegt werden. Dann gilt:

Breite = Segment-Außenmaß in Hauptrichtung.

Höhe = Segment-Außenmaß senkrecht zur Hauptrichtung (Winkel von +90° zur Hauptrichtung).



3.8.16.7 Rechenregeln für Außenmaße

Die Definition des Außenmaßes kann auch auf eine *Folge* von Segmenten ausgedehnt werden. Es lässt sich leicht zeigen, dass die Außenmaße **additiv** sind, d.h. das Außenmaß der Segment-Folge ist gleich der Summe der Außenmaße der Einzelnen Segmente[19].

Die Additivität kann auch verwendet werden, um ein einzelnes Segment in elementare Teilsegmente zu zerlegen; ein normales Segment kann hierbei als Folge von 3 Segmenten behandelt werden:

1) Segment mit Biegung am Beginn, gerades Teilstück der Länge 0, keine Biegung am Ende.
2) Gerades Teilstück.
3) Segment mit keiner Biegung am Anfang, gerades Teilstück der Länge 0, Biegung am Ende.

3.8.16.8 Konventionelle Außenmaße

Falls ein Segment-Ende keine Biegung hat, bezieht sich das Außenmaß auf den Eisen-Mittelpunkt (Eisen-Kern). Dies wurde so definiert, um möglichst einfache und klare theoretische Verhältnisse zu schaffen.

Für den Anwender wirkt es jedoch eher unnatürlich, wenn am Beginn und am Ende des Eisens auf den Eisen-Kern Bezug genommen wird. Aus diesem Grund wird der Begriff des **konventionellen**

[19] Die Additivität der Außenmaße gilt natürlich nur dann, wenn für alle dieselbe Bemaßungsrichtung verwendet wird, und allen Außenmaßen das korrekte Vorzeichen zuordnet wird. So hat beispielsweise ein Segment, das orthogonal zur Bemaßungsrichtung steht und am Beginn 90° und am Ende -90° hat, ein Außenmaß, das gleich dem negativen Drahtdurchmesser ist.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Außenmaßes eingeführt: hierbei wird am Beginn und am Ende des Eisens nicht auf den Eisen-Kern Bezug genommen, sondern auf eine Außenseite des Eisens, und zwar auf jene Seite, die auch für die Bemaßung des anderen Segment-Endes verwendet wird[20].

3.8.16.9 Allgemeine Berechnungen zum Biege-Radius

Für die Länge b des Bogens einer Biegung um α gilt:

$$b = R \cdot |\alpha| .$$

Für das gerade Stück ℓ gilt:

$$\ell = L - d, \quad d = R \tan(\frac{\alpha_M}{2}), \quad \alpha_M := \min(|\alpha|, 90°)$$

Für die reale Länge L_R des Eisens gilt demnach

$$L_R = \ell + \frac{b}{2} = L + \frac{b}{2} - d = L + R(\frac{|\alpha|}{2} - \tan(\frac{\alpha_M}{2})) .$$

Das alles gilt natürlich nur, wenn $L >= d$ ist, oder anders ausgedrückt, wenn

$$L >= R \tan(\frac{\alpha_M}{2}) .$$

Es gibt verschiedene Möglichkeiten, diese Bedingung zu erzwingen; die einfachste ist wohl die, L bei Bedarf auf obigen Mindest-Wert anzuheben.

3.8.16.10 Bögen und Spiralen in klassischer Spiral-Form

Traditionell werden Bögen und Spiralen in gesonderter Form behandelt, um möglichst direkt und explizit als Bögen behandelt werden zu können. Um die Kompatibilität zu älteren Systemen zu wahren, wird auch in PXML die Möglichkeit einer gesonderten Bogendarstellung geboten, und zwar durch spezielle *spiral*-Segmente[21].

Eine *spiral*-Segment wird durch die Typ-Kennung „spiral“ gekennzeichnet.

Die Segment-Parameter haben dann folgende Bedeutung:

- ***BendY***: Windungs-Winkel der Spirale
  (kann beliebige Werte annehmen, ist also *nicht* auf ±360° beschränkt).
- ***L***: Radius der Spirale.
- ***R***: Hat keine Bedeutung und wird ignoriert[22].
- ***RotX***: Definiert die Ganghöhe der Wendel[23].
  Die Höhen-Zunahme pro Windung ist $G = L \cdot \sin RotX$.

Verbindung des Bogens zum geraden Segment:

Falls Bögen (Spiralen) mit geraden Segmenten verbunden werden, ist der Übergang Bogen-Strecke stets *ohne Biegung*. Am Beginn des Bogens muss das schon aufgrund des Datenformats so sein: da *BendY* den Bogenwinkel beschreibt, hat man keine Möglichkeit mehr, zusätzlich einen Biegewinkel anzugeben. Am Ende des Bogens, also am Beginn des darauffolgenden geraden Segmentes, wäre

---

[20] Für ein gerades Eisen ist dies nicht definiert. Dort macht es aber keinen Unterschied ob man die Bemaßung auf den Eisen-Kern oder auf eine Außenseite bezieht.

[21] Obschon *spiral* Segmente in PXML vorgesehen sind, sollten neuere Systeme erwägen, die Bögen in normaler PXML-Darstellung zu beschreiben, wie das im Abschnitt 3.8.16.11 beschrieben wird.

[22] Beachte: die Spezifikation legt fest, dass *R* ignoriert wird, und folglich beliebige Werte annehmen kann. Es wird den Implementierungen *nicht* freigestellt, *R* anderweitig zu verwenden und dadurch die Freiheit beim Setzen von *R* einzuschränken. Diese Festlegung ist notwendig, weil der Radius *R* oft über übergeordnete Parameter (Matrizen-Parameter) für alle Segmente gesetzt wird – es soll dann nicht nötig sein, hiervon Spiralen auszunehmen.

[23] Auf die Möglichkeit, bei Spiralen ein *echtes RotX* anzugeben, wird hier bewusst verzichten, da dies in der Praxis kaum nötig sein wird. Sollte das dennoch einmal benötigt werden, muss man der Spirale ein Hilfs-Segment der Länge 0 voranstellen.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

es im Prinzip möglich, eine Biegung anzugeben. Dieser Freiheitsgrad soll jedoch nicht genutzt werden, d.h. das nachfolgende Segment soll *BendY=0* haben. Diese Einschränkung ist sinnvoll um am Beginn und am Ende des Bogens symmetrische Verhältnisse zu haben, und insgesamt die Rechnerischen Verhältnisse zu vereinfachen. Zudem passt diese Einschränkung auch gut zur BVBS-Kodierung und zur Funktionsweise der Biegemaschinen: in beiden Fällen müssen Biegungen am Beginn und am Ende eines Bogens über separate Segmente kodiert werden.

Spiralen und Darstellungsart:

Die Behandlung von Spiralen ist grundsätzlich nur für die *realistic*-Darstellung definiert (Siehe *ShapeMode*, Abschnitt3.8.1). Die *schematic*- und *polygonal*-Darstellungs-Konzepte lassen sich nicht sinnvoll mit dem Spiral-Darstellungs-Konzept verbinden, und die beiden Konzepte gehören auch applikationstechnisch ganz unterschiedlichen Domänen an.

3.8.16.11 Bögen normaler PXML-Form

Ein Bogen kann in PXML auch einfach als Biegung dargestellt werden. Der Bogen ist dann nicht mehr als solcher gesondert gekennzeichnet und unterscheidet sich von einer normalen Biegung nur dadurch, dass er einen großen Biegeradius hat, und daher typischerweise nicht durch Biegen über eine Matrize hergestellt wird, sondern über andere Maschinenvorrichtungen. Tatsächlich kann es aber von Vorteil sein, die Art der maschinellen Herstellung nicht in den Daten vorwegzunehmen, sondern der Maschinensoftware zu überlassen; die PXML-Daten beschränken sich dann auch die geometrische Beschreibung des Produktes, ohne das Herstellungsverfahren vorzuschreiben.

Wenn Bögen mit normalen PXML-Segmenten beschrieben werden, müssen folgende Punkte berücksichtigt werden:

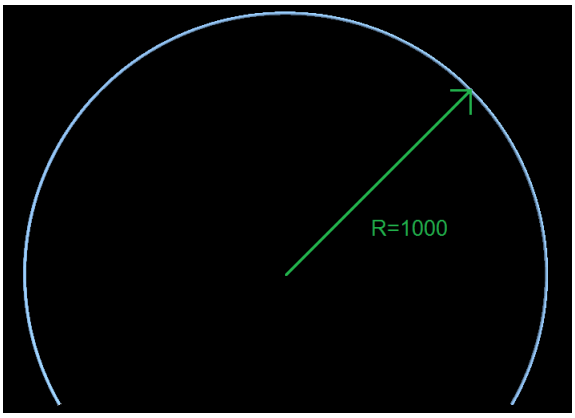
a) Am Beginn des ersten Segmentes kann es keine Biegung geben. Man braucht für die Darstellung eines Bogen-Eisens somit mindestens 2 Segmente. (Das ist der einzige wirkliche Nachteil der normalen PXML-Segmente gegenüber den *spiral*-Segmenten).

b) Wenn es vor dem Bogen eine Biegung gibt, muss diese in einem eigenen Segment untergebracht werden. (Das ist auch bei *spiral*-Segmenten so).

c) Ein Bogen ist durch seinen Radius und den Winkel bereits vollständig definiert. Der L-Wert des Segmentes ist redundant und müsste exakt den Wert $L = R\tan(\alpha_M/2)$ einnehmen (Siehe Abschnitt 3.8.16.9). Um diese Redundanz in den Daten zu vermeiden, wird empfohlen, *L=0* zu setzen. (Außer dann, wenn auf den Bogen ein gerades Stück folgt, dann ist L entsprechend größer als $R\tan(\alpha_M/2)$).

d) Die Darstellung einer Wendel-Gang-Höhe von Spiralen ist (derzeit) nicht vorgesehen.

Beispiele 1: Einfacher Bogen



Ein Bogen mit Bogenradius 1000mm (Kernmaß) und einem Winkel von 120° wird über 2 Segmente dargestellt:

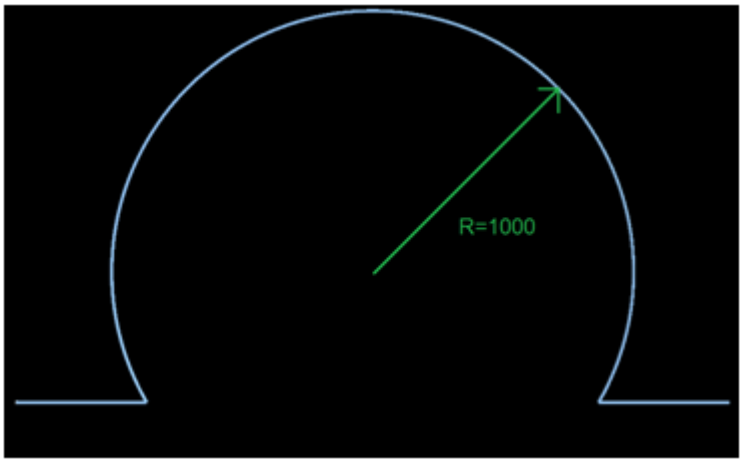
Segment 1: L=0, BendY=0, R=0

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Segment 2: L=0, BendY=240, R=1000

Die Länge des Bogens ist 4189mm; diese Länge wird nicht explizit angegeben, da sie aus Radius und Winkel errechnet werden kann. Die Segmentlänge *L* wird (wie oben erwähnt) ebenfalls nicht angegeben, oder einfach auf 0 gesetzt.

Beispiele 2: Bogen mit Endaufbiegungen



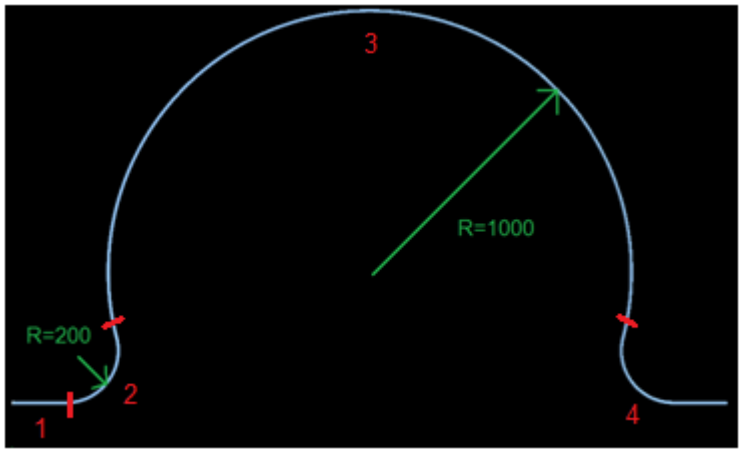
Segment 1: L=500, BendY=0, R=0

Segment 2: L=0, BendY=120, R=0

Segment 3: L=0, BendY=-240, R=1000

Segment 4: L=500, BendY=120, R=0

Beispiele 3: Bogen mit Endaufbiegungen mit Biegeradius



Segment 1: L=404, BendY=0, R=0

Segment 2: L=0, BendY=104, R=0

Segment 3: L=0, BendY=-208, R=1000

Segment 4: L=404, BendY=104, R=0

Die roten Linien und roten Ziffern im Bild beschreiben die Segmente.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.8.16.12 Allgemeine Berechnungen zur Koordinaten-Drehung

Es gibt insgesamt 3 Arten von Koordinaten-Drehungen:*RotZ*, *RotX* und *BendY*. Die Drehung um die z-Achse (*RotZ*) erfolgt für jeden Stab nur einmal, und ist – wie weiter oben erläutert – eigentlich frei wählbar. Die Drehungen um die x- und y-Achsen (*RotX* und *BendY*) können für jedes Teilstück angegeben werden.

Sei $\underline{v}$ die Koordinatendarstellung eines Vektors im ursprünglichen Koordinatensystem und $\underline{v}'$ die entsprechende Darstellung im gedrehten System; dann ist die Transformation durch eine **Dreh-Matrix** *D* beschrieben:

$$\underline{v}' = D \cdot \underline{v}.$$

Für die oben genannte Drehungen gilt[24]:

$$D_{RotZ} = \begin{pmatrix} \cos RotZ & \sin RotZ & 0 \\ -\sin RotZ & \cos RotZ & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$D_{RotX} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos RotX & \sin RotX \\ 0 & -\sin RotX & \cos RotX \end{pmatrix}, \quad D_{BendY} = \begin{pmatrix} \cos BendY & 0 & \sin BendY \\ 0 & 1 & 0 \\ -\sin BendY & 0 & \cos BendY \end{pmatrix}$$

$$D_{RotX,BendY} = D_{BendY} \cdot D_{Rotx} = \begin{pmatrix} \cos BendY & -\sin RotX \sin BendY & \cos RotX \sin BendY \\ 0 & \cos RotX & \sin RotX \\ -\sin BendY & -\cos BendY \sin RotX & \cos BendY \cos RotX \end{pmatrix}$$

(In diesen Angaben wurde bereits berücksichtigt, dass *BendY* eine Drehung um die *negative* y-Achse beschreibt).

Sei nun $\underline{v} = (v_x, v_y, v_z)$ die Darstellung eines Einheitsvektors, die bei der Koordinatendrehung $D_{RotX,BendY}$ in den Darstellung $\underline{x}^0 = (1,0,0)$ übergeht. Es ist also $D_{RotX,BendY} \cdot \underline{v} = \underline{x}^0$, und folglich $\underline{v} = D^T_{RotX,BendY} \cdot \underline{x}^0$. Daraus ergibt sich unmittelbar:

$$v_x = \cos(BendY), \quad v_y = -\sin(RotX)\sin(BendY), \quad v_z = \cos(RotX)\sin(BendY).$$

Durch Kombinieren dieser Gleichungen erhält man

$$\tan(RotX) = -\frac{v_y}{v_z}.$$

Wenn man nun *RotX* auf den Bereich]-90°,90°] einschränkt (was, wie weiter oben erwähnt, immer möglich ist), ergibt sich:

$$\boxed{RotX = -\arctan\frac{v_y}{v_z}.}$$

Die Einschränkung auf den Bereich]-90°,90°] bewirkt nun auch dass $\cos(RotX) \geq 0$ ist. Aus $v_z = \cos(RotX)\sin(BendY)$ folgt dann $\operatorname{sgn}(v_z) = \operatorname{sgn}(\sin(BendY))$. Mit $v_x = \cos(BendY)$ erhält man schließlich

$$\boxed{BendY = \begin{cases} \arccos(v_x), & v_z \geq 0 \\ -\arccos(v_x), & v_z < 0 \end{cases}.}$$

[24] Dreh-Matrizen sind stets *orthogonal*, d.h. es gilt $D^{-1}=D^T$.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Einen Sonderfall hat man für $v_y = v_z = 0$. Man überlegt sich leicht, dass das genau dann der Fall ist, wenn $BendY = 0$ oder $BendY = 180°$ ist. Obiger "atctan"-Ausdruck zur Berechnung von *RotX* ist dann undefiniert, undtatsächlich ist *RotX* in diesem Fall nicht eindeutig bestimmbar, da bei $BendY = 0$ oder $BendY = 180°$ zwei aufeinanderfolgende *RotX* um die selbe Achse drehen; es ist dann nur die Summe dieser *RotX* eindeutig festgelegt, nicht aber die Einzel-Werte. Wenn man diesen Fall nicht gesondert behandelt, wird man dennoch ein gültiges Ergebnis erhalten: *RotX* hat dann einen zufälligen Wert, der durch Rundungsfehler dominiert wird[25]; wenn man mit diesem "zufälligen" *RotX* konsequent weiterrechnet, ist das Ergebnis insgesamt noch korrekt (das Zufälligkeit des *RotX* wird im nächsten *RotX* kompensiert). Es ist aber günstiger, diesen Fall wie folgt gesondert zu behandeln:

$$\text{bei} \quad BendY_i = 0: \quad RotX_i + RotX_{i+1} \rightarrow RotX_i, \quad 0 \rightarrow RotX_{i+1}$$

D.h. die Summe von $RotX_i$ und $RotX_{i+1}$ wird vollständig in $RotX_i$ hineingenommen; $RotX_{i+1}$ auf 0 gesetzt[26].

Ein gewisses Problem hat man dagegen, wenn *BendY* (näherungsweise) ±180° ist, also im Fall $\underline{v} = (v_x, v_y, v_z) = (-1,0,0)$. Auch dann ist *RotX* mathematisch unbestimmt, ebenso wie das Vorzeichen von *BendY*. In der Praxis hat man aber einen von 0 verschiedenen Biegeradius, so dass *BendY* und *RotX* eindeutig bestimmt sind; die Biegeradius-Information ist aber in der vereinfachten Darstellung von geraden Teilsegmenten nicht enthalten. Hier muss man *RotX* und *BendY* über zusätzliche Informationen bestimmen, eventuell auch über ein abgewandeltes *v*, das, unter Einbeziehung der tatsächlichen Biegerichtung, nichtverschwindende y- und z-Komponenten liefert[27].

---

[25] Dem numerische Sonderfall des NaN-Ergebnisses muss man auch einen reellen Wert zuordnet (z.B. 0).

[26] Diese Vorgangsweise stellt sicher, dass *RotX* innerhalb der Biegeform nur dann von 0 verschieden ist, wenn es eine echte Drehung der Biegeform gibt. Bei ebenen Biegeformen wird *RotX* nur benötigt, um den Lagewinkel des gesamten Eisens anzugeben, und man möchte dann, dass diese Angabe in $RotX_0$ erfolgt (und nicht in $RotX_1$).
Schließlich sei noch erwähnt, dass der Falls *BendY*=0 im Innern der Biegeform kaum auftritt (das wäre eine Aufeinanderfolge von 2 Segmenten ohne dazwischenliegende Biegung – das ist eigentlich ein entarteter Fall). Der erste *BendY*-Winkel ist aber sehr häufig 0, und zwar für Formen deren erstes Segment in der XY-Ebene liegt.
Der Vollständigkeit halber sei noch erwähnt, dass der Fall $v_y \neq 0$, $v_z = 0$ numerisch unproblematisch ist. Die "arctan"-Funktion zur Bestimmung von *RotX* liefert dann ±90°, wobei das Vorzeichen wiederum zufällig aus Rundungsfehler entsteht (das v_z kann rundungsfehlerabhängig geringfügig positiv oder geringfügig negativ sein). Diese Zufälligkeit wird dann in den folgenden Berechnungen wieder kompensiert und stellt kein Problem dar, da kein "künstlicher" von 0 verschiedener Wert eingefügt wurde.

[27] Für das erste Segment gilt dies Überlegung nicht, da hier der reale Biegeradius 0 ist (*BendY* gibt beim ersten Segment nur eine Richtung angibt, und keine Biegung). Das ist aber kein Problem, da der Fall *BendY*=±180° beim ersten Segment ohnedies vermieden werden kann, wenn man die Einschränkung aus Abschnitt 3.8.5 beachtet, die fordert, dass *RotZ* so gewählt wird, dass $\vec{v}_r \cdot \vec{v}_0 \geq 0$; hierbei ist $\vec{v}_0$ das erste Segment und $\vec{v}_r := (\cos(RotZ), \sin(RotZ), 0)$.
Das sieht man wie folgt:
Es ist $\underline{v}_r = D_{RotZ} \cdot \begin{pmatrix} \cos(RotZ) & \sin(RotZ) & 0 \end{pmatrix}^T$ die Koordinatendarstellung von $\vec{v}_r$ nach erfolgter *RotZ*-Koordinatendrehung. Setzt man für D_{RotZ} die Drehmatrix-Koordinaten ein, so erhält man $\underline{v}_r = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$. Nimmt man nun an, dass $\underline{v} = \begin{pmatrix} -1 & 0 & 0 \end{pmatrix}^T$ ist (das ist gleichbedeutend mit *RotZ*=±180°), so ergibt sich:
$\underline{v}_r \cdot \underline{v}_0 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 & 0 \end{pmatrix}^T = 1$, was offensichtlich ein Widerspruch zu $\vec{v}_r \cdot \vec{v}_0 \geq 0$ ist.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.8.16.13 Konvertierung von und zu UNICAM

In UNICAM gibt es die Einschränkung, dass das erste Segment in der XY-Ebene liegen muss. Unter Umständen muss in UNICAM daher ein zusätzliches horizontales **Dummy-Segment** eingeführt werden, dessen Länge 0 ist. Dieses *Dummy-Segment* wird in PXML nicht benötigt[28]. Ein weiterer Unterschied zwischen UNICAM und PXML liegt in der Biege-Ebene: während in UNICAM alle Biegungen in einer Ebene liegen müssen, kann in PXML vor jeder Biegung eine Drehung der Biege-Ebene vorgenommen werden (*RotX*); bei der Konvertierung PXML → UNICAM geht dieser Freiheitsgrad natürlich verloren.

Im Folgenden werden folgende Benennungen für die UNICAM-Werte verwendet:

α_{toX}	=	Winkel zur X-Achse
α_L	=	Lagewinkel der Biege-Ebene (0 für vertikale Biege-Ebene)[29]
d_i		Länge des Segmentes i (i=0, 1, 2, ...)
α_i	=	Winkel am Ende des Segmentes i (i=0, 1, 2, ...)

UNICAM → PXML ohne *Dummy-Segment* ($d_0 \neq 0$):

$$RotZ = \alpha_{toX}$$

$$RotX_0 = -\alpha_L, \quad BendY_0 = 0, \quad L_0 = d_0$$

$$RotX_i = 0, \quad BendY_i = \alpha_{i-1}, \quad L_i = d_i, \quad i \geq 1$$

UNICAM → PXML mit *Dummy-Segment* (d0=0):

$$RotZ = \alpha_{toX}$$

$$RotX_0 = -\alpha_L, \quad BendY_0 = \alpha_0, \quad L_0 = d_1$$

$$RotX_i = 0, \quad BendY_i = \alpha_i, \quad L_i = d_{i+1}, \quad i \geq 1$$

PXML → UNICAM, allgemeine Überlegungen:

Bei Verwendung eines *Dummy-Segments* lässt sich PXML recht gradlinig zu UNICAM konvertieren. Die Sonderfälle ohne *Dummy-Segment* müssen dagegen gesondert behandelt werden, da hier die Information über $BendY_0$ nicht direkt übertragen werden kann (das *Dummy-Segment*, das diese Information tragen könnte, existiert in diesen Fällen nicht). Die Information über $BendY_0$ muss folglich irgendwie in die anderen Größen hineingenommen werden (der Fall ohne *Dummy-Segment* ist eben dadurch charakterisiert, dass das möglich ist).

Ferner wird man stets zwischen einfachen Eisen und gebogenen Eisen unterscheiden müssen: einfach Eisen haben keine Biege-Ebene, gebogene Eisen haben eine Biege-Ebene, die durch die erste echte Biegung ($BendY_1$) bestimmt wird. Natürlich muss bei der Konvertierung zu UNICAM angenommen werden, dass alle echten Biegungen in derselben Ebene liegen.

PXML → UNICAM bei nur einem Segment:

Sonderfall $RotX_0 = \pm 90°$ oder $BendY_0 = 0°$ (kein *Dummy-Segment* nötig):

$$\alpha_L = 0$$

$$\alpha_{toX} = RotZ - BendY_0 \operatorname{sgn}(\sin RotX_0)$$

$$d_0 = L_0, \quad \alpha_0 = 0$$

[28] In PXML ist es zwar prinzipiell erlaubt Segmente mit Länge 0 zu haben, es ist jedoch nicht notwendig (und daher auch nicht sinnvoll) ein horizontales Dummy-Segment zu verwenden.

[29] Hier wird α_L im Bereich]-180°,180°] angenommen. In der UNICAM-Datei wird, je nach Format, entweder α_L direkt abgespeichert, oder ein entsprechender Winkel im Bereich [0, 360°[.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Sonderfall $BendY_0 = \pm 180°$ (kein*Dummy-Segment* nötig, aber Richtungsinversion):

$$\alpha_L = 0$$

$$\alpha_{toX} = RotZ + 180°$$

$$d_0 = L_0, \quad \alpha_0 = 0$$

Alle anderen Fälle (*mit* Dummy-Segment):

$$\alpha_L = -RotX_0$$

$$\alpha_{toX} = RotZ$$

$$d_0 = 0, \quad d_1 = L_0, \quad \alpha_0 = BendY_0, \quad \alpha_1 = 0$$

<u>PXML → UNICAM bei mehreren Segmenten:</u>

Da die Biege-Ebene durch die erste Biegung bestimmt wird, ist es sinnvoll, $BendY1 \neq 0$ vorauszusetzen. Falls die erste Biegung entartet ist (Biegewinkel 0), werden die anliegenden Segmente zu einem Segment zusammengefasst (datenmäßig oder theoretisch).

$$\alpha_L = -\arcsin(\sin RotX_0 \cos RotX_1 + \cos BendY_0 \cos RotX_0 \sin RotX_1)$$

$$\alpha_{toX} = \arg(a, b), \quad \text{mit}$$

$$a = \cos RotX_0 \cos RotX_1 \cos RotZ$$
$$- \sin RotX_1 \cdot (\cos BendY_0 \cos RotZ \sin RotX_0 + \sin BendY_0 \sin RotZ)$$

$$b = \cos RotZ \sin BendY_0 \sin RotX_1$$
$$+ \sin RotZ \cdot (\cos RotX_0 \cos RotX_1 - \cos BendY_0 \sin RotX_0 \sin RotX_1)$$

Bei $RotX_0 = \pm 90°$ oder $BendY_0 = 0$ oder $BendY_0 = \pm 180°$ (<u>kein Dummy-Segment</u>):

$$d_i = L_i, \quad \alpha_i = BendY_{i+1}, \quad i \geq 0 \text{ (der letzte Winkel ist 0)}$$

Bei $RotX_0 \neq \pm 90°$ und $BendY_0 \neq 0$ und $BendY_0 \neq \pm 180°$ (<u>mit Dummy-Segment</u>):

$$d_0 = 0$$

$$\alpha_0 = \arg(u, v), \quad \text{mit}$$

$$u = \cos BendY_0 \cos RotX_0 \cos RotX_1 - \sin RotX_0 \sin RotX_1$$

$$v = \cos RotX_0 \sin BendY_0$$

$$d_i = L_{i-1}, \quad \alpha_i = BendY_i, \quad i \geq 1 \text{ (der letzte Winkel ist 0)}$$

Der Fall ohne*Dummy-Segment* bedarf jedoch noch einer Vorverarbeitung. Vor dem Export muss die Biegeform so transformiert werden, dass folgende Bedingungen erfüllt sind:

$$RotX_0 \in [-90°, 90°]$$

$$BendY_0 = 0$$

$$RotX_1 = 0$$

Diese **Dummy-Rektifizierung**kann wie folgt bewerkstelligt werden:

Dummy-Rektifizierung bei $RotX_0 = \pm 90°$:

$$RotZ_{new} = RotZ_{old} - BendY_{0,old} \operatorname{sgn}(\sin RotX_{0,old})$$

$$BendY_{0,new} = 0$$

$$RotX_{0,new} = RotX_{0,old} + RotX_{1,old}$$

$$RotX_{1,new} = 0$$

$$RotX_{0,new} \rightarrow [-90°, 90°]$$

Dummy-Rektifizierung bei $BendY_0 = 0$:

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

$$RotX_{0,new} = RotX_{0,old} + RotX_{1,old}$$

$$RotX_{1,new} = 0$$

$$RotX_{0,new} \rightarrow [-90°,90°]$$

Dummy-Rektifizierung bei $BendY_0 = \pm 180°$:

$$RotZ_{new} = RotZ_{old} + 180°$$

$$BendY_{0,new} = 0$$

$$RotX_{0,new} = 180° - RotX_{0,old} + RotX_{1,old}$$

$$RotX_{1,new} = 0$$

$$RotX_{0,new} \rightarrow [-90°,90°]$$

Hierbei ist

$$RotX_{0,new} \rightarrow [-90°,90°]$$

eine Transformation, die dafür sorgt, dass $RotX_0$ in diesem eingeschränkten Winkelbereich liegt. Falls nötig, muss man also $RotX_0$ um $\pm 180°$ verändern; dies kompensiert man, indem alle *BendY*-Werte invertiert ($BendY_0$ ist hiervon nicht betroffen, da dieser Wert zuvor bereits auf 0 gesetzt wurde).

3.8.16.14 Konvertierung von und zu BVBS-BF2D

Im BVBS-Format gibt es die Möglichkeit für jede Biegung einen Biege-Radius anzugeben. Einige Maschinen interpretieren – in Abweichung von der BVBS-Spezifikation – eine Biegung mit Radius-Angabe grundsätzlich als Spirale (eine eventuelle Längen-Angabe wird dann als Bogenlänge aufgefasst). Es wird daher empfohlen, beim Export nach BVBS die Biege-Radien nicht in den Geometrie-Block zu schreiben, sondern durch die Biegerollen-Angabe im Header-Block zu spezifizieren: Der Biegerollen-Durchmesser wird auf das das Doppelte der größten Biegeradius-Angabe gesetzt. Die Radius-Angaben für Spiralen müssen natürlich im Geometrie-Block als Radius angegeben werden; der Wert des Spiral-Radius wird dem L-Parameter des PXML-Segmentes entnommen (siehe Abschnitt 3.8.16.10).

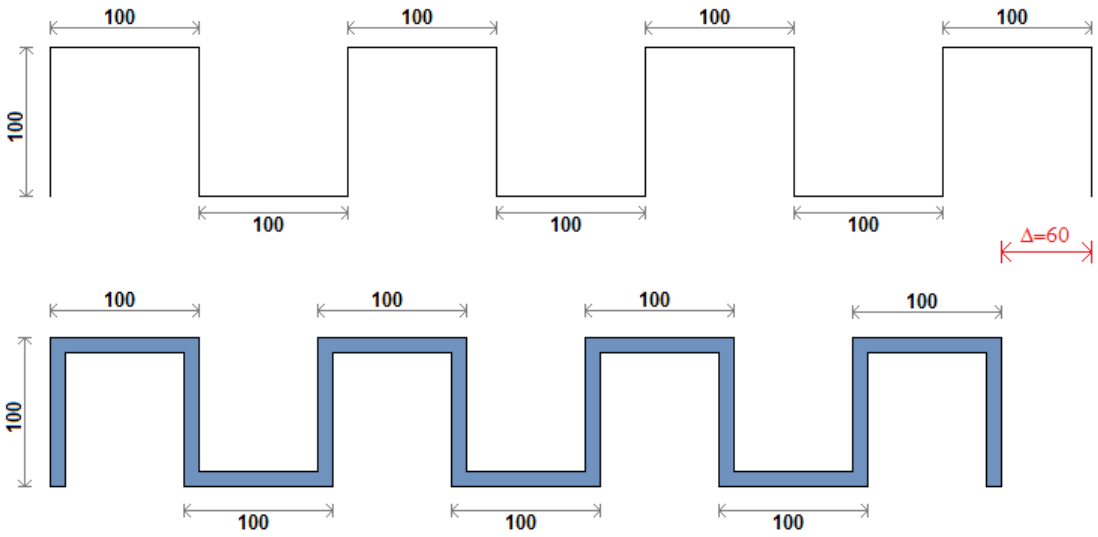
3.8.16.15 Konvertierung von und zu BVBS-BF3D

Die BVBS-Spezifikation für BF3D ist (im Gegensatz zu BF2D) alles andere als vollständig und eindeutig[30]. Es verwundert daher nicht, dass viele unterschiedliche, untereinander inkompatible, Implementierungen im Umlauf sind.

Uneinheitlich ist beispielsweise die Auffassung, ob in BF3D Außenmaße oder Eisenkernmaße zu betrachten sind. In BVBS sind ja an sich Außenmaße üblich. Mehrheitlich ist aber die Auffassung vorherrschend, dass für BF3D Kernmaße zu verwenden wären. Tatsächlich ist die Verwendung von Außenmaßen in BF3D auch eher seltsam, da die BF3D-Vektoren dann nicht mehr im oder am Eisempfad liegen, wie man an folgendem Beispiel sieht:

[30] Wir beziehen uns hier auf die allgemein verwendete BVBS-Spezifikation aus dem Jahre 2000 – eine jüngere ist uns nicht bekannt.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.



(Oben ist die BF3D-Form dargestellt, unten die reale Eisenform – wenn man annimmt, in BF3D würden Außenmaße angegeben).

Uneinheitlich ist ferner die Darstellung der Biegewinkel über 90°. Hier gibt es mindestens 3 verwendete Varianten:

a) Auffassung der BF3D-Vektoren als reine Längen- und Richtungsvektoren der Eisensegmente (Ist jedoch bei Biegewinkeln von 180° nicht mehr eindeutig).

b) Aufteilung des Winkels in Teilwinkel, mit jedem Teilwinkel kleiner gleich 90°.

c) Polygonale Darstellung (analog der polygonalen Darstellung in PXML). Ab einer bestimmten Winkelgröße (z.B. 135°) wird der Biegewinkel in Teile unterteilt (Denn Biegewinkel nahe an 180° sind in dieser Darstellung nicht möglich).

Eine weitere ungeklärte Frage ist, wie in BF3D die Absolut-Position des Eisens dargestellt werden soll. Sind die "Stabblock"-Einträge zu verwenden? ("X", "Y" oder "E"?). Es gibt Implementierungen, die hierfür eigene private Felder x/y/z im "Private"-Block einführen.

Aufgrund der genannten Auffassungsunterschiede ist es also nicht möglich eine einheitliche Definition für BF3D anzugeben. Wenn man aber versucht, die am häufigsten verwendeten BF3D-Auffassungen unter einer gemeinsamen Sichtweise zu vereinen, so ergibt sich folgende Festlegung, die zumindest kompatibel zu vielen wichtigen existierenden Implementierungen ist:

1) BF3D betrachtet Kernmaße (also nicht Außenmaße).

2) Die Geometrie der BF3D-Vektoren entspricht jener der polygonalen Darstellung in PXML. Biegewinkel über 90° können, so wie das für die polygonale Darstellung in PXML beschrieben wurde, aufgeteilt werden. Biegewinkel nahe 180° müssen aufgeteilt werden. Umgekehrt gilt (wie für die polygonale PXML-Darstellung), dass kurze Segmente zwischen zwei Biegungen als reine Darstellungshilfssegmente Aufzufassen sind; an der Maschine sind die beiden Biegungen wieder zu einer einzigen Biegung zusammenzuziehen.

3) Falls Absolutkoordinaten des Eisens angegeben werden sollen, erfolgt das im "Private"-Block mit den Feldern "x", "y" und "z".

3.9 Girder

3.9.1 PieceCount, X, Y, Z, GirderName, Length, AngleToX

AngleToX gibt die Verlege-Richtung an (ausgehend von der x-Achse).
Die Werte liegen im Bereich]-180°, 180°].

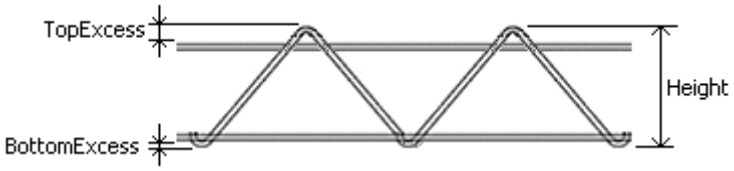
Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

X, *Y*, und *Z* beschreiben die Lage der Mittelachse, die in der Mitte der beiden Untergurt-Eisen liegt (Kernmaß des Untergurtes).

3.9.2 NoAutoProd

Gesetzt, wenn der Gitterträger *nicht* automatisch produziert werden soll.

3.9.3 Height, TopExcess, BottomExcess



Die *Height* ist die Z-Differenz vom tiefsten zum höchsten Punkt.

Der *TopExcess* ist der Z-Abstand des höchsten Punktes zur Oberkante des Obergurts.

Der *BottomExcess* istder Z-Abstand des tiefsten Punktes zur Unterkante des Untergurtes.

3.9.4 Weight, TopFlangeDiameter, BottomFlangeDiameter

*TopFlangeDiameter*und *BottomFlangeDiameter*sind in mm; *Weight* in Kg.

3.9.5 GirderType

GirderType ist eine numerische Typ-Kennung. Es wird empfohlen, folgende Werte-Festlegungen zu verwenden:

0 = Standardträger (ohne zusätzliche Definition).

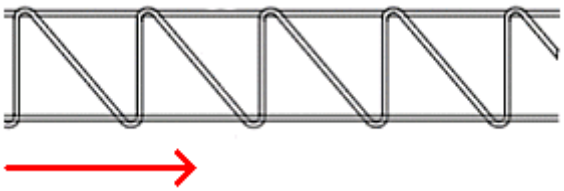
1 = Allgemeiner Grundbewehrungsträger.

2 = Schubträger.

3 = Zulageträger.

Entspricht der Einer-Stelle der UNICAM-Trägerart.

Richtung der Schubträger: die Richtung der Schubträger ist so definiert, dass die schrägen flankend abfallend sind:



3.9.6 MountingType

Einbau-Anweisung:

0 = Keine Angabe.

1 = Manuell zu befestigen (Nacharbeit nötig).

2 = Manuell einzulegen.

Entspricht der Zehner-Stelle der UNICAM Trägerart.

3.9.7 Machine

Text-Feld für Angabe der Produktions-Maschine.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Eine maschineninterne Produktionsliste kann optional auch als eigene Maschine behandelt werden. In diesem Fall wird empfohlen, folgendes Format zu wählen:

GTA:2

In diesem Beispiel ist "GTA" die eigentliche Maschinenangabe, und "2" die Nummer der Produktionsliste.

3.9.8 Period, PeriodOffset

Im Feld *Period* kann die Gitterträger-Periode angegeben werden (in mm). Typisch ist hier ein Wert von 200. Falls der Wert nicht angegeben wird, oder 0 ist, ist ein applikationsabhängiger Default-Wert anzunehmen.

Das Feld *PeriodOffset* gibt an, in welchem Abstand vom Gitterträger-Anfang der erste Perioden-Tiefpunkt liegt. Beim nicht geschnittenen Gitterträger ist dieser Wert normalerweise 0; beim gerastert geschnittenen Gitterträger ist dieser Wert 0 oder gleich der halben Periodenlänge; beim beliebig geschnittenen Gitterträger sind alle Werte möglich, positive, wie auch negative.

Export nach UNICAM: Die Felder *Period* und *PeriodOffset* werden beim UNICAM-Export in die 2 Reserve-Felder geschrieben, die sich am Beginn der Zeile 5 des BRGIRDER-Blocks befinden.

3.9.9 Width

Im Feld *Width* kann die Gitterträger-Breite angegeben werden (in mm); gemessen wird ganz außen am Gitterträger. Typisch ist hier ein Wert von 80. Falls der Wert nicht angegeben wird, oder 0 ist, ist ein applikationsabhängiger Default-Wert anzunehmen.

3.9.10 AnchorBar

Verankerungsstäbe.

Type: Typ-Kennung.

Length: Länge in mm.

Position: Verlegeposition in mm.

3.9.11 GirderExt

Generische Untertabelle des Gitterträgers.

Die genaue Bedeutung der Felder ist abhängig vom Typ.

Für alle Typen gibt es das **Position**-Feld: dieses beschreibt eine Position in mm, gemessen vom Gitterträger-Anfang.

Das **Flags**-Feld enthält 32 Bits, die typspezifisch und/oder applikationsspezifisch verwendet werden können. Die Bedeutung der Felder **Val0, …, Val3** ist ebenfalls typspezifisch und/oderapplikationsspezifisch.

3.9.11.1 Type = splicePos

Spleißpunkt an dem der Gitterträger bei der Gitterträger-Position gestückelt wurde.

- Val0: Phasensprung in mm am Schweißpunkt. Entspricht der Gitterträger-Länge, die am Spleißpunkt 'ausgelassen' wurde. Der Wert kann auch negativ sein.
- Val1: Überlappungslänge am Spleißpunkt. Entlang dieser Strecke hat man einen doppelten Gitterträger. Der Wert kann auch negativ sein.

3.9.11.2 Type = FixingPos

Punkt an dem der Gitterträger auf der Bewehrung fixiert wird (Verschweißt oder geheftet).

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Die Werte Val0, …, Val03 sind applikationsabhängig. Bei Applikationen, die mit festen Schweiß-Tabellen übergibt man in Val0 das Schweißprogramm. In anderen Applikationen können Schweißstrom, Schweißzeit und Ähnliches angegeben werden.

3.9.11.3 Type = GirderGripPos

Position an der der Gitterträger gegriffen wird um eingelegt zu werden (Grippposition für den Transport des Einzel-Gitterträgers).

3.9.11.4 Type = MeshGripPos

Position an der der Gitterträger gegriffen wird wenn das Element transportiert wird.

3.9.11.5 Type = SupportPos

Position an der ein Auflage-Zwischenstück anzubringen ist.

3.9.11.6 Export nach UNICAM

Die GirderExtRows werden beim Export nach UNICAM als Gitterträgerschweißpunkte eingetragen. Die Schweißleistung enthält die Typ-Information:

- Schweißleistung = **0xx**: *Unknown*.
- Schweißleistung = **1xx**: *SplicePos*.
- Schweißleistung = **2xx**: *FixingPos*.
- Schweißleistung = **3xx**: *GirderGripPos*.
- Schweißleistung = **4xx**: *MeshGripPos*.
- Schweißleistung = **5xx**: *SupportPos*.

Der Wert **'xx'** wird mit ***k = Round(Val0 / 5.0)*** gefüllt. Um sicherzustellen, dass *k* im Bereich 0 bis 100 liegt, wird noch folgende Transformation gemacht: *k* wird auf [-50, 49] beschränkt und bei negativen werten wird 100 hinzuaddiert (d.h. bei einem *k* = -7 wird xx = 93 gesetzt).

3.9.12 Section

Über die *Section*-Tabelle kann das Periodenmuster des Gitterträgers exakt festgelegt werden. Für Eigenschaften, die nicht über *Section*-Einträge definiert werden, gelten die übergeordneten Angaben im *Girder*-Eintrag, bzw. Defaultwerte, falls gar nichts spezifiziert wird.

Ein *Section*-Eintrag beschreibt einen Periodenabschnitt, der immer mit einem Periodentiefpunkt beginnt und mit einem Periodentiefpunkt endet.

3.9.12.1 Felder der Section-Tabelle

- **L**: Länge eines Periodenabschnitts in mm. Typisch sind Werte um 200 mm.
- **S**: Verschiebung des Obergurtschweißpunktes in Bezug auf die Mitte des Periodenabschnitts (in mm). Für einen normalen Gitterträger ist der Wert 0. Für einen Schubträger liegt der Wert typisch um -100.

3.9.12.2 Zuordnung der Section-Angaben

Die Zuordnung der *Section*-Einträge zu den effektiven Gitterträger-Abschnitten erfolgt nach folgenden Regeln:

a) Die *Section*-Einträge beschreiben ein sich wiederholendes Muster. Wenn beispielsweise die 3 *L*-Angaben 195, 195, 200 gemacht werden, bedeutet das, dass die ersten beiden Perioden eine Länge von 195 mm haben, und die dritte Periode eine Länge von 200 mm hat. Die vierte und fünfte Periode haben wieder die Länge 195 mm, usw.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

b) Die *Section*-Angaben beschreiben nur ein Muster; sie sagen nichts über die tatsächliche Länge des Gitterträgers aus. Diese wird im *Length*-Feld des *Girder*-Eintrags festgelegt.

c) Das Periodenmuster beginnt normalerweise beim Gitterträgeranfang. Wenn aber ein von 0 verschiedener *PeriodOffset*-Wert angegeben wird, ist der Beginn des Musters entsprechend verschoben[31].

d) Für die *Section*-Muster-Erstellung werden nur Non-Null-Werte herangezogen. Und die einzelnen *Section*-Felder sind in ihrer Abfolge unabhängig voneinander. Die beiden folgenden *Section*-Tabellen sind demnach exakt äquivalent[32]:

L	**S**
195	-50
195	0
200	-50
195	0
195	-50
200	0

L	**S**
195	
195	
200	
	-50
	0

[31] Beachte: Bei einem positiven Wert von *PeriodOffset* wird der Start des Periodenmusters in den Gitterträger hinein verschoben. Der erste Teil des Gitterträgers liegt dann vor dem Beginn des Periodenmusters. Dieser Teil hat aber dennoch eine definierte Struktur, da sich das Muster periodisch wiederholt, und somit auch vor seinem Startpunkt definiert ist (Vor dem Beginn des Musters reiht sich das das Ende des selben Musters, eben weil das Muster in unendlicher folge wiederholt wird).

[32] Demnach steht es einer Implementierung frei, die für jeden *Section*-Wert eine eigene Tabelle aufzubauen (oder auch eine Tabelle mit einem Wert und einer Feldtyp-Kennung zu verwenden).

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.10 Alloc

Ein *Alloc*-Block beschreibt ein Staffelungsmuster für Eisen oder Gitterträger.

Verwendet wird dies vor allem für die Beschreibung von *Körben*: der *Alloc*-Block beschreibt dort das Verlegungsmuster der Bügel.

Type-Attribut

Es gibt folgende Typen:

- **Bar:** Der *Alloc*-Block bezieht sich auf Eisen (Bügel, Staffeleisen).
- **Girder:** Der *Alloc*-Block bezieht sich auf Gitterträger.

3.10.1 GuidingBar (Leiteisen)

Falls im Feld *GuidingBar* ein gültiger Eisen-Index steht, erfolgt die Staffelung entlang dieses Eisens. Die *Region*-Werte definieren dann Staffelpositionen auf diesem Leiteisen, d.h. die "geleiteten" Eisen werden entlang des Leiteisensan diesen Staffelpositionen angeordnet.

Begriffsdefinition:

- **Leiteisen (guiding bar):** jenes Eisen, das den Pfad für die Staffelung vorgibt.
- **Staffelpositionen (guiding positions):** jene Punkte auf dem Leiteisen, an denen die geleiteten Eisen angeordnet werden.
- **Geleitetes Eisen (guided bar):** jenes Eisen, das vom Leiteisen "geführt" werden soll.
- **Gestaffelte Eisen (allocated bars):** jene Eisen, die an den realen Positionen liegen, die sich aus der der Staffelung ergeben.

Es gelten folgende Regeln:

a) Der vom Leiteisen definierte Pfad verläuft im Eisenkern des Leiteisens und hat bei Biegungen den Biegeradius 0.

b) Für die Ermittlung der Koordinaten der gestaffelten Eisen, addiert man die *relativen Staffelpositions-Koordinaten* zu den Eisenkoordinaten. Die *relativen Staffelpositionen* sind hierbei die Staffel-Positionen auf dem Leiteisen, bezogen auf den Anfang des Leiteisens.
Das originale (ungestaffelte) geleitete Eisen ist also am Startpunkt des Leiteisens zu positionieren.

c) Wenn das Leiteisen Biegungen hat, werden die gestaffelten Eisen zusätzlich gedreht. Gedreht wird <u>nach</u> der Verschiebung, und zwar um den Staffelpositionspunkt. Für jede Leiteisenbiegung, bis zur betreffenden Staffelposition, erfolgt eine Drehung um die Achse der Biegung[33].

3.10.2 Richtungsbestimmung ohne GuidingBar

Falls es keine GuidingBar gibt, wird die Richtung durch das gesteffelte Objekt selbst bestimmt:

Das Eisen bestimmt den Start-Punkt über die entsprechenden X/Y/Z-Koordinaten des *Bar*-Items. Die Richrung wird wie folg festgelegt:

---

[33] Das erste *BendY* des Leiteisens ist hierbei *nicht* als Biegung aufzufassen; d.h. es geht nur um die Biegungen innerhalb des Eisens.
An dieser Stelle mag es verwundern, dass *RotZ* und *BendY0* des Leiteisens nicht in die gestaffelten Eisen eingerechnet werden. Das hat zum einen praktische Gründe, da es recht unanschaulich wäre, wenn das*RotZ* des geleiteten Eisens nicht absolut, sondern relativ zum Leiteisen wäre. Tatsächlich gibt es aber recht fundamentale Gründe, die verhindern *RotZ* und *BendY0* des Leiteisens in die gestaffelten Eisen einzurechnen: *RotZ* und *BendY0* sind für ein gegebenes Leiteisen nicht eindeutig, da es unterschiedliche Kombinationen dieser Werte gibt, die für das Leiteisen äquivalent sind, die aber für die gestaffelten Eisen, wenn sie einberechnet würden, unterschiedliche geometrische Verhältnisse hervorrufen würden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

1) Koordinatensystem um *RotZ* drehen (um die positive z-Achse).
2) Koordinatensystem um $RotX_0$ drehen (Um die positive x-Achse).
3) Koordinatensystem um $BendY_0$ drehen (Um die negative y-Achse).
4) Koordinatensystem um $RotX_1$ drehen (Um die positive x-Achse); dies nur dann, wenn mehr als ein Segment existiert.
5) Die negative y-Achse zeigt dann in die Richtung von positiven Pitch-Werten. Das ist die Achse, über die die erste Biegung definiert wird.

Beachte: in UNICAM ist der Pitch-Wert für Längs- und Quereisen unterschiedlich definiert[34].

- Für Eisen in +X-Richtung (0°): UnicamPitch = – PxmlPitch
- Für Eisen in –X-Richtung (180°): UnicamPitch = PxmlPitch
- Für Eisen in +Y-Richtung (90°): UnicamPitch = PxmlPitch
- Für Eisen in –Y-Richtung (-90°): UnicamPitch = – PxmlPitch
- Für alle anderen Winkel: in UNICAM müssen Einzeleisen angegeben werden.

Die UNICAM-Definition wirkt etwas vertrauter, kann aber nicht konsistent für allgemeine Winkel definiert werden.

Für **Gitterträger** gilt analoges: der Startpunkt wird über die Gitterträger-Koordinaten festgelegt: die Richtung des Abstandes ist wie folgt definiert:

1) Koordinatensystem um AngleToX drehen (um die positive z-Achse).
2) Die negative y-Achse zeigt dann in die Richtung von positiven Pitch-Werten. Das ist die Achse, über die die erste Biegung definiert wird.

Wieder besteht der oben genannte Unterschied zur UNICAM-Definition.

3.10.3 Region

Das Verlegungsmuster besteht aus einer Folge von *Regions*: jede *Region* beinhaltet folgende Informationen:

- **IntervalCount:** Anzahl der Abstände (Intervalle).
  Der Wert ist ganzzahlig und nichtnegativ.
- **Pitch:** Intervallbreite in mm. Der Wert kann auch nicht-ganzzahlig sein, und darf auch negativ sein.
- **IncludeBegin:** Falls gesetzt, gehört der Anfang des Intervalls (und das dort liegende Eisen) zur Region dazu.
- **IncludeEnd:** Falls gesetzt, gehört das Ende des Intervalls zur Region dazu.
- **RefIndex:** Integer-Wert, der das referenzierte Item (Eisen oder Gitterträger) bestimmt. Ein Wert von 0 bedeutet, dass das erste Eisen (oder der erste Gitterträger) des betreffenden *Steel*-Blocks referenziert wird.
  Falls der *RefIndex* NULL, negativ, oder zu groß ist, gilt er als nicht zugewiesen. Die *Region* definiert dann einen Abstandsoffset ohne Zuordnung eines Eisens.

Anmerkungen:

i) Die **Länge** der *Region* ergibt sich aus**L = IntervalCount * Pitch**.
ii) Eine *Region* definiert Positionen
Die Position *k* der *m*-ten Region wird wie folgt berechnet:

$$P_{m,k} = \sum_{i=0}^{m-1} L_i + k \cdot Pitch_m, \quad L_i = IntervalCount_i \cdot Pitch_i .$$

[34] Siehe z.B. Unitechnik-Schnittstellen-Definition 6.0, 4.9.5, Punkt 10: "Positive Teilung bedeutet positive Koordinatenrichtung, …, Die Verwendung der Teilung ist NUR für Winkel von 0°, 90°, 180° und 270° erlaubt."

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Die so berechnete Position ist zunächst nur ein skalarer (eindimensionaler) Wert. Über die Verfahren aus den Abschnitten 3.10.1 und 3.10.2 kann daraus ein konkreter Raum-Punkt bestimmt werden.

iii) Wenn die *Region* kein gültiges*RefIndex* hat, sind ihr keine Bügel zugeordnetwerden[35].

iv) Wenn die *Region* ein gültiges*RefIndex* hat, werden mindestens *IntervalCount-1* Bügel zugeordnet. Wenn *IncludeBegin* gesetzt ist, kommt ein weiter Bügel hinzu. Wenn *IncludeEnd* gesetzt ist, kommt ebenfalls ein Bügel hinzu. Maximal können der *Region* somit *IntervalCount+1* Bügel zugeordnet werden.

v) Üblicherweise wird man*IncludeBegin* einer Region und *IncludeEnd* der vorhergehenden Region synchronisieren. Wirklich frei ist man daher nur am Anfang der ersten und am Ende der letzten *Region*. Diese Synchronisierung ist jedoch nicht zwingend vorgeschrieben.

vi) Aufgrund des oben stehenden Musters werden ergibt ein *Alloc*-Block für einen darin referenzierten Bügel eine Stückzahl. Diese als **AllocCount** zu bezeichnende Stückzahl wird also wie folgt berechnet:

$$AllocCount = \sum_{i=0}^{RegionCount-1} D_i \cdot (IntervalCount_i - 1 + E_i + B_i), \quad AllocCount >= 0$$

$B_i = 1$ falls $IncludeBegin$ gesetzt

$E_i = 1$ falls $IncludeEnd$ gesetzt

$D_i = 1$ falls der betreffende Bügel in $Region_i$ vorkommt, andernfalls 0

Beachte:*AllocCount*-Wert ist per definitionem nichtnegativ, obwohl er rein rechnerisch bei *IntervalCount=0* negativ sein könnte.

Beachte:ein *Alloc*-Blockdefiniert für jeden Bügel, der im *Alloc*-Block referenziert wird, einen eigenen *AllocCount*-Wert.

vii) Die Summe der *AllocCounts* zu einem Bügel gibt an, wie viele Bügel eine zugewiesene Staffelposition haben. Üblicherweise wird diese Anzahl mit der Stückzahl des Bügels übereinstimmen (*Bar.PieceCount*). Diese Übereinstimmung ist aber nicht zwingend vorgeschrieben, d.h. *Bar.PieceCount* kann überzählige oder fehlende Bügel aufweisen.
Falls es **überzählige Bügel** gibt, werden diese ihrer originalen X/Y/Z-Position zugeordnet; diese überzähligen Bügel werden also getrennt von den *Alloc*-Blöcken betrachtet.
Falls es **fehlende Bügel** gibt, werden die letzten Bügel der letzten *Alloc*-Blöcke als fehlend betrachtet.

viii) Eine *Region* mit ***Pitch=0*** kann rechnerisch behandelt werden wie jede andere *Region*. Verwendet wird dieser Sonderfall üblicherweise um Bügel darzustellen, die nicht verschweißt werden, sondern lose mitgeliefert werden (typischerweise am vordere und am hinteren Ende eines Korbes).

ix) Die Definition der *Alloc*- und *Region*-Datenstrukturen lässt es zu, dass ein Eisen in mehreren *Regions* referenziert wird, die auch in unterschiedlichen *Alloc*-Blöcken liegen können. Üblicherweise wird ein Eisen aber nur in *Region*-Blöcken referenziert, die zu einem einzigen *Alloc*-Blöck gehören. Das Eisen hat also üblicherweise höchstens ein Staffelmuster (=*Alloc*-Block). Umgekehrt hat ein Staffelmuster oft mehrere unterschiedliche Eisen.

3.11 SteelExt

Zusatz-Einträge zum Steel-Block. Die Art der Einträge ist applikationsabhängig und wird über das *Type*-Attribut unterschieden. Die Bedeutung des *Info*-Feldes ist abhängig vom *Typ*-Attribut. Zusätzlich wird es üblicherweise weitere applikationsspezifische Felder geben (I_P_-Tags).

[35] Der Einfachheit halber wird hier nur von *Bügeln* gesprochen. Analoges gilt aber auch für andere Staffeleisen und auch für Gitterträger.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.12 Feedback

Die **Feedback**-Tabelle unterscheidet sich ganz wesentlich von den anderen PXML-Tabellen: die *Feedback*-Tabelle dient nicht der Beschreibung von Produktionsdaten, sondern beinhaltet Maschinen-Rückmeldung. Wie im weiter unten noch näher erläutert, gibt es 2 Arten von Rückmeldungen:

- *PTS*-Meldung: informiert vorab über die Produzierbarkeit bestimmter Items.
- *Machine-Return*: informiert über die erfolgte Produktion.

Beim Datentransfer vom CAD zur Maschine wird es üblicherweise keine *Feedback*-Tabelle geben. Beider Rückmeldung der Maschine (oder des Test-Systems) an das CAD wird es häufig nur die *Feedback*-Tabelle geben, und keine weiteren Daten; in speziellen Fällen können aber auch andere PXML-Daten von der Maschine an das CAD- oder Leitsystem zurückgemeldet werden.

3.12.1 Production Test Service (PTS)

Der **Production Test Service** (kurz **PTS**) erlaubt es einem CAD- oder Leitsystem eine Anfrage beim Produktionssystem zu stellen, um die Produzierbarkeit eines Produktes abzufragen. Die Datenerstellung kann dann unter direkter Einbeziehung der Maschinen-Möglichkeiten erfolgen; man erreicht damit eine große Prozesssicherheit und eine große Optimalität des Produktionsablaufs.

Der Bedarf an *PTS*-Systemen ist vor allem im Zusammenhang modernen Mattenbiegeanlagen aufgekommen. Diese Anlagen haben einen immer weiter wachsenden Leistungsumfang, der aber, mit zunehmender technischer Komplexität, nicht mehr in Form von einigen Grenzwerten beschrieben werden kann. Die vollautomatische Produzierbarkeit eines komplexen Bewehrungskorbes hängt von so vielen Details ab, dass sie letztlich nur unter direkter Einbeziehung der Maschinensoftware zuverlässig geprüft werden kann; und genau das soll mit *PTS* erreicht werden.

PTS besteht aus 2 Teilen:

- **PTS-Server:** ist eine Service-Anwendung, die vom Maschinenhersteller bereitgestellt werden muss; dieser Service läuft auf den CAD-Workstations oder irgendwo zentral im Netz. Der *PTS*-Server wartet auf Anfragen und beantwortet diese; er muss dazu die Berechnungslogik und die Parameter der Maschinensoftware berücksichtigen.
- **PTS-Client:** ist vom CAD-Hersteller in der CAD-Anwendung bereitzustellen (und/oder im Leitsystem). Mittels *PTS*-Client werden aus der CAD-Anwendung heraus Anfragen an den *PTS*-Server gestellt. Die Server-Antwort muss dann im CAD-System visualisiert werden.

Die Zuordnung einer Antwort zu einer vorher gesendeten Anforderung erfolgt über die *GlobalID* der *DocInfo*-Tabelle. Diese ID sollte daher für jede Anforderung individuell gesetzt werden.

Die **Anfrage (Request)** des *PTS*-Client an den PTS-Server erfolgt in Form eines PXML-Dokumentes, das Produktionsdaten enthält, also *Order*-Blöcke mit den zugehörigen Child-Strukturen.

Die **Antwort (Feedback)** des *PTS*-Servers erfolgt in Form eines PXML-Dokumentes, das *Feedback*-Blöcke enthält. Wenn der *PTS*-Server Produktionsdaten von sich aus verändert, kann er zusätzlich zu den *Feedback*-Blöcken auch *Order*-Blöcke zurückmelden, um so die veränderten Daten zu beschreiben[36]. Die Rückmeldung der *Order*-Blocke (=Produktionsdaten) ist aber als optionale Zusatzfunktion anzusehen; das zentrale Element der der Rückmeldung sind die *Feedback*-Blöcke.

3.12.2 Machine Return

Machine Return ist eine Funktionalität, die eng verwandt mit der *PTS*-Rückmeldung ist. Im Gegensatz zum *PTS* geht es hierbei aber nicht um eine Test-Rückmeldung, sondern um eine Produkti-

[36] Die zurückgemeldeten Produktionsdaten müssen natürlich wieder über die *GlobalID* auf die Originaldaten Bezug nehmen. Fall ein Item der Originaldaten in der Rückmeldung mehrfach referenziert wird (weil beispielsweise ein originales Eisen in mehrere Positionen aufgetrennt werden musste), muss in allen betreffenden Rückmeldungs-Items die selbe*GlobalID* angegeben werden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

onsrückmeldung der Produktionsmaschinen an das übergeordnete Leitsystem. Im Wesentlichen werden hierbei produzierte Stückzahlen zurückgemeldet.

Falls die Produktionsmaschine eigenständig Produktionsdaten verändert, kann es sinnvoll sein, in der Rückmeldung auch die veränderten Produktionsdaten anzugeben (*Order*-Blöcke samt Child-Struktur). Diese (optionale) Funktionalität kann sogar so weit gehen, dass Daten, die an der Maschine manuell eingegeben werden, auf diesem Wege in das übergeordnete Leitsystem zurückgepflegt werden.

3.12.3 Felder des Feedback-Blocks

3.12.3.1 Feedback-Attribute

1) **ItemType**: Gibt den PXML-Tabellennamen des Items an, auf das sich die Rückmeldung bezieht.
2) **GlobalID**:Gibt an, auf welches Produktionsdaten-Item sich die Rückmeldung bezieht. Um einen *Feedback*-Eintrag eindeutig zuordnen zu können, benötigt man sowohl den *ItemType* als auch die *GlobalID*. Nur beide zusammen identifizieren das betroffenen Produktions-Item.

Beide genannte Attribute sollen pro *Feedback*-Block genau einmal angegeben werden.

3.12.3.2 Feedback-Felder

1) **MessageType**: Definiert den Typ der Meldung. Folgende Werte sind möglich:
   a. **info**: Reine Information, ohne Warnungscharakter. "info" ist auch der Default-*MessageType*, der angenommen wird, wenn es keine Angabe gibt.
   b. **hint**: Hinweis mit kleiner Warn-Stufe. Das wird beispielsweise für Items verwendet, die sich problemlos produzieren lassen, für die aber auf fehlende Optimierung in Bezug auf Produktionsgeschwindigkeit hingewiesen werden soll.
   c. **warning**: Warnung mittlerer Stufe. Das Item kann zwar produziert werden, aber nur unter Schwierigkeiten; eventuell wird es in minderer Qualität oder unter großer Belastung der Maschine produziert.
   d. **error**: Fehler. Das Item kann so nicht produziert werden (Das Item wird entweder gar nicht produziert, oder nur in stark abgeänderter Form).
2) **Code**: Alphanumerischer Code, der die Bedeutung der *Feedback*-Meldung identifiziert. Die Liste von Codes mit der ihnen zugeordneten Bedeutung wird vom Maschinenhersteller bereitgestellt.Bei *PTS*-Meldungen wird empfohlen, über das*Code*-Feld eine eindeutige Fehler-Kodes anzugeben.Im Rahmen von *Machine-Return*-Meldungen kann der *Code* benutzt werden, um Rückmeldungs-Events zu identifizieren. Typisch sind hier Codes wie *imported* (datenmäßig importiert), *started* (Produktion gestartet), *produced* (als Einzelposition produziert; bei Betonelementen: betoniert), *embedded* (eingebaut, verschweißt; bei Betonelementen: eingelagert), *delivered* (ausgeliefert, übergeben an das Folgesystem; bei Betonelementen: ausgelagert/abgehoben).
3) **InfoValue**: Zusatz-Information zur *Feedback*-Meldung. Hier kann ein numerischer oder ein alphanumerischer Wert angegeben werden. Die genaue Bedeutung dieses Feldes hängt vom *Code* ab.
4) **PieceCount**: Stückzahl (Integer-Wert); gibt an, wie viele Stück vom Angegebenen Item zurückgemeldet werden. Typischerweise steht hier 1.
   Dieses Feld wird für *Machine-Return*-Meldungen verwendet, und fehlt in *PTS*-Meldungen normalerweise.
5) **MaterialType**: Typ des verwendeten Materials.
   Bei *Bar*-Items wird hier typischerweise der Drahtdurchmesser eingetragen; falls es zu einem Durchmesser verschiedene Draht-Typen gibt, wird hier eine allgemeinere alphanumeri-

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

sche Draht-ID eingetragen.
Bei Gitterträgern wird typischerweise der Träger-Typ eingetragen.
Dieses Feld wird für *Machine-Return*-Meldungen verwendet, und fehlt in *PTS*-Meldungen normalerweise.

6) **MaterialBatch**: Chargenkennung des eingesetzten Materials.
Für Rundstahl vom Coil wird folgendes Format empfohlen: **"12345@AR177228C"**. Hierbei ist "12345" eine ID, die das Coil eindeutig identifiziert; "AR177228C" ist hier die Stahl-Charge. Da die Coil-Charge aus der ID abgeleitet werden kann, ist es häufig sinnvoll, nur die ID anzugeben (also nur "12345"); wenn keine ID bekannt ist, kann man auch nur die Charge angeben, angeführt vom Trennzeichen '@' (also "@AR177228C").
Dieses Feld wird für *Machine-Return*-Meldungen verwendet, und fehlt in *PTS*-Meldungen normalerweise.

7) **MaterialWeight**: Gewicht des eingesetzten Materials in kg (Gesamtgewicht für alle produzierten Einheiten).
Dieses Feld wird für *Machine-Return*-Meldungen verwendet, und fehlt in *PTS*-Meldungen normalerweise.

8) **ProdDate**: Produktions-Zeitstempel (Ende-Zeitpunkt der Produktion). Wird typischerweise in *Machine-Return*-Meldungen angegeben, um den genauen Produktionszeitpunkt festzuhalten.

9) **Machine**: Identifikation der Maschine oder des PTS-Servers, der die Meldung generiert. Dieses Feld ist insbesondere dann bedeutsam, wenn mehrere PTS-Server hintereinandergeschaltet sind.

10) **Description**: Optionale Texteinträge, um die Meldung in Textform zu beschreiben. Es können mehrere Sprachen angegeben werden. Hierbei ist für das *Culture*-Attribut mittels ISO 639 Language Codes anzugeben, optional erweitert durch die ISO 3166 Country Codes; man hat also Angaben der Form "en" oder "en-US".
Der Description-Text kann auch lang und mehrzeilig sein. Neben der eigentlichen Fehlermeldung kann es auch Korrekturempfehlungen geben, oder Angaben über automatisch durchgeführte Korrekturen.
Beispiel: "Rastermaß Y=50mm nicht eingehalten. Eisen in Y um -25mm verschoben."
Ein *Feedback*-Eintrag sollte pro Sprache nur einen*Description*-Eintrag haben.

In *Machine-Return*-Meldungen, in denen lediglich produzierte Stückzahlen gemeldet werden, wird man den *MessageType* nicht angeben, oder ihn auf "info" setzen. Im Prinzip kann aber eine Produktionsmeldung auch eine Warnung miteinschließen, so dass eine *Machine-Return*-Meldung auch den anderen *MessageType*s kombiniert werden kann.

3.12.4 Beispiele für PTS-Meldungen

Die genaue Liste der möglichen *PTS*-Meldungen wird vom Maschinenhersteller definiert. Die folgende Liste ist daher nur als Beispiel zu verstehen:

Beispiele für ItemType="Segment":

- Maximum segment length exceeded. [error]
- Segment too short. [error]
- Segment as L0 too short. [error]
- Bending too close as L0 too short. [error]
- Invalid bending angle. [error]

Beispiele für ItemType="Bar":

- Invalid bar diameter. [error]
- Bar with too few Welding points. [error]

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

- Bar too close to next bar. [error]
- Bar too close to next bar – corrected. [warning]
- Multiple bars aligned on same y; production will be slow. [hint]

Beispiele für ItemType="Steel":

- Invalid mesh size (too large). [error]
- Invalid mesh size (too small). [error]
- Mesh not transportable. [error]
- Mesh not transportable – corrected. [warning]

Da die Meldungen am CAD-Bildschirm eventuell nicht immer eindeutig visuell zuordenbar sind, sollte der *ItemType* aus dem aus dem *Description*-Text erkennbar sein.

Das folgende Beispiel zeigt das PXML-Dokument einer *PTS*-Rückmeldung:

```
<?xml version="1.0" encoding="utf-8"?>
<PXML_Document xmlns="http://progress-m.com/ProgressXML/Version1">
  <DocInfo GlobalID="7C7E1FC5-0A46-48c5-A3B2-249D75B70BCF">
    <MajorVersion>1</MajorVersion>
    <MinorVersion>2</MinorVersion>
    <Comment>MSystem PTS 3.77.12.123, List 1, Params 2010-07-12</Comment>
  </DocInfo>
  <Feedback ItemType="Bar" GlobalID="12345">
    <MessageType>error</MessageType>
    <Code>MaxBarLen</Code>
    <Description Culture="en" Text="Maximum bar length exceeded."/>
    <Description Culture="de" Text="Max. Eisenlänge überschritten."/>
  </Feedback>
  <Feedback ItemType="Bar" GlobalID="2057">
    <MessageType>warning</MessageType>
    <Code>MinBarLen</Code>
    <Machine>BGM</Machine>
    <Description Culture="en" Text="Bar too short."/>
    <Description Culture="de" Text="Eisen zu kurz."/>
  </Feedback>
</PXML_Document>
```

In obigem Beispiel sieht man 2 *PTS-Feedback*-Meldungen: eine Fehlermeldung für das Eisen "12345", und ein Warnung für das Eisen "2057". In diesem Beispiel sei "MinBarLen" ein Warnungs-Code, der Eisen betrifft, die von der Maschine automatisch verlängert werden; das Eisen kann also in automatisch korrigierte Form produziert werden.

Besonders hervorzuheben ist der *Comment*-Eintrag im *DocInfo*-Teil: hier sollte eine Identifikation des *PTS*-Servers stehen, aus der auch hervorgeht, wann seine Parameter zuletzt mit den Maschinen abgeglichen wurden. Idealerweise sollte der *PTS*-Client diese Information für den Benutzer sichtbar machen.

3.12.5 Beispiele für Machine-Return-Meldungen

Das folgende Beispiel zeigt das PXML-Dokument einer *Machine-Return*-Meldung, in der die Produktion zweier Eisen gemeldet wird.

```
<?xml version="1.0" encoding="utf-8"?>
<PXML_Document xmlns="http://progress-m.com/ProgressXML/Version1">
  <DocInfo GlobalID="7C7E1FC5-0A46-48c5-A3B2-249D75B70BCF">
    <MajorVersion>1</MajorVersion>
    <MinorVersion>2</MinorVersion>
  </DocInfo>
  <Feedback ItemType="Bar" GlobalID="2057">
    <PieceCount>3</PieceCount>
    <MaterialType>16A</MaterialType>
    <MaterialBatch>12345@AR177228C</MaterialBatch>
```

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

```
    <ProdDate>2010-07-30T09:06:03+02:00</ProdDate>
  </Feedback>
  <Feedback ItemType="Bar" GlobalID="2058">
    <PieceCount>1</PieceCount>
    <MaterialType>8</MaterialType>
    <MaterialBatch>12345@AR177228C</MaterialBatch>
    <ProdDate>2010-07-30T09:06:05+02:00</ProdDate>
  </Feedback>
</PXML_Document>
```

Das folgende Beispiel zeigt das PXML-Dokument einer *Machine-Return*-Meldung, in der die Produktion von 3 Stück eines Elementes gemeldet wird; hierbei wird auch angegeben, welches Material verwendet wurde, und wie viele Kg benötigt wurden.

```
<?xml version="1.0" encoding="utf-8"?>
<PXML_Document xmlns="http://progress-m.com/ProgressXML/Version1">
  <DocInfo GlobalID="7C7E1FC5-0A46-48c5-A3B2-249D75B70BCF">
    <MajorVersion>1</MajorVersion>
    <MinorVersion>2</MinorVersion>
  </DocInfo>
  <Feedback ItemType="Slab" GlobalID="12007">
    <PieceCount>3</PieceCount>
    <ProdDate>2010-07-30T09:06:03+02:00</ProdDate>
  </Feedback>
  <Feedback ItemType="Slab" GlobalID="12007">
    <MaterialType>10</MaterialType>
    <MaterialBatch>12345@AR177228C</MaterialBatch>
    <MaterialWeight>123.7</MaterialWeight>
    <ProdDate>2010-07-30T09:06:05+02:00</ProdDate>
  </Feedback>
  <Feedback ItemType="Slab" GlobalID="12007">
    <MaterialType>12</MaterialType>
    <MaterialBatch>12345@AR177228C</MaterialBatch>
    <MaterialWeight>162.4</MaterialWeight>
    <ProdDate>2010-07-30T09:06:05+02:00</ProdDate>
  </Feedback>
  <Feedback ItemType="Slab" GlobalID="12007">
    <MaterialType>KTS810</MaterialType>
    <MaterialWeight>98.7</MaterialWeight>
    <ProdDate>2010-07-30T09:06:05+02:00</ProdDate>
  </Feedback>
</PXML_Document>
```

Die Feedback-Blöcke beziehen sich hier alle auf den selben *Slab*-Eintrag. Der erste Feedback-Block gibt an, dass 3 Stück produziert wurden, die anderen melden Materialverbrauch, der für diesen *Slab*-Eintrag aufgewendet wurde.

Beachte: den Applikationen steht es frei, die Rückmeldungen auf grobe oder feine Hierarchiestufen zu beziehen. So ist es beispielsweise möglich, die Rückmeldung nur für den gesamten *Order*-Eintrag zu machen, also lediglich zurückzumelden, dass der *Order*-Eintrag abgearbeitet wurde, und eventuell wie viel und welches Material hierfür verwendet wurde. Andere Applikationen könnten dagegen Rückmeldungen auf *Bar*-Ebene machen. Wenn man aber verschiedene Hierarchiestufen zurückmeldet, so ist zu beachten, dass **summierbare Größen** (wie das Gewicht) sind immer **akkumulierbar** sein müssen: wenn also beispielsweise *Bar*-Feedbacks mit Gewicht gemeldet werden, und *Slab*-Feedbacks mit Gewicht gemeldet werden, so darf das gemeldete *Slab*-Gewicht nur jenes Zusatzgewicht beschreiben, das nicht in *Bar*-Feedbacks enthalten ist.

3.12.6 Kommunikations-Arten des PTS

Für die Kommunikation zwischen *PTS*-Client und *PTS*-Server gibt es viele Möglichkeiten. Jedes *PTS*-Server-System sollte aber mindestens die im folgenden angeführten Möglichkeiten anbieten.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

3.12.6.1 Kommunikation auf File-Basis

Client und Server kommunizieren über zwei Verzeichnisse, ein **Request**-Verzeichnis und ein **Feedback**-Verzeichnis. Der Server beobachtet das *Request*-Verzeichnis; sobald dort eine Datei liegt, wird die Datei gelesen und gelöscht, und im *Feedback*-Verzeichnis wird eine Antwort-Datei bereitgestellt, die den selben Namen hat, wie die Anfrage-Datei.

Der Server bietet die Möglichkeit beliebig viele *Request/Feedback*-Verzeichnis-Paare zu definieren, so dass es für jeden Client ein eigenes Verzeichnis-Paar gibt.

Jedes *Request/Feedback*-Verzeichnis-Paar entspricht am *PTS*-Server auch einem Parameter-Block (Produktionsliste, Produktionsmodus[37], usw.); es kann daher vorkommen, dass ein *PTS*-Client unterschiedliche *Request/Feedback*-Verzeichnis-Paare anspricht.

3.12.6.2 Kommunikation über TCP/IP-Sockets

Die Kommunikation erfolgt über einen TCP-Port. Defaultmäßig ist das der Port 56693; andere Port-Werte sollen aber ebenfalls eingestellt werden können. Der Verbindungsaufbau wird vom Client initiiert; ebenso der Verbindungsabbau. Dem Client steht es frei, eine dauerhaft aktive Verbindung zu verwenden, oder für jede Anfrage eine neue Verbindung aufzubauen.

Unterschiedliche Server haben unterschiedliche IP- oder Portnummern. Wenn an einem Server unterschiedliche Parameter-Blöcke angesprochen werden sollen, so erfolgt das ebenfalls über unterschiedliche Port-Nummern.

Anfrage und Antwort-Stream enthalten je ein PXML_Document. Über die *GlobalID* der *DocInfo*-Tabelle kann der Feedback-Stream eindeutig einer Anfrage zugeordnet werden.

3.12.7 Kommunikations-Arten des Machine Return

3.12.7.1 Kommunikation auf File-Basis

Die Maschine schreibt fortlaufend *PXML*-Dateien, die übergeordneten Leitsystem gelesen und optional gelöscht werden.

Um den lesenden System die Auswertung zu erleichtern, sollten die Dateinamen die Sequenz der geschriebenen Dateien wiedergeben (entweder durch einen Timestamp im Dateinamen oder durch eine fortlaufende ID im Dateinamen).

Der Maschine steht es frei, für jede *Feedback*-Meldung eine eigene Datei zu generieren, oder mehrere Feedback-Meldungen in eine Datei zu packen. Eine Datei, die einmal geschrieben wurde darf aber nachträglich nicht mehr verändert werden. D.h. die Maschine darf eine bestehende Datei nicht neuerlich öffnen, etwa um weitere *Feedback*-Blöcke anzuhängen. In diesem Punkt unterscheidet sich dieses Konzept ganz klar von den Regeln der Unitechnik-Report-Files.

3.12.7.2 Kommunikation über TCP/IP-Sockets

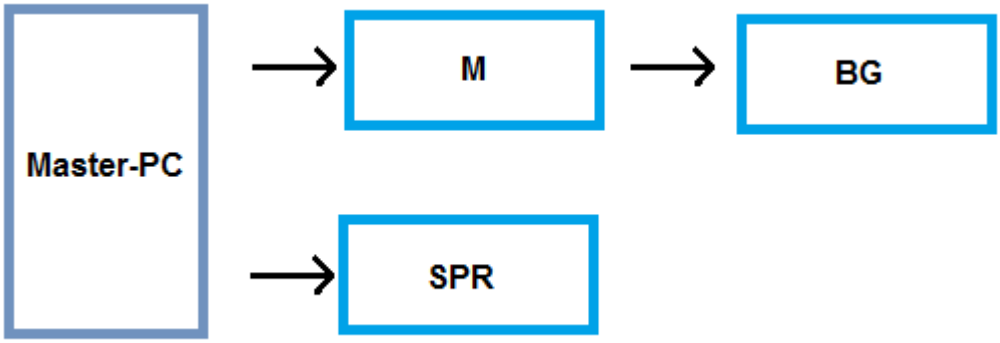
Die Machine-Return-Kommunikation über Sockets ist derzeit nicht verbindlich definiert.

3.12.1 Parallel- und Reihenschaltung von PTS-Servers

Eine komplexe Produktionsanlage kann unter Umständen mehrere PTS-Server haben. So könnte es beispielsweise eine MattenschweißanlageM mit nachfolgender Mattenbiegeanlage BG geben, und parallel dazu einen Schalungsroboter SPR:

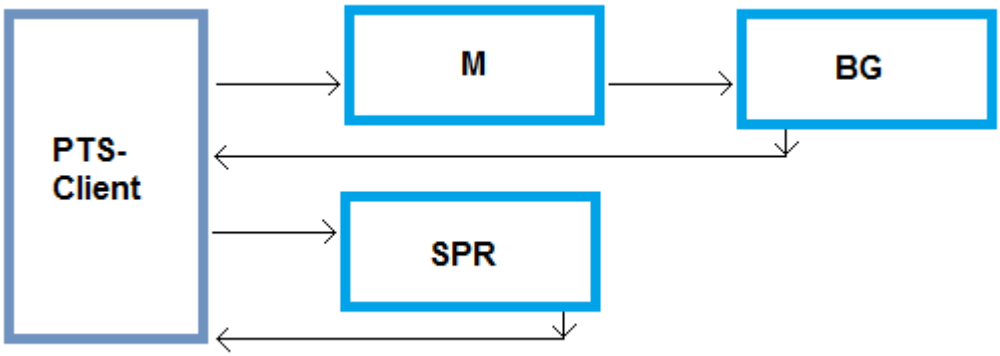
[37] Ein Beispiel einer *Produktionsmodus*-Variante ist folgendes: manche Client-Systeme wollen bei der PTS-Anfrage, dass die Maschinen als zu 100% verfügbar betrachtet werden (diese Anforderung ist für CAD-PST-Clients typisch); andere PTS-Clients (z.B. Taktplatzansichten von Leitrechner) wollen dagegen, dass die Maschinen mit ihrem aktuellen Betriebszustand berücksichtigt werden, dass also beispielweise ein deaktivierter Biegekopf berücksichtigt wird.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.



Die Maschine M und die Maschine BG sind hintereinandergeschaltet, in dem Sinne, dass BG von M mit Daten versorgt wird. Das Leitsystem ("Master-PC") schickt Daten parallel zu M und zu SPR.

Die 3 Maschinen werden durch 2 parallele PTS-Server abgebildet:



Ein PTS-Client sollte in diesem Sinne in der Lage sein, mehrere parallele PTS-Server abzufragen.

Die Kombination der Maschinen M/BG stellt dagegen nach außen nur ein PTS-System dar: die Weitergabe der PTS-Feedback-Datei von M nach BG ist von außen nicht erkennbar. Diese Verkettung ist nötig, da M die Daten eventuell geringfügig korrigiert, und BG mit den korrigierten Daten weiterarbeiten muss. Demgegenüber arbeitet SPR ganz unabhängig von M.

3.12.2 Filtern, klassifizieren und sortieren von PTS-Meldungen

Der PTS-Server nimmt nur eine grobe Klassifizierung der Meldungen vor, indem er sie in Fehler, Warnungen, Hinweise und Informationen einteilt (Siehe Abschnitt 3.12.3.2). Da der Server im Prinzip mehrere Clients bedienen kann, die unter Umständen unterschiedliche Aufgaben und Konzepte verfolgen, ist es nicht Aufgabe des Servers, PTS-Meldungen zu filtern, zu sortieren oder genauer zu klassifizieren. Der PTS-Server meldet alles aus Sicht der Maschine: er sagt, was produzierbar ist, was nicht, und was in eingeschränkter Form – er bewertet nicht, welche dieser Meldungen wichtig oder unwichtig sind.

Es ist Aufgabe des Clients, Möglichkeiten vorzusehen, PTS-Meldungen geeignet darzustellen. Dazu benötigt der Client Mechanismen, um PTS-Meldungen zu filtern und zu sortieren. Idealerweise gibt es auch die Möglichkeit, bestimmte Meldungen besonders hervorzuheben.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

4 Vorschläge für zukünftige Erweiterungen

4.1 Erweiterung der Shapes auf 3D – Differenz-Methode

Beim *SVertex* soll es folgende Zusatzfelder geben: DX, DY. Die Vertex-Koordinate ist dann bei Z=0 durch (X, Y) gegeben, wie bisher; am oberen Z-Ende ist die Vertex-Koordinate aber durch (X+DX, Y+DY) gegeben. Die *Shapes* beschreiben dann nicht mehr Quader, sondern allgemeinere Formen, da die nach oben gehenden Kanten nicht mehr parallel zu Z sein müssen.

Wenn DX und DY nicht angegeben werden, oder einen Wert von 0 haben, hat man wieder normale Quader. Die Erweiterung ist daher voll rückwärtskompatibel.

Anwendung:

- Dämmplatten mit schrägen Kanten.

Kritik:

- Wenn zwei benachbarte "Vertex-Kanten" nicht parallel sind, sonder windschief, ist die dazwischen liegende Fläche nicht eben.

4.2 Drehung für Outline

Die Outlines könnten 3 optionale Dreh-Winkel haben, die von der Geometrie analog zu den ProdRot-Winkeln definiert sind.

Anwendung:

- ComBars könnten korrekt dargestellt werden.
- Man könnte das gesamte Element Rotieren.

Kritik:

- Um das gesamte Element rotieren u können, müssten auch die GTs mit optionalen Rotationswinkeln versehen werden.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

5 Versions-History

Version 1.1:

Neue Felder im *Bar*-Abschnitt

- **Bin:** Fach

Version 1.2:

Neue Felder im *Bar*-Abschnitt

- **Pos:** Position
- **Note:** Bemerkung
- **Machine:** Maschinenzuweisung
- **BendingDevice:** Biegevorrrichtung
- **NoAutoProd:** Ersetzt das *AutomaticProduction*-Feld (ist aber invertiert)

Neue Felder im *Girder*-Abschnitt

- **NoAutoProd:** Ersetzt das *AutomaticProduction*-Feld (ist aber invertiert)

Felder im *Slab*-Abschnitt wurden entfernt

- LateralFormworkType
- LongitudinalFormworkType
- PlasterThicknessBottom
- PlasterQualityBottom
- UnitWeightBottom
- PlasterVolumeBottom
- PlasterThicknessTop
- PlasterQualityTop
- UnitWeightTop
- PlasterVolumeTop

Neue *Outline*-Felder

- **Z**-Koordinate. Ersetzt *MountPartInstallHeight*
- **Height**. Ersetzt *MountPartThickness*.

Entfernte *Outline*-Felder

- MountPartThickness
- MountPartInstallPosX
- MountPartInstallPosY
- MountPartInstallHeight
- Area

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Neuer *Outline*-Typ: *lot*

Dient zur Darstellung von Beton-Parzellen. Ersetzt auch die Typen *contout* und *cutout*.

Neues *Shape*-Feld: *Cutout*

Wenn gesetzt, ist das entsprechende Shape-Polygon eine Aussparung

Entfernen der Concrete-Layer-Elemente

Es gibt keine Concrete-Layer-Elemente mehr, da die Betonschicht-Eigenschaften in der *lot*-Outline enthalten ist. Beim Export nach UNICAM muss eine globale Betonschicht angegeben werden; hier wird die Information des ersten *lot*-Eintrags noch mal koiert.

Der *LayerType* (Kennzeichen der Schicht) entfällt und wird nicht mehr unterstützt.

Entfernen der *contour*- und *cutout*-Outline-Typen.

Diese Outline-Typen werden nicht mehr benötigt, da durch *lot*-Outline ersetzt.

Wenn es in UNICAM eine Gesamt-Kontur mit mehreren Betonschichten gibt, so wird dies in einzelne flächenkongruente *lot*-Elemente zerlegt.

Neu-Definition der Multi-Layer-Elemente.

Einführung des **MasterLayer**-Flags; Neudefinition des Multi-Layer-Konzepts in PXML

Default-Werte sind für jeden Datentyp einheitlich festgelegt

Alle individuellen Defaultwert-Festsetzungen sind damit außer Kraft gesetzt.

Version 1.2 – Later Additions

Neue Felder im *Steel*-Abschnitt (Added on 2007-08-14)

- **Steel.WeldingDensity:** Schweißdichte.
- **Steel.BorderStrength:** Stärke des Matten-Randes.

Neues Feld im Bar-Abschnitt (Added on 2007-09-25)

- **Bar.Ident:** Numerische ID für Systemweite Identifikation.

Neue Felderim Steel-Abschnitt (Added on 2007-10-03)

- **Steel.GenericInfo03…06:** Freie Informationsblöcke (bisher gab es nur 2 Info-Blöcke).

Neuer Steel-Typ (Added on 2007-10-13)

- **Steel.Type = "cage":** Korb für Korbfertigungsanlagen.

Alloc-Tabelle eingeführt (Added on 2007-11-17)

- **Tabellen "Alloc" und "Region":** Verlegungs-Muster für Bügel, Staffeleisen und Gitterträger.

SteelExt-Tabelle eingeführt (Added on 2007-12-05)

- **Tabelle "SteelExt":** Collection von applikationsspezifischen Zusatz-Daten zum Steel-Block.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Neues Feld im Girder-Abschnitt (Added on 2008-07-21)

- **Girder.Machine:** Text-Feld für Maschinen- und Listen-Angabe.

Neue Felderim Girder-Abschnitt (Added on 2008-11-28)

- **Girder.Period:** Gitterträger-Periode.
- **Girder.PeriodOffset:** Abstand der des ersten Perioden-Tiefpunktes vom Gitterträger-Anfang.

GirderExt-Tabelle eingeführt (Added on 2008-12-12)

- **Tabelle "GirderExt":** Generische Untertabelle zur GirderRow.

Neue Felderim Girder-Abschnitt (Added on 2009-02-18)

- **Girder.GirderType:** Bedeutung dieses Feldes genauer festgelegt. Schubträger-Typ definiert.
- **Girder.MountingTyp:** Einbau-Art für Gitterträger.

Neues Feld im Girder-Abschnitt (Added on 2009-03-10)

- **Girder.BottomFlangeDiameter:** Untergurt-Durchmesser in mm.

Neue Felderim Girder-Abschnitt (Added on 2009-03-19)

- **Girder.TopExcess, Girder.BottomExcess:** Überstand des Perioden-Eisens gegenüber Ober- und Untergurt.

Export nach UNICAM für GirderExtRows (Added on 2009-04-01)

- Der Export erfolgt über Gitterträgerschweißzeilen.

Neues Feld im Product-Abschnitt (Added on 2010-02-11)

- **Product.TurnWidth:** Angenommene Palettenbreite für Doppelwände.

Neues Feld im Order-Abschnitt (Added on 2010-06-15)

- **Order.DeliveryDate:** Lieferdatum.

Global ID eingeführt (Added on 2010-07-29)

- **<Table>.GlobalID:**Systemübergreifende ID, die in allen PXML-Tabellen.
- **Bar.Ident** ist nun obsolet, und wurde aus dem Standard entfernt.

ProdRot eingeführt (Added on 2010-07-29)

- **Steel.ProdRotX/Y/Z:** Dreh-Empfehlung für Bewehrungsproduktion hinzugefügt.
- **Slab.AngelOfRotation:** ist nun obsolet, und wurde aus dem Standard entfernt.

PTS-Spezifikation (Added on 2010-07-29)

- **Production Test Service:** Rückmeldung vom Prüfsystem.
- **Machine Return:** Rückmeldung von der Maschine.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Klarstellung zum Character-Encoding (Added on 2010-07-29)

- **Character-Encoding:** Es gelten die gängigen XML-Encoding-Regeln.

Erweiterungsvorschläge (Added on 2010-07-29)

- Es wurde ein eigenes Kapitel hinzugefügt, das eventuelle zukünftige Erweiterungen enthält.

Neue MeshType-Werte (Added on 2010-07-29)

- 5 = Deckelmatte 2D; wie Typ '1', wird aber zusammen mit Typ '4' ausgeliefert.
- 6 = Deckelmatte 3D; wie Typ '3', wird aber zusammen mit Typ '4' ausgeliefert.

Neue Felderim Girder-Abschnitt (Added on 2010-08-18)

- **Girder.Width:** Trägerbreite in mm, zwischen den zwei äußersten Punkten.

Neue Felderim Feedback-Abschnitt (Added on 2010-11-25)

- **MaterialWeight:** Gewicht des gemeldeten Materials.

Neue Feld im Slab-Abschnitt (Added on 2011-01-31)

- **ExpositionClass:** Expositionsklasse.

Neue Feld im Bar-Abschnitt (Added on 2011-03-16)

- **ShapeMode:** Darstellungsart der Biegeform.

Mit der Einführung dieses Feldes wurde auch die Thematik der "schematischen" Darstellung überarbeitet. Diese bislang nur unscharf definierte Möglichkeit wurde präzisiert und vereinfacht und hierbei auch an die Realität bestehender Implementierungen angepasst.

Erweiterung der Definition des Steel.MeshType (Added on 2011-04-22)

Neben der eigentlichen Typ-Angabe soll es auch möglich sein, typspezifische Details anzugeben.

PXML-Spezifikation nun auch in Englisch verfügbar (2011-06-03)

Neues Feld im Feedback-abschnitt (Added on 2011-10-04)

- **InfoValue:**Zusätzlicher Informations-Wert.

Neues Feld im DocInfo-Abschnitt (Added on 2012-02-02)

- **ConvertConventions:** Für deklarierte Abweichungen vom PXML-Standard.

Vereinfachung des Multilayer-Konzeptes (2012-02-08)

- **Outline.Layer:**Wurde hinzugefügt.
- **Steel.Layer:**Wurde hinzugefügt.
- **Slab.MasterLevel:**Wurde entfernt.

Parallel- und Reihenschaltung von PTS-Servers (2012-09-14)

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.

Neues Feld im Feedback-Abschnitt (Added on 2012-11-29)

- **Machine:** Maschine oder PTS-Server der die Meldung generiert hat.

Ergänzungen zur Konvertierung von und zu BVBS (2013-06-10)

Erweiterungen für Gitterträger mit variabler Rasterbreite (Added on 2013-11-10)

- **PeriodOffset:** Ist nun nicht mehr auf Werte zwischen 0 und 200 eingeschränkt.
- **Section**: Neue Untertabelle für individuelle Perioden.

Mode-Tabelle hinzugefügt (2013-22-25)

Neues Feld im Product-Abschnitt (Added on 2013-11-27)

- **RotationPosition**.

Copyright © 2005-2013 Progress Maschinen & Automation AG. Alle Rechte vorbehalten.