

Ein Produktmodell für den Komplettbau

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der Fakultät für

Bauingenieur- und Vermessungswesen

der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

DISSERTATION

von

Dipl.-Ing. Christoph Hörenbaum

aus Lippstadt

Tag der mündlichen Prüfung: 19. Juni 2002

Hauptreferent: Prof. Dr.-Ing. Helmut Saal

Korreferent: Prof. Dr.-Ing. Georg Pegels

Karlsruhe 2002

DANKSAGUNG

Diese Dissertation ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Stahl- und Leichtmetallbau der Universität Karlsruhe (TH) entstanden.

Besonderer Dank gebührt dafür in erster Linie meinem Doktorvater Herrn Univ.-Prof. Dr.-Ing. Helmut Saal. Durch seinen Vorschlag dieses hochinteressanten Themas und seine Unterstützung wurde ein Forschungsgebiet aus der Bauinformatik an einem traditionell ingenieurtechnisch geprägten Institut fortgeführt. Dieser Hintergrund war zugleich Anspruch und Chance für mich, der Arbeit einen wirklichen Praxisbezug im Bauwesen zu geben.

Herrn Univ.-Prof. Dr.-Ing. Georg Pegels danke ich ganz herzlich für die Übernahme des Korreferats. Er brachte vielfältige Anregungen aus dem Fachgebiet der Bauinformatik ein, ohne jedoch die erwähnte praktische Ausrichtung der Arbeit aus den Augen zu lassen.

Hilfestellung bei der Einarbeitung in das Thema der Produktmodellierung erhielt ich von meinem Vorgänger Herrn Dr.-Ing. Andreas Klinkenberg und Herrn Dr.-Ing. Hans-Walter Haller, der dieses Thema an der Universität Stuttgart bearbeitet hat.

Herr Dr. Thomas Liebich von der IAI hat mir die Welt der Industry Foundation Classes eröffnet und mir damit die Möglichkeit gegeben, an die internationale Produktmodellnormung anzuknüpfen.

Die Ansprechpartner in den folgenden Bau- und Software-Unternehmen sowie Ingenieurbüros haben mich durch wertvolle Information und Beratung unterstützt: BO-CAD Software GmbH (Bochum), Corus Bausysteme GmbH (Koblenz), Design Systems Consult GmbH (Karlsruhe), Donges Stahlbau GmbH (Darmstadt), DSD Dillinger Stahlbau GmbH (Saarlouis), Freyler Industriebau GmbH (Kenzingen), Friedrich + Lochner GmbH (Stuttgart), Goldbeckbau GmbH (Bielefeld), Haller Industriebau GmbH (Schwenningen), Herkon Herbolzheimer Konstruktion GmbH (Herbolzheim), IFL Ingenieurbüro für Leichtbau, Ing. Software Dlubal GmbH (Tiefenbach), KIWI Software GmbH (Kreuztal), Maurer Söhne GmbH & Co. KG (München), Mg Engineering Stahlbau Plauen (Plauen), SCIA Group nv (Herk-de-Stad, Belgien), Schüngel – Bau GmbH (Altenburg), Serena Fassaden- und Dämmtechnik GmbH (Taucha), Stahlbau Schäfer GmbH (Mannheim), Tekla GmbH Germany (Eschborn), Vollack Ingenieur- und Stahlbau GmbH & Co. (Mihla) und Vollack Stahltechnik GmbH & Co. (Karlsruhe). Hier sind besonders die Bemühungen von Herrn Dr.-Ing. Michael Falck, Herrn Dipl.-Ing. Rainer Holz, Herrn Dr.-Ing. Michael Huhn, Herrn Dr.-Ing. Peter Knödel und Herrn Dr.-Ing. Dieter Koch zu nennen.

Nicht zuletzt möchte ich die Kollegen und studentischen Hilfskräfte am Lehrstuhl für Stahl- und Leichtmetallbau und an der Versuchsanstalt für Stahl, Holz und Steine erwähnen, die mich in vielfältiger Weise unterstützt haben.

Ihnen allen gebührt mein herzlicher Dank – ohne ihre Hilfe würde meine Dissertation nicht in dieser Form vorliegen.

Ganz besonders bedanke ich mich schließlich bei meiner Freundin Susy Schafheutle für die moralische Unterstützung und die Korrektur des Manuskriptes.

Diese Arbeit ist aus dem Forschungsprojekt Nr. 12386 N „Weiterentwicklung der Produktschnittstelle Stahlbau für die Anforderungen im Komplettbau“ der Arbeitsgemeinschaft industrieller Forschungsvereinigungen „Otto von Guericke“ e.V. (AiF) entstanden, das mit Mitteln des Bundesministeriums für Wirtschaft (BMWi) finanziell gefördert wurde.

Karlsruhe, im August 2002

Christoph Hörenbaum

KURZFASSUNG

Ein Produktmodell ermöglicht den Datenaustausch zwischen Anwendungsprogrammen. Herkömmlichen EDV-Schnittstellen ist es durch seinen ganzheitlichen Ansatz überlegen: Einerseits ist es nicht auf enge Anwendungsgebiete beschränkt, sondern kann den gesamten Lebenszyklus eines Produktes, beispielsweise eines Bauwerkes, abdecken. Andererseits beschreibt es nicht lediglich bestimmte Eigenschaften des Bauwerkes, wie zum Beispiel dessen Geometrie, sondern behandelt die Bestandteile des Bauwerkes als ganzheitliche Objekte.

Für das Bauwesen existieren Produktmodellstandards im Bereich der Ingenieurplanung zur Konstruktion und statischen Berechnung von Tragwerken in Stahl- und Holzbauweise sowie im Bereich der Architekturplanung. Die Entwicklungen in diesen beiden Bereichen wurden bislang unabhängig voneinander vorangetrieben.

In dieser Arbeit wird ein moderner Produktmodellstandard zum Datenaustausch für die Komplettbauplanung im Stahlbau entwickelt. Dies beinhaltet die Konstruktion der das Stahltragwerk ergänzenden Gewerke für Dach- und Außenwandbekleidung in Elementbauweise, für den Ausbau von deren Öffnungen, für den Glasbau sowie die Schalplanung von grundlegenden Elementen des Stahlbetonbaus. Das Anwendungsgebiet des Modells betrifft nicht nur die unmittelbar an diesen Gewerken arbeitenden Konstrukteure, sondern auch Architekten und Planer anderer Ingenieurdisziplinen, beispielsweise der Bauphysik. Deshalb wird durch diese Arbeit nicht nur erstmals die Möglichkeit geschaffen, Ingenieurprogramme des Komplettbaus miteinander zu koppeln, sondern auch mit Programmen der Architekturplanung. Die Anforderungen an das Produktmodell sind dementsprechend vielfältig.

Als Voruntersuchung werden die neuesten Technologien und Standards für die Spezifikation von Produktmodellen sowie der Entwicklungsstand der Software im Bauwesen betrachtet. Um bestehende Anbindungen von kommerzieller Software an die oben erwähnten, für das Bauwesen relevanten Produktmodelle und gegebenenfalls darin erprobte Ansätze zu nutzen, werden diese Modelle daraufhin untersucht, ob und wie das in dieser Arbeit zu entwickelnde Produktmodell mit ihnen harmonisiert werden kann. Außerdem werden die Anforderungen an das Modell aus Sicht der beteiligten Planer festgelegt. Dies betrifft die zu realisierenden Datenaustauschwege und die zu modellierenden Dateninhalte.

Als Kernstück des Produktmodells wird ein Datenmodell in der formalen Sprache EXPRESS spezifiziert. Es ist nicht auf bestimmte Austauschszenarien festgelegt. Die für konkrete Austauschszenarien notwendigen Randbedingungen werden separat davon definiert. Diese Konstellation hat im Hinblick auf das breite Anwendungsgebiet des Modells den Vorteil, daß Randbedingungen für weitere Austauschszenarien bei Bedarf nachträglich aufgestellt werden können. Das Modell wird als Erweiterung der *Industry Foundation Classes* (IFC) definiert, einem Produktmodell, dessen geplantes Anwendungsgebiet das gesamte Bauwesen umfaßt.

ABSTRACT

A product model facilitates data exchange between application software. It is superior to conventional interfaces because of its global approach: On the one hand its scope is not necessarily limited to specific fields of application, but can cover the whole life cycle of a product, for example a building. On the other hand, a product model does not only represent certain properties of the building, like for instance its geometry, but treats the building's elements as entire objects.

Regarding the building industry, there are product model standards in the field of engineering for construction and structural analysis of steel and timber structures as well as in the field of architectural planning. Up to now the developments within these two fields have been carried out independently.

In this thesis an up-to-date product model standard is developed for exchanging data of the supplementary constructions „around“ a building's load-bearing steel structure. This comprises constructional data for roofing and walling with prefabricated elements, for openings and glazing as well as data for the layout of forms of basic concrete elements. The application of the model does not only concern the immediate designers of these constructions, but also architects and engineers of other disciplines, for instance building physics. Thus, not only the possibility of coupling engineering software for these constructions with each other is created, but also with software for architectural planning. Correspondingly, the requirements of the product model are manifold.

As a preliminary investigation, the latest technologies and standards for specifying product models as well as the state of software for the building industry are regarded. In order to make use of existing implementations of the above mentioned product models in commercial software, these models are examined with regard to whether the product model developed in this thesis can be harmonised with them. Furthermore, the requirements from the point of view of the user are constituted. This concerns the ways of data exchange to be realised and the data contents to be modelled.

A data model is specified in the formal language EXPRESS as the core piece of the product model. It is not fixedly orientated towards rigid exchange scenarios. Instead, boundary conditions for specific exchange scenarios are defined separately. This constellation is appropriate for the model's wide field of application, since boundary conditions for additional exchange scenarios can be defined afterwards if required. The product model is specified as an extension of the *Industry Foundation Classes* (IFC). The IFC are a product model the planned scope of which is the entire building industry.

INHALTSVERZEICHNIS

1	Einleitung	
1.1	Problemstellung	1
1.2	Ziel dieser Arbeit	3
1.3	Vorgehensweise und Gliederung	4
2	Grundlagen der Produktmodellierung	
2.1	Prinzip des Produktmodells	
2.1.1	Grundlegendes	5
2.1.2	Modellierungsarten	6
2.1.3	Modellarchitektur	7
2.2	Standardisierung nach ISO 10303	
2.2.1	Allgemeines	8
2.2.2	Datenmodellierung (Description Methods)	9
2.2.3	Implementierung (Implementation Methods)	10
2.2.4	Konformitätsprüfung (Conformance Testing)	12
2.2.5	Gemeinsame Ressourcen (Common Resources)	13
2.2.6	Produktmodelle in ISO 10303 (Application Protocols)	13
2.3	Datenmodellierung mit XML	14
2.4	Bewertung	16
3	Software und Produktdatenaustausch im Bauwesen	
3.1	Allgemeines	19
3.2	Software im Bauwesen	
3.2.1	Allgemeines	19
3.2.2	Stand der Technik	20
3.2.3	Ausblick	24
3.3	Produktmodelle des Bauwesens	
3.3.1	Allgemeines	26
3.3.2	Application Protocol 225 (AP 225)	26
3.3.3	Industry Foundation Classes (IFC)	27
3.3.4	CIMSteel Integration Standards (CIS)	29
3.3.5	Produktschnittstelle Stahlbau	31
3.3.6	Produktmodell Datentransfer im Holzbau (DtH)	33
3.4	Modelle zur Unterstützung des Datenaustausches	
3.4.1	Allgemeines	34
3.4.2	Prozeßmodelle	34
3.4.3	Zentrale Projektdatenbanken	34
3.4.4	Verteilte Systeme	35
3.4.5	Kommunikation über das Internet	35
3.5	Anwendung der Modelle in der Baupraxis	35
3.6	Fazit	37

4	Ingenieurgemäße Anforderungen	
4.1	Allgemeines	39
4.2	Anwendungsprozesse	
4.2.1	Allgemeines	39
4.2.2	Gewerkeübergreifende Betrachtung	40
4.2.3	Dach und Wand	41
4.2.4	Dach- und Wandöffnungen	42
4.2.5	Stahlbetonbau	43
4.2.6	Glasbau	44
4.2.7	Besondere Anforderungen	45
4.3	Technische Beschreibung der Gewerke	
4.3.1	Allgemeines	46
4.3.2	Gewerkeübergreifendes Modell	46
4.3.3	Dach und Wand	
4.3.3.1	Bauarten und Anwendungen	46
4.3.3.2	Bauteile	48
4.3.3.3	Bemessung	52
4.3.3.4	Besondere Anforderungen	52
4.3.4	Dach- und Wandöffnungen	
4.3.4.1	Allgemeines	52
4.3.4.2	Fenster und Fenstertüren	53
4.3.4.3	Türen	53
4.3.4.4	Industrietore	54
4.3.4.5	Dachbelichtung	54
4.3.4.6	Rauch- und Wärmeabzugsanlagen (RWA)	55
4.3.5	Stahlbetonbau	55
4.3.6	Glasbau	
4.3.6.1	Allgemeines	56
4.3.6.2	Bauteile	57
4.3.6.3	Bemessung	60
4.3.6.4	Besondere Anforderungen	60
4.4	Zum Nutzen eines mechanischen Teilmodells	61
4.5	Fazit	62
5	Das Datenmodell der Industry Foundation Classes	
5.1	Allgemeines	63
5.2	Vereinbarungen	
5.2.1	Terminologie	63
5.2.2	Top-Down-Ansatz	64
5.2.3	Vererbungsregeln	64
5.2.4	Verweisregeln	64
5.2.5	Notation	64
5.3	Ressourcen	
5.3.1	Allgemeines	66
5.3.2	Maßeinheiten	66
5.3.3	Geometrie	66
5.3.4	Materialdefinition	67

5.4	Generische Datenstrukturen	
5.4.1	Allgemeines	67
5.4.2	Objekte	68
5.4.3	Relationen	69
5.4.4	Eigenschaftsdefinitionen	71
5.5	Konkrete Datenstrukturen	
5.5.1	Allgemeines	74
5.5.2	Projektstruktur	75
5.5.3	Bauwerkstruktur	75
5.5.4	Bauelemente der IFC2x	76
5.5.5	Bauelemente des ST-4-Projektes	78
5.6	Bewertung hinsichtlich des Komplettbaus	80
6	Datenmodell Komplettbau	
6.1	Allgemeines	83
6.2	Vorüberlegungen	
6.2.1	Leitgedanken für die Modellierung	83
6.2.2	Zentrale Begriffe	84
6.2.3	Notation	85
6.2.4	Übersicht und Einordnung in die IFC2x	85
6.3	Grundlegende Datenstrukturen	
6.3.1	Profilformen	87
6.3.2	Bauprodukte	88
6.3.3	Befestigungen	90
6.3.4	Allgemeines Bauteil: <i>IfcConstructionalElement</i>	90
6.3.5	Auflösung von Bauelementen in Bauteile	91
6.4	Gewerkeübergreifendes Modell	
6.4.1	Grundzüge der Darstellung	92
6.4.2	Zusammengesetzte Wände	
6.4.2.1	Bauelement: <i>IfcWallIsDecomposed</i>	95
6.4.2.2	Anforderungen: <i>Pset_RoofAndWallRequirements</i>	95
6.4.2.3	Beispiel	95
6.4.3	Glasfassaden	
6.4.3.1	Bauelement: <i>IfcGlassFacade</i>	96
6.4.3.2	Bauelement: <i>IfcGlassFace</i>	97
6.4.3.3	Anforderungen: <i>Pset_GlassFaceRequirements</i>	97
6.4.3.4	Beispiel	97
6.4.4	Fundamente: <i>IfcShallowFoundation</i>	98
6.4.5	Fensteranordnungen	
6.4.5.1	Bauelement: <i>IfcWindowArrangement</i>	99
6.4.5.2	Beispiel	99
6.4.6	Industrietore	
6.4.6.1	Bauelement: <i>IfcDoorIndustrial</i>	100
6.4.6.2	Typdefinition: <i>IfcDoorIndustrialStyle</i>	100
6.4.6.3	Rahmengeometrie: <i>IfcDoorIndustrialFrameProperties</i>	101
6.4.6.4	Torflügelgeometrie: <i>IfcDoorIndustrialLeafProperties</i>	101
6.4.6.5	Anforderungen: <i>Pset_DoorIndustrialRequirements</i>	102
6.4.6.6	Beispiel	103

6.4.7	Dachbelichtungselemente	
6.4.7.1	Bauelement: <i>IfcRooflightElement</i>	103
6.4.7.2	Typdefinition: <i>IfcRooflightElementStyle</i>	104
6.4.7.3	Anforderungen: <i>Pset_RooflightElementRequirements</i>	104
6.4.7.4	Beispiel	105
6.4.8	RWA-Einrichtungen	106
6.4.9	Anforderungen an Bauelemente der IFC2x	106
6.5	Dach und Wand	
6.5.1	Grundzüge der Darstellung	106
6.5.2	Detailplanung: Schichtaufbau und Bauprodukte	108
6.5.3	Verlegeplanung	
6.5.3.1	Auflösung der Bauelemente in Bauteile: <i>IfcSlabsIsDecomposed</i>	108
6.5.3.2	Verlegebauteile: <i>IfcLayoutMember</i>	109
6.5.3.3	Typdefinition für Verlegebauteile: <i>IfcLayoutMemberStyle</i>	109
6.5.3.4	Gruppierung der Bauteile zu Schichten: <i>IfcBuildingElementLayer</i>	110
6.5.3.5	Befestigung der Verlegebauteile: <i>IfcFasteningElementProxy</i>	110
6.5.3.6	Palettierung	111
6.5.4	Beispiel: Kassettenwand.	111
6.6	Dach- und Wandöffnungen	
6.6.1	Allgemeines	114
6.6.2	Typdefinition für Lichtkuppeln: <i>IfcDomeLightStyle</i>	115
6.6.3	Geometrie der Kuppel: <i>IfcDomeProperties</i>	116
6.6.4	Geometrie des Aufsetzkranzes: <i>IfcDomeLightUpstandProperties</i>	116
6.6.5	Beispiel	117
6.7	Glasbau	
6.7.1	Grundzüge der Darstellung	118
6.7.2	Auflösung der Bauelemente in Bauteile: <i>IfcGlassFaceIsDecomposed</i>	121
6.7.3	Verglasungsbauteile: <i>IfcGlazingMember</i>	122
6.7.4	Typdefinition für Verglasungsbauteile: <i>IfcGlazingMemberStyle</i>	122
6.7.5	Befestigungsbauteile: <i>IfcGlazingSupportMember</i>	124
6.7.6	Typdefinition für Befestigungsbauteile: <i>IfcGlazingSupportMemberStyleLinear</i>	124
6.7.7	Beispiel 1: Pfosten-Riegel-Verglasung	124
6.7.8	Beispiel 2: Punktgestützte Verglasung	126
7	Sichten	
7.1	Allgemeines	129
7.2	Vertikale Systemkopplung	130
7.3	Horizontale Kopplung von Teilprozessen	132
7.4	Bestandteile der Sichtdefinitionen	136
8	Zusammenfassung und Bewertung	
8.1	Zusammenfassung	139
8.2	Bewertung der Ergebnisse	140

9	Ausblick	
9.1	Umsetzung der Ergebnisse	141
9.2	Längerfristige Ziele	142
10	Schrifttum	145
11	Glossar	155

Anhang A: Klassendefinitionen

A.1	Schemainformationen	167
A.2	<i>IFCMATERIALRESOURCEEXTENSION</i>	
	A.2.1 Klasse <i>IfcProductSpecification</i>	168
A.3	<i>IFCKERNELEXTENSION</i>	
	A.3.1 Klasse <i>IfcRelFixes</i>	168
A.4	<i>IFCPRODUCTEXTENSIONEXTENSION</i>	
	A.4.1 Klasse <i>IfcRelFixesSemantic</i>	169
	A.4.2 Klasse <i>IfcRelFixesGeometric</i>	169
	A.4.3 Klasse <i>IfcConstructionalElement</i>	169
A.5	<i>IFCSHAREDBLDGELEMENTSEXTENSION</i>	
	A.5.1 Klasse <i>IfcWallIsDecomposed</i>	170
	A.5.2 Klasse <i>IfcGlassFacade</i>	170
	A.5.3 Klasse <i>IfcGlassFace</i>	170
	A.5.4 Klasse <i>IfcShallowFoundation</i>	170
	A.5.5 Klasse <i>IfcWindowArrangement</i>	171
	A.5.6 Klasse <i>IfcDoorIndustrial</i>	171
	A.5.7 Klasse <i>IfcDoorIndustrialStyle</i>	171
	A.5.8 Klasse <i>IfcDoorIndustrialFrameProperties</i>	173
	A.5.9 Klasse <i>IfcDoorIndustrialLeafProperties</i>	174
	A.5.10 Klasse <i>IfcRooflightElement</i>	175
	A.5.11 Klasse <i>IfcRooflightElementStyle</i>	175
	A.5.12 Klasse <i>IfcSmokeAndHeatControlUnit</i>	177
	A.5.13 Klasse <i>IfcSmokeAndHeatControlUnitStyle</i>	178
A.6	<i>IFCOUTERSHELLDOMAIN</i>	
	A.6.1 Klasse <i>IfcSlabIsDecomposed</i>	179
	A.6.2 Klasse <i>IfcLayoutMember</i>	179
	A.6.3 Klasse <i>IfcLayoutMemberStyle</i>	179
	A.6.4 Klasse <i>IfcBuildingElementLayer</i>	180
	A.6.5 Klasse <i>IfcFasteningElementProxy</i>	181
A.7	<i>IFCOPENINGCONSTRUCTIONDOMAIN</i>	
	A.7.1 Klasse <i>IfcDoorStyleWithProducts</i>	183
	A.7.2 Klasse <i>IfcWindowStyleWithProducts</i>	184
	A.7.3 Klasse <i>IfcDoorIndustrialStyleWithProducts</i>	185
	A.7.4 Klasse <i>IfcSmokeAndHeatControlUnitStyleWithProducts</i>	186
	A.7.5 Klasse <i>IfcDomeLightStyle</i>	186
	A.7.6 Klasse <i>IfcDomeProperties</i>	187
	A.7.7 Klasse <i>IfcDomeLightUpstandProperties</i>	188
	A.7.8 Klasse <i>IfcPyramidLightStyle</i>	189
	A.7.9 Klasse <i>IfcPyramidProperties</i>	190

A.7.10	Klasse <i>IfcPyramidLightUpstandProperties</i>	190
A.7.11	Klasse <i>IfcRooflightBarrelVaultStyle</i>	192
A.7.12	Klasse <i>IfcRooflightVaultSegmentProperties</i>	193
A.7.13	Klasse <i>IfcRooflightVaultFrameProperties</i>	193
A.7.14	Klasse <i>IfcRooflightVaultTransomProperties</i>	193
A.7.15	Klasse <i>IfcRooflightGableRoofStyle</i>	195
A.7.16	Klasse <i>IfcRooflightGableSegmentProperties</i>	196
A.7.17	Klasse <i>IfcRooflightGableFrameProperties</i>	196
A.7.18	Klasse <i>IfcRooflightGableTransomProperties</i>	196
A.8	IFCGLAZINGCONSTRUCTIONDOMAIN	
A.8.1	Klasse <i>IfcGlassFaceIsDecomposed</i>	198
A.8.2	Klasse <i>IfcGlazingMember</i>	198
A.8.3	Klasse <i>IfcGlazingMemberStyle</i>	198
A.8.4	Klasse <i>IfcGlazingSupportMember</i>	199
A.8.5	Klasse <i>IfcGlazingSupportMemberStyleLinear</i>	199
A.9	IFCPROFILERESOURCEEXTENSION	
A.9.1	Klasse <i>IfcZetaShapeProfileDef</i>	200
A.9.2	Klasse <i>IfcEavesBeamProfileDef</i>	201
A.9.3	Klasse <i>IfcTrapezoidalSheetProfileDef</i>	202
A.9.4	Klasse <i>IfcCorrugatedProfileDef</i>	203
A.9.5	Klasse <i>IfcTrapezoidalCompoundProfileDef</i>	204
A.9.6	Klasse <i>IfcRoofSheetingProfileDef</i>	205
A.9.7	Klasse <i>IfcStandingSeamProfileDef</i>	206
A.9.8	Klasse <i>IfcFacadePanelProfileDef</i>	207
A.9.9	Klasse <i>IfcC-PanelProfileDef</i>	208
A.9.10	Klasse <i>IfcPlaneSandwichProfileDef</i>	209
A.9.11	Klasse <i>IfcProfiledSandwichProfileDef</i>	210
A.9.12	Klasse <i>IfcDoubleProfiledSandwichProfileDef</i>	211

Anhang B: Anforderungen

B.1	<i>Pset_RoofAndWallRequirements</i>	213
B.2	<i>Pset_GlassFaceRequirements</i>	214
B.3	<i>Pset_DoorIndustrialRequirements</i>	216
B.4	<i>Pset_RooflightElementRequirements</i>	217
B.5	<i>Pset_HeatAndSmokeControlUnitRequirements</i>	217
B.6	<i>Pset_SlabRequirements</i>	218
B.7	<i>Pset_BeamAndColumnRequirements</i>	219

Anhang C: Bezeichnungen für Befestigungsmittel

C.1	Vereinbarungen	221
C.2	Befestigungsmittel für Bleche	221

1 Einleitung

1.1 Problemstellung

Im Bauwesen sind seit etwa fünfzehn Jahren zwei Tendenzen zu beobachten, die ausschlaggebend für diese Arbeit sind. Die erste ist der flächendeckende Einzug der digitalen Planungswerkzeuge in Baufirmen und Ingenieurbüros. Dieses sind neben den klassischen Anwendungsprogrammen, die es mittlerweile zur Unterstützung des Planers bei den meisten Ingenieuraufgaben gibt, in jüngerer Zeit auch die modernen *Informations- und Kommunikationstechnologien* (IuK). Die zweite Tendenz ist die Verlagerung der Betätigungsfelder vieler Bauunternehmen und Ingenieurbüros. Gerade Stahlbauunternehmen beschäftigen sich außer mit dem Stahltragwerk zunehmend mit dem wesentlich lukrativeren Komplett- und Schlüsselfertigbau. Um sich von dem nicht eindeutig belegten Begriff des *Ausbaus* abzusetzen, wird in dieser Arbeit der Begriff des *Komplettbaus* für das erweiterte Betätigungsfeld des Stahlbauers zur Erstellung eines typischen Industrie- oder Bürogebäudes samt Hülle verwendet. Neben dem Tragwerk beinhaltet dies die Gewerke der Dach- und Außenwandbekleidung, den Ausbau von deren Öffnungen, den Glasbau, also Glasfassaden und –dächer, sowie Decken, Fundamente und Hallenböden.

Zumindest in Deutschland werden derartige Aufträge nicht vornehmlich von großen Konzernen geplant und abgewickelt, sondern durch kleine und mittelständische Bauunternehmen und Ingenieurbüros, die ad-hoc in General- und Nachunternehmerstrukturen oder Arbeitsgemeinschaften zusammenarbeiten. *Pegels* führt mehrere Fallbeispiele solcher Unternehmen und Büros an, die sich mit architektonisch und ingenieurtechnisch anspruchsvollen Projekten auf einem hart umkämpften Markt gegen Konkurrenten mit Niedriglohnpolitik behaupten können [90]. Ihr Erfolg beruht auf Zuverlässigkeit, Schnelligkeit und Qualität. Dies ist einerseits durch Nutzung leistungsfähiger Computerwerkzeuge in Verbindung mit moderner Informations- und Kommunikationstechnologie möglich, und andererseits durch flexible Zusammenarbeit mit wechselnden Partnern oder in strategischen Allianzen.

Durch diese Tendenzen ist der Kommunikationsbedarf im betrachteten Zeitraum ständig gestiegen, und zwar nicht nur zwischen den beteiligten Personen, sondern auch zwischen ihren Anwendungsprogrammen. Eines der wichtigsten Ziele auf dem Weg zu einer *integralen Projektplanung* ist deshalb die vollständige Kopplung aller Anwendungsprogramme, um die beteiligten Personen von der arbeitsaufwendigen, eintönigen und fehlerintensiven Mehrfachdateneingabe zu entlasten. Diese Kopplung kann mit herkömmlichen, bilateralen Schnittstellen nicht zufriedenstellend bewerkstelligt werden, da diese Schnittstellen nur bestimmte Arten von Daten übertragen, meistens die Geometrie von CAD-Zeichnungen. Alle übrigen Daten gehen deshalb bei jedem Austauschschritt verloren.

Ein ganzheitlicher Lösungsansatz ist das sogenannte *Produktmodell*. Ein Produktmodell ist ein standardisiertes Datenformat, das ein Bauwerk nicht in Form von Zeichnungsdaten beschreibt, sondern als räumliches *Objektmodell*, das alle im Projekt anfallenden Daten umfaßt. Wie in Abbildung 1.1 sehr vereinfacht dargestellt ist, kann sich jedes Anwendungsprogramm aus diesem Datenbestand bedienen. Somit ist eine redundanzfreie Systemkopplung möglich, bei der im Idealfall überhaupt kein Datenverlust stattfindet. Diese Kommunikation wird im folgenden als *Produktdaten-*

austausch bezeichnet. Voraussetzung ist allerdings, daß alle beteiligten Programme intern ebenfalls über Objektmodelle verfügen, oder daß sie das zentrale Modell zumindest in geeigneter Weise interpretieren können.

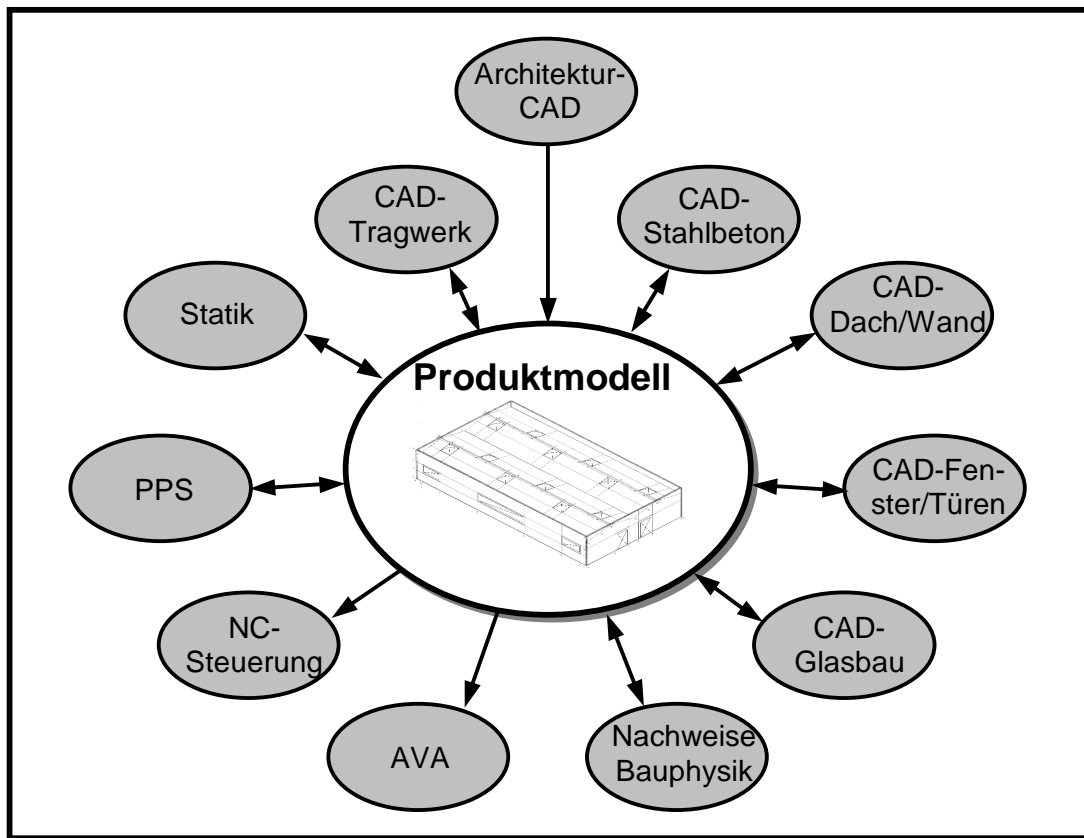


Abbildung 1.1: Prinzip des Produktmodells

In den letzten zehn Jahren wurden mehrere Produktmodelle entwickelt, die jeweils die Systemkopplung in Teilbereichen des Bauwesens erbringen sollten [11], [47], [48], [75]. Das Stahltragwerk betreffend wurde das von *Haller* entworfene Modell [47] zur *Produktschnittstelle Stahlbau* weiterentwickelt [32]. Sie findet, wenn auch zögerlich, Anwendungen in der Stahlbaupraxis [55]. Das ursprüngliche Ziel dieser Arbeit war es deshalb, auf diesem Stand der Technik aufzubauen und die Produktschnittstelle um die immer wichtiger werdenden Gewerke des Komplettbaus zu erweitern, da diese bislang von keinem der genannten Modelle abgedeckt sind. Dabei zeichneten sich jedoch die folgenden Probleme ab:

- 1) Das allein auf den Stahlbau ausgerichtete Konzept der Produktschnittstelle stößt bei einer Erweiterung um andere Gewerke schnell an seine Grenzen. Dies gilt besonders in bezug auf raumbildende Elemente und die daraus entstehenden höheren Ansprüche an Geometrie und *Gebäudetopologie*.
- 2) Ein vorrangig auf deutsche Verhältnisse ausgerichteter Software-Standard hat heutzutage nur sehr geringe Aussichten auf breite Akzeptanz.
- 3) Die Möglichkeiten der Datenmodellierung haben sich seit der Entstehung der Produktschnittstelle weiterentwickelt. Insbesondere die Methode der objektorientierten Modellierung, die bei der Produktschnittstelle nicht verwendet wurde, ist heutzutage Stand der Technik.

- 4) Die neuen Informations- und Kommunikationstechnologien, besonders das verteilte Arbeiten in Netzwerken, bieten einerseits neue Anwendungsmöglichkeiten für ein Produktmodell, andererseits entstehen durch sie aber auch neue Anforderungen.
- 5) 1997 erschien mit den *Industry Foundation Classes* (IFC) erstmals ein Produktmodell, das ausdrücklich auf die Integration des gesamten Bauwesens abzielt [58]. Die aktuelle Version *IFC2x* unterstützt bereits die Anwendungsgebiete der Architektur, der Installation von Heizung, Klima, Lüftung und Elektrizität, des Baubetriebs und des Facility Managements [59], nicht jedoch die Gewerke des Komplettbaus im Sinne dieser Arbeit.

Neben der Frage, ob eine Erweiterung der Produktschnittstelle Stahlbau überhaupt möglich ist, werfen die aufgeführten Punkte auch Zweifel an der späteren Akzeptanz einer erweiterten Produktschnittstelle durch die Software-Hersteller auf.

Keiner der aufgeführten Punkte spricht jedoch gegen das Produktmodell als Mittel zur Integration an sich, zu dem es nach wie vor keine Alternative gibt. Dies gilt auch für die Informations- und Kommunikationstechnologien, die zwar die räumlichen Entfernungen beim Datenaustausch überbrücken und teilweise die Verteilung der Daten übernehmen, die jedoch keine Standardisierung der Dateninhalte leisten.

1.2 Ziel dieser Arbeit

Das Ziel dieser Arbeit wurde daraufhin geändert: Es soll zwar weiterhin ein Produktmodellstandard entwickelt werden, der die Kopplung der Anwendungsprogramme im Komplettbau ermöglicht, er soll jedoch den im vorangegangenen Abschnitt aufgeführten Problemen Rechnung tragen. Dazu ist es notwendig, die derzeitige Produktschnittstelle Stahlbau als Stand der Technik kritisch zu hinterfragen, und die übrigen Produktmodelle des Bauwesens daraufhin zu untersuchen, ob eine Harmonisierung des neuen Modells mit ihnen möglich und sinnvoll ist. Außerdem soll die Verwendung neuartiger Modellierungstechniken und die Nutzung der modernen Informations- und Kommunikationstechnologien geprüft werden.

Der zu entwickelnde Standard soll ein Objektmodell definieren, das die folgenden Gewerke abdeckt: Dach- und Außenwandbekleidungen des Büro- und Industriebaus, den Ausbau von deren Öffnungen, also Außentüren, -fenster, Tore, Dachbelichtung, Rauch- und Wärmeabzugseinrichtungen, sowie den Glasbau. Diese Gewerke müssen mit für Ingenieurzwecke ausreichender Detaillierung dargestellt werden können. Da eine vollständige Behandlung des Stahlbetonbaus den Rahmen dieser Arbeit sprengen würde, sollen lediglich die wichtigsten Bauelemente, also Decken, Wände, Fundamente und Hallenböden, in grober Detaillierung aufgenommen werden. Außerdem soll die Kopplung mit dem bislang durch die Produktschnittstelle abgedeckten Stahltragwerk möglich sein. Alle weiteren Leistungsphasen der Planung (zeichnerische Darstellungen, Mengen- und Massenermittlungen oder Erstellung von Leistungsverzeichnissen) sollen nicht Teil des Modells sein. Sie können jedoch von entsprechender Anwendungs-Software daraus abgeleitet werden.

Da die Marktsituation im Bauwesen mehr denn je individuelle Planungsstrukturen für jedes Projekt erfordert, muß das Produktmodell möglichst flexibel angewandt werden können und darf nicht auf starre Planungsszenarien beschränkt sein. So muß es auch möglich sein, anfänglich nur Teilprozesse zu integrieren, bevor eine breitere Anwendung des Modells erreicht wird.

Ziel dieser Arbeit ist es jedoch nicht, den Datenaustausch zu automatisieren: Es sollen weder Software noch Datenbanken programmiert werden. Noch weniger kann es

das Ziel sein, die Planung selbst zu automatisieren. Dies ist generell nicht sinnvoll, da keine Technologie die Kreativität und die Erfahrung von Ingenieuren und Architekten ersetzen kann.

1.3 Vorgehensweise und Gliederung

In Kapitel 2 werden die für diese Arbeit benötigten theoretischen Grundlagen der Produktmodellierung erläutert. Der Schwerpunkt liegt auf den Neuerungen, die seit der Arbeit von *Haller* stattgefunden haben [47].

In Kapitel 3 wird das Umfeld untersucht, in dem das Produktmodell eingesetzt werden soll. Dazu wird ein Überblick über den Entwicklungsstand der Software für die Planungsaufgaben im Komplettbau gegeben. Außerdem werden, wie oben bereits angesprochen, die im Bauwesen zur Verfügung stehenden Produktmodelle hinsichtlich einer möglichen Harmonisierung bewertet. Dazu gehört auch die Untersuchung der tatsächlichen Anwendung dieser Modelle in der Baupraxis. Schon vorwegnehmend: Die Bewertung ergibt, daß es möglich und sinnvoll ist, das Komplettbaumodell als Erweiterung der IFC2x zu definieren.

Bevor mit der eigentlichen Modellierung des Produktmodells begonnen werden kann, muß in Kapitel 4 der aus Ingenieursicht sinnvolle Umfang des Modells genauer abgesteckt werden. Dies geschieht anhand einer Analyse der Planungsschritte und Informationsflüsse im Komplettbau. Außerdem wird durch eine detaillierte Betrachtung der Bauarten und Bauteile für die verschiedenen Gewerke festgelegt, welche Dateninhalte und Detaillierungen im Modell darstellbar sein müssen.

Anschließend werden in Kapitel 5 die für den Komplettbau relevanten Datenstrukturen und -inhalte der IFC2x erläutert, um darauf aufbauend in Kapitel 6 das neue Modell für den Komplettbau zu definieren. In Kapitel 7 werden schließlich mehrere sogenannte *Sichten* auf das Komplettbaumodell festgelegt, die den Software-Herstellern eine Auswahl zwischen unterschiedlichen Modellbereichen bieten, die eine Software unterstützen kann. Die gewählte Sicht bestimmt die Fähigkeiten der Software beim Produktdatenaustausch.

Kapitel 8 enthält eine Zusammenfassung der Arbeit und eine Bewertung der Ergebnisse. In Kapitel 9 wird ein Ausblick auf die Umsetzung dieser Ergebnisse und auf zukünftige Aufgaben in der Produktmodellierung gegeben.

2 Grundlagen der Produktmodellierung

2.1 Prinzip des Produktmodells

2.1.1 Grundlegendes

Wie einleitend bereits erläutert wurde, können auf der Grundlage von Produktmodellen Anwendungsprogramme mit gleichen oder unterschiedlichen Anwendungsgebieten *kommunizieren*. Dies bedeutet mehr als den reinen Datenaustausch zwischen zwei Programmen, da alle im betrachteten Zeitraum des Lebenszyklus eines Produktes zum Einsatz kommenden Programme eingeschlossen sind. Die Vorgabe, daß möglichst keine Daten verloren gehen sollen, macht dieses Vorhaben wesentlich komplizierter als den reinen Datenaustausch. In dieser Arbeit soll der Begriff *Produktdatenaustausch* für diese Art der Kommunikation verwendet werden. Synonym dazu werden im Schrifttum auch die Begriffe der *Integration* [102] oder der *Interoperabilität* [59] verwendet.

Das Produkt, also zum Beispiel ein Bauwerk, wird dazu modellhaft dargestellt. *Grabowski* definiert das Produktmodell als „umfassende, digitale Produktbeschreibung“ [44], *Scherer und Katranuschkov* als „die Menge aller Daten und Wissen über das Produkt und dessen Produktion“ [102]. Dazu wird das Produkt in gedankliche Einheiten zerlegt, die sogenannten *Entities*, die über Eigenschaften in Form von *Attributen* verfügen. Der Anwendungsbereich des Produktmodells umfaßt in der Regel mehr Daten als jedes der beteiligten Programme interpretieren kann. Da das Modell jedoch nicht in Einzelteile zerfallen darf, müssen die *Entities* miteinander verknüpft sein. Wenn außerdem die nicht interpretierten Daten nicht verloren gehen sollen, müssen zusätzliche Regeln für die Handhabung des anfallenden Datenbestandes festgelegt werden. Diese sollen die Konsistenz, die Redundanzfreiheit und die semantische Integrität des Datenbestandes sichern.

Das Produktmodell benötigt dazu die folgenden Bestandteile, die in Form eines herstellerneutralen Standards zu dokumentieren sind:

- 1) Als *Vorüberlegung* sollte der angestrebte Anwendungsbereich des Produktmodells genau definiert, die darin vorkommenden Planungs- und Fertigungsprozesse analysiert und die anfallenden Informationsflüsse dargestellt werden. Um dem Anwender den Zugang zum Produktmodell zu erleichtern, müssen diese Vorüberlegungen ebenfalls dokumentiert werden.
- 2) Das Kernstück eines Produktmodells bildet das *Datenmodell*, das die unter 1) definierten Sachverhalte in geeigneter Weise modellhaft darstellt, das also die logische Datenstruktur festlegt. Es sollte möglichst das ingenieurgemäße Denken zum Vorbild haben.
- 3) Um für Anwendungs-Software die Möglichkeit zu schaffen, lediglich Teile des Datenmodells zu unterstützen, muß das Modell sinnvoll in sogenannte *Sichten* (auch als *Konformitätsklassen* oder *Implementierungsbereiche* bezeichnet) aufgeteilt werden. Bildlich gesprochen definiert eine Sicht einen Teil des Modells, der nur für bestimmten Gruppen von Software sichtbar ist. Für die Unterstützung jeder dieser Sichten müssen Anforderungen und Regeln aufgestellt werden, die sogenannten *Konformitätsanforderungen*.

- 4) Da das Modell rein beschreibender Natur ist, es also keine aktiven Software-Komponenten besitzt, erfolgt der Datenaustausch durch Anbindung von Anwendungs-Software an das Produktmodell. Man spricht von der *Implementierung* des Modells in die Software.
- 5) Schließlich muß gewährleistet sein, daß die an das Produktmodell angebundene Software die Konformitätsanforderungen erfüllt. Ist dies der Fall, ist die Software *konform* zu den Regeln des Produktmodells. Die Überprüfung der Anwendungsprogramme heißt dementsprechend *Konformitätsprüfung*. Sie soll von einer neutralen Institution durchgeführt werden und dem Anwender das reibungslose Funktionieren des Produktdatenaustausches garantieren.

Das größte Problem der in der Vergangenheit entwickelten Produktmodelle war nicht etwa datentechnischer Art, sondern einfach die mangelnde Akzeptanz durch die Software-Hersteller. Dies liegt jedoch im Wesen des Produktmodells: Der Anspruch, die durch alle beteiligten Programme anfallenden Daten verlustfrei zu transportieren, steht immer im Konflikt zum Anwendungsbereich des einzelnen Programms und zum programminternen Objektmodell. Die Implementierung ist deshalb in der Regel sehr aufwendig. Der finanzielle Aufwand für die Implementierung der Produktschnittstelle Stahlbau [32] in das CAD-Programm BOCAD erreichte beispielsweise die Größenordnung von € 250.000,- und bedeutete die entsprechende Bindung hochqualifizierten Personals. Einerseits bewirkt ein sehr großer angestrebter Anwendungsbereich bei hohem Detaillierungsgrad eine ausufernde Komplexität des Produktmodells, was der Akzeptanz sicherlich abträglich ist. Andererseits ist eine Einschränkung des Anwendungsbereiches nur durch die Beschränkung des Modells auf Teile im Lebenszyklus des Produktes zu erreichen, was wiederum dem Allgemeingültigkeitsanspruch widerspricht. Dabei ist auch zu bedenken, daß die nachträgliche Erweiterung eines Produktmodells sehr problematisch ist, da dies in der Regel nicht ohne größere Veränderungen im bestehenden, bereits implementierten Modell einhergeht.

Eine optimale Lösung des Problems hängt deshalb bereits entscheidend von der Sorgfalt und Gründlichkeit der Vorüberlegungen ab. Die Problematik betrifft aber auch die Einteilung des Modells in Sichten, da einerseits wenige große Teile die Anwendungsprogramme überfordern, weil deren Anwendungsgrenzen überschritten werden, andererseits viele kleine, leicht zu implementierende Teilmodelle wiederum keine wirkliche Kommunikation zulassen, weil es keinen ausreichenden gemeinsamen Nenner gibt. Schließlich spielen bei der Umsetzung der Vorüberlegungen im Datenmodell die *Art der Modellierung* und die Wahl einer geeigneten *Modellarchitektur* eine große Rolle, letztere im Hinblick auf das Zusammenwirken der Teilmodelle und auf möglichst flexible Möglichkeiten zur Implementierung in unterschiedlichen Detaillierungsgraden. Aus diesem Grunde sollen die Aspekte der Modellierungsart und der Modellarchitektur in den folgenden Abschnitten genauer betrachtet werden.

2.1.2 Modellierungsarten

Um dem Vorbild des menschlichen Denkens möglichst nahe zu kommen, ist es für das Datenmodell von großer Bedeutung, auf welche Art es modelliert wird. Werden nur die Dateninhalte der Entities betrachtet, kann unterschieden werden zwischen der rein *geometrischen* Modellierung, die die Form und räumliche Lage von Körpern beschreibt, der *topologischen* Modellierung, die die Zusammenhänge dieser Körper darstellt, und der *semantischen* Modellierung, bei der andere als die im

CAD-Programm sichtbaren Objekte vermittelt werden. Ein komplexes Produkt wie ein Bauwerk läßt sich jedoch nicht durch eine dieser drei Modellierungsarten allein darstellen. Andere Arten der Modellierungen vereinen mehrere dieser Eigenschaften. Die *feature-basierte* Modellierung beruht darauf, daß den Bestandteilen des Modells eine beliebige Anzahl von Operationen und dadurch implizite Geometrien zugeordnet werden. *Saal* und *Haller* zeigen die Anwendung solcher *features* anhand der Anarbeitung eines Stahlbauteils [100]. Bei der *objektorientierten Modellierung* ist der zentrale Begriff das Objekt, das nach *Scherer und Katranuschkov* als abgeschlossene Einheit definiert ist, die beliebige Eigenschaften sowie Funktionen enthalten kann, sich hierarchisch strukturieren läßt und Vererbungen zuläßt [102]. Mit dem Mittel der Vererbung lassen sich datentechnisch handhabbare Strukturen darstellen, die am besten von allen genannten Modellierungsarten dem menschlichen Denken entsprechen. Weitere Vorteile der Objektorientierung sind die durch Objektkapselung und Vererbung erreichbare einfachere Pflege des Modells und die Möglichkeit zur Mehrfachverwendung von Programmteilen bei der Schnittstellenprogrammierung. Dadurch verringert sich der Aufwand für die Implementierung.

Objektorientierte Ansätze zur Problemlösung wurden für das Bauwesen zuletzt im DFG-Schwerpunktprogramm „Objektorientierte Modellierung in Planung und Konstruktion“ untersucht [52], [51]. Von den dort genannten Modellierungskonzepten hat die Methode der *Unified Modeling Language* (UML) in der Software-Entwicklung breite Anwendung gefunden [99]. In dieser Arbeit soll UML jedoch nicht verwendet werden, da im Gegensatz zur Software-Entwicklung statische Aspekte für die Produktmodellierung von vorrangiger Bedeutung sind. Das heißt, daß die Darstellung beispielsweise von Bauteilen und deren Eigenschaften Vorrang vor der Darstellung von dynamischen Ereignissen, Zustandsänderungen und Datenflüssen hat.

2.1.3 Modellarchitektur

Mit dem Begriff der Modellarchitektur wird in dieser Arbeit die Strukturierung des Datenmodells und das Zusammenwirken der Teilmodelle bezeichnet. Die Schnittmenge der Teilmodelle bestimmt die Dateninhalte, die zwischen ihnen und damit letztendlich zwischen verschiedenen Anwendungsprogrammen ausgetauscht werden können. Die im folgenden aufgeführten Modellarchitekturen unterscheiden sich durch unterschiedliche Ansätze zur Erreichung eines optimalen Kompromisses zwischen den Polen des größtmöglichen gemeinsamen Nenners und des geringsten Aufwandes für die Implementierung.

Hannus und *Karstila* unterscheiden in mehrere Typen [50], von denen hier nur die wichtigsten aufgeführt sind. Das *neutrale Modell*, siehe Abbildung 2.1a), entspricht dem Idealtypus des Produktmodells, bei dem jedes Anwendungsprogramm in der Lage sein muß, alle Daten zu bewältigen. Bei komplexen Modellen bedeutet dies einen enormen Implementierungsaufwand. Das *Applikationsdomänenmodell*, siehe Abbildung 2.1b), setzt voraus, daß spezielle, genau eingrenzbar Anwendungsgebiete existieren. Innerhalb dieser *Domänen* findet der Datenaustausch statt. Die Domänen müssen nicht notwendigerweise Gemeinsamkeiten haben, der Produktdatenaustausch zwischen den Domänen ist dann jedoch nicht möglich. Die Komplexität der Implementierungen hängt von der Größe der jeweiligen Domäne ab. Das in Abbildung 2.1c) skizzierte Modell ermöglicht dagegen auf niedrigem Niveau auch den Produktdatenaustausch zwischen den Domänen, indem *gemeinsame Ressourcen* genutzt werden. Eine Ressource ist ein Modellteil, in dem ein grundlegendes und allgemeingültiges Darstellungskonzept definiert ist, zum Beispiel für grundlegende

Datentypen oder Geometriedaten. Abbildung 2.1d) zeigt die Kooperation verschiedener Domänen über einen *gemeinsamen Modellkern*, der grundlegende, im Gegensatz zu den gemeinsamen Ressourcen jedoch greifbare Objekte darstellt, die auch untereinander verknüpft sein können. Dies können zum Beispiel Bauteile mit niedrigem Detaillierungsgrad sein. Innerhalb der einzelnen Domänen können diese grundlegenden Objekte durch Vererbung weiter spezialisiert werden. Die durch diese Spezialisierung erzielten Daten höherer Detaillierung gehen jedoch im Austausch zwischen den Domänen verloren. Die am weitesten entwickelte Variante ist ein Modell mit gemeinsamem Kern und einem sogenannten *Metamodell*, siehe Abbildung 2.1e). Das Metamodell enthält grundlegende, generische Datenstrukturen, die in den Domänen in spezialisierter Form wieder aufgegriffen werden. Greifbare Objekte sind, wie bei Variante 2.1d), im Modellkern definiert. Die zusätzlichen gemeinsamen Strukturen des Metamodells versprechen darüber hinaus, daß die einzelnen Domänen einander ähnlicher werden, und daß das gesamte Modell damit homogener wird. Umfang und Aufwand für die Implementierung werden dadurch nicht größer. Diese Modellarchitektur vereinfacht außerdem die modulare Weiterentwicklung des Modells und die Entwicklung von modellspezifischen Programmierwerkzeugen, da überall im Modell immer gleiche Datenstrukturen auftreten.

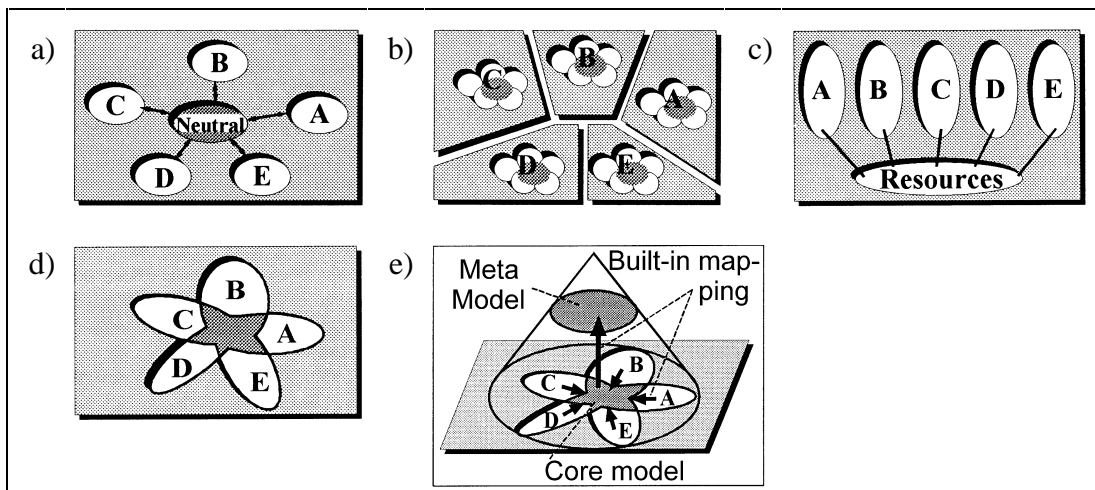


Abbildung 2.1 aus [50]: a) Neutrales Modell, b) Applikationsdomänenmodell, c) gemeinsame Ressourcen, d) gemeinsamer Modellkern, e) gemeinsamer Modellkern und Metamodell

2.2 Standardisierung nach ISO 10303

2.2.1 Allgemeines

Jede EDV-Schnittstelle basiert auf einer Vereinbarung zwischen den beteiligten Software-Herstellern. Im Sinne eines weltweit anerkannten und allgemeingültigen Produktmodells wurde Mitte der 1980er-Jahre mit der Entwicklung einer internationalen Norm begonnen, die einheitliche Regelungen und Mechanismen zur Definition und Anwendung von Produktmodellen festlegen sollte: ISO 10303 – *Standard for the Exchange of Product Data* (STEP). Vorherige, vornehmlich aus dem Flugzeug-, Maschinen- und Automobilbau stammende Standards sind in die Entwicklung eingeflossen [1], [86], [87], [111], [116]. 1992 erschienen die ersten Teile der Norm als Entwurf; mittlerweile existiert eine umfangreiche Normenreihe [73]. ISO 10303 bietet Werkzeuge und Regeln für die Definition von Produktmodellen, die unabhän-

gig von technischen Anwendungsgebieten sind, sowie genormte Methoden für die Implementierung der Modelle. Darüber hinaus beinhaltet ISO 10303 eigenständige Produktmodelle für bestimmte Anwendungsgebiete. Bei der Entwicklung der Norm wurde größter Wert auf die Unabhängigkeit aller Teile von Hard- und Software gelegt. Die Verwendbarkeit von Software-Werkzeugen, zum Beispiel zur Schnittstellenprogrammierung, ist jedoch ausdrücklich vorgesehen. Ein weiteres Grundprinzip von ISO 10303 ist die strikte Trennung der in Abschnitt 2.1.1 dieser Arbeit genannten Bestandteile eines Produktmodells. Der Zugang zu einer aktuellen Übersicht über die zahlreichen Teile der Norm ist im Schrifttum angegeben [74].

2.2.2 Datenmodellierung (Description Methods)

Zur Definition von Datenmodellen bietet Teil 11 von ISO 10303 die formale Spezifikationssprache *EXPRESS*. Die Vorteile einer formalen Spezifikation sind einerseits die Eindeutigkeit und andererseits die Maschinenlesbarkeit im Hinblick auf die Anwendung von Programmierwerkzeugen. *EXPRESS* ist jedoch keine Programmiersprache. Die wichtigsten Sprachmittel wie Entities, Attribute, Typen und Verweise sind bereits von *Haller* erläutert worden [47]. Ergänzend dazu soll an dieser Stelle lediglich das bereits oben angesprochene Prinzip der Vererbung gezeigt werden, da es die Grundlage der objektorientierten Modellierung ist.

Beispiel 2.1a) EXPRESS-Spezifikation

```
SCHEMA beispiel;

ENTITY bauteil
  SUPERTYPE OF
    (ONEOF (traeger, rechteck_wand));
  bezeichnung : STRING;
  position    : STRING;
END_ENTITY;

ENTITY traeger
  SUBTYPE OF (bauteil);
  profil      : STRING;
  laenge     : REAL;
END_ENTITY;

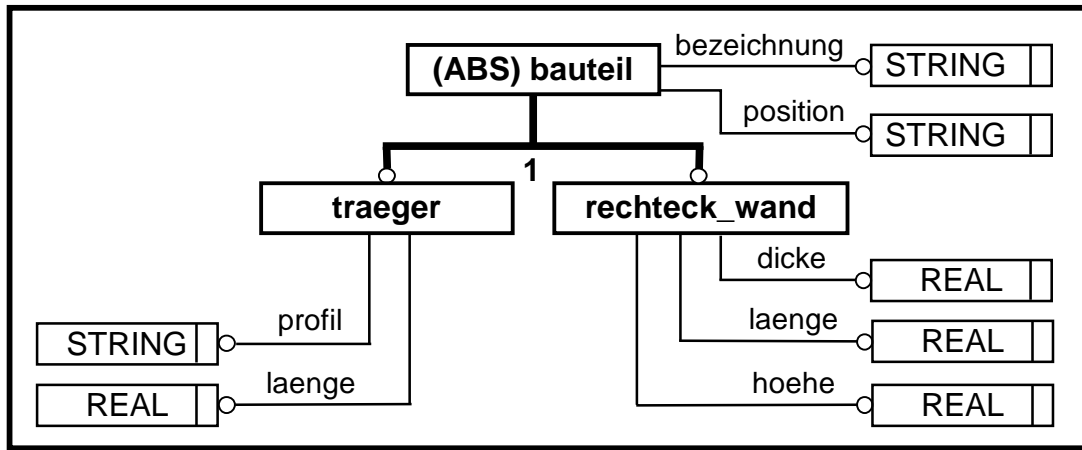
ENTITY rechteck_wand
  SUBTYPE OF (bauteil);
  dicke      : REAL;
  laenge     : REAL;
  hoehe     : REAL;
END_ENTITY;

END_SCHEMA;
```

Beispiel 2.1 zeigt ein sehr einfaches Datenmodell, in dem eine Vererbungsstruktur für Bauteile definiert ist: Das Entity *bauteil* besitzt als Eigenschaften lediglich eine Bezeichnung und eine Positionsnummer (Attribute *bezeichnung* und *position*). Diese Attribute werden an seine Untertypen vererbt, die Entities *traeger* und *rechteck_wand*, die jeweils weitere Eigenschaften erhalten. Der Träger wird durch eine Profilbezeichnung und eine Länge dargestellt, die rechteckige Wand durch Dicke, Länge und Höhe. In Richtung der Vererbung findet also eine Spezialisierung statt.

Beispiel 2.1a) zeigt die Spezifikation in EXPRESS, wobei die Vererbung durch die Deklarationen *SUPERTYPE OF* und *SUBTYPE OF* definiert ist. In der anschaulicheren graphischen Variante *EXPRESS-G* wird die Vererbung durch eine dicke Linie symbolisiert, siehe Beispiel 2.1b).

Beispiel 2.1b) EXPRESS-G-Spezifikation



Beispiel 2.1a) zeigt ein vollständiges *EXPRESS-Schema*, das durch die Deklarationen *SCHEMA* und *END_SCHEMA* eingeschlossen ist. Hier genügt ein einziges Schema zur Spezifikation des gesamten Datenmodells. Bei komplexeren Modellen ist es dagegen sinnvoll, mehrere Schemata zu verwenden, die sich untereinander mit sogenannten *Interfaces* referenzieren können. Die verschiedenen, in Abschnitt 2.1.3 genannten Bestandteile der Modellarchitektur beispielsweise werden in der Regel in getrennten EXPRESS-Schemata definiert.

2.2.3 Implementierung (Implementation Methods)

ISO 10303 definiert vier Stufen des Produktdatenaustausches, sogenannte *Implementation Levels*. Sie unterscheiden sich durch Komplexität und Leistungsvermögen. Auf der ersten Stufe ist die Systemkopplung über Austauschdateien vorgesehen, auf der zweiten Stufe über ein standardisiertes, im Speicher gehaltenes Zwischenformat, dem sogenannten *Repository*. Auf der dritten Stufe erfolgen Datenspeicherung und -zugriff über ein Datenbankmanagementsystem (DBMS), und auf der vierten Stufe über ein wissensbasiertes System. In allen vier Fällen ist die Implementierung unabhängig von der Definition des Produktmodells. Die Möglichkeiten der Implementierung sind in den folgenden Teilen der Norm standardisiert: Teil 21 legt die Syntax für Austauschdateien fest, die Teile 22 bis 27 das sogenannte *Standard Data Access Interface (SDAI)*. SDAI definiert einen standardisierten Satz an Befehlen, die den Datenzugriff sowohl für den Einsatz von Programmierwerkzeugen zur Schnittstellenprogrammierung (*STEP-Toolkit*) als auch für den Zugriff auf zentrale Datenbanken ermöglicht. Derzeit werden die Programmiersprachen C/C++ und Java unterstützt.

Beispiel 2.2a) zeigt ein Fragment einer Austauschdatei. Dabei wurde das Beispiel aus dem vorangegangenen Abschnitt aufgegriffen. Die dort definierten Entities werden in der Austauschdatei mit Attributwerten versehen, so daß sie konkrete Bauteile darstellen. Die mit Werten versehenen Datensätze heißen *Instanzen*. Man sieht, daß die Untertypen *traeger* und *rechteck_wand* die Attribute *bezeichnung* und *position*

ihres Obertyps *bauteil* geerbt haben, der beim sogenannten *Internal Mapping*¹ selbst nicht in die Austauschdatei geschrieben wird.

Beispiel 2.2a) Austauschdatei nach ISO 10303, Teil 21 – Internal Mapping

```
ISO-10303-21;
...
#1=TRAEGER( 'Deckentraeger', 'POS1',
            'HE300A', 7500.0);
#2=RECHTECK_WAND('Seitenwand', 'POS2',
                 250.0, 15000.0, 2750.0);
...
END-ISO-10303-21;
```

In Beispiel 2.2b) sind die gleichen Daten instanziiert, jedoch nach dem Verfahren des *External Mapping*. Im Unterschied zu Fall a) werden in der Attributliste die einzelnen Entities der Vererbungshierarchie genannt. Das Beispiel verdeutlicht, wie die Vererbung als Mittel zur flexiblen Steigerung des Detaillierungsgrades und der Spezialisierung eingesetzt werden kann: Eine Software-Applikation, die lediglich den groben Detaillierungsgrad des Entities *bauteil* interpretieren kann, kann diese Daten auch aus einer Austauschdatei beziehen, die Daten des höheren Detaillierungsgrades mit den Entities *traeger* und *rechteck_wand* enthält.²

Beispiel 2.2b) Austauschdatei nach ISO 10303, Teil 21 – External Mapping

```
ISO-10303-21;
...
#1=( BAUTEIL('Deckentraeger', 'POS1')
    TRAEGER('HE300A', 7500.0)           );
#2=( BAUTEIL('Seitenwand', 'POS2')
    RECHTECK_WAND(250.0, 15000.0, 2750.0) );
...
END-ISO-10303-21;
```

Das Mapping erfolgt durch Funktionen im Anwendungsprogramm, die zwischen internem und externem Format übersetzen. Abbildung 2.2 zeigt die dazu notwendigen Softwarestrukturen bei Einsatz eines typischen Programmierwerkzeuges, das über ein eigenes Repository und SDAI verfügt. Der Schnittstellenprogrammierer muß hier lediglich Übersetzungsfunktionen zwischen Applikationsdatenbank und Werkzeug erstellen. Es besteht hierbei sowohl die Möglichkeit zum einfachen Datenaustausch als auch zur Anbindung an eine zentrale Datenbank; beides übernimmt das Toolkit. Ein weiterer Vorteil des Programmierwerkzeuges ist die Anpassung der Implementierung an neue Schnittstellenversionen, die ihm über das computerlesbare EXPRESS-Schema zugänglich gemacht werden. Dadurch sind im Idealfall überhaupt keine Änderungen der Übersetzerprogramme notwendig. Dies ist

¹ Als *mapping* wird der Vorgang des Schreibens von Daten einer programminternen Applikationsdatenbank in ein externes Austauschformat oder umgekehrt bezeichnet.

² Anmerkung: Dies ist natürlich auch mit dem Verfahren des Internal Mapping möglich, nur daß dies am Beispiel nicht so offensichtlich wird. Der eigentliche Sinn des External Mappings ist im Zusammenhang mit Mehrfachvererbungen zu sehen, die hier jedoch nicht behandelt werden.

durch die strikte Trennung von Modelldefinition und Implementierung in ISO 10303 möglich.

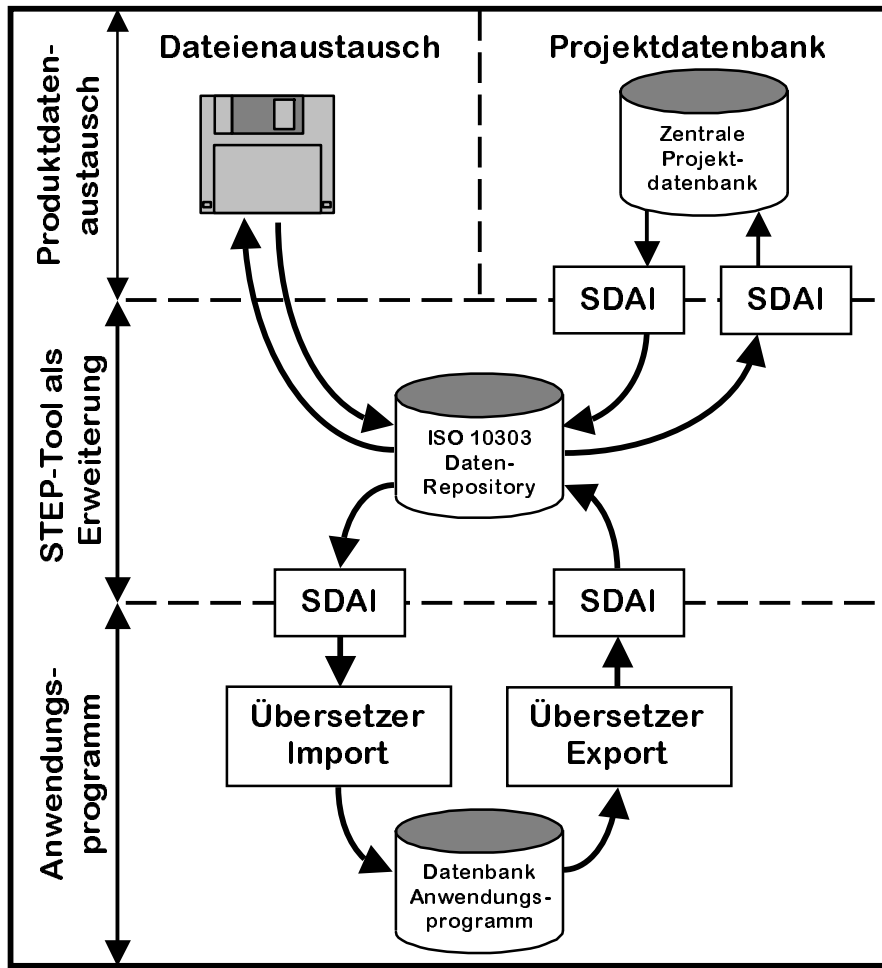


Abbildung 2.2: Schnittstellenimplementierung mit Programmierwerkzeug

Der Vollständigkeit halber sollen hier noch zwei Teile von ISO 10303 genannt werden, die eigentlich zum Komplex der Modellierung gehören: In Teil 12 wird die formale Sprache *EXPRESS-I* zur Darstellung von instanziierten Beispielen definiert und im Teil 14 die Sprache *EXPRESS-X*. Letztere ist eine sogenannte *Mapping-Sprache*, mit der die Übertragung von Daten aus einem Datenmodell in ein anderes definiert werden kann.

2.2.4 Konformitätsprüfung (Conformance Testing)

Zur Konformitätsprüfung sieht ISO 10303 ein sehr formales Vorgehen vor, das in den Teilen 31 bis 35 definiert ist. Das Prinzip besteht darin, daß für jeden zu testenden Sachverhalt ein sogenannter *abstrakter Testfall* definiert wird, der exakte Kriterien für Bestehen oder Nichtbestehen des Tests enthält. Der Test wird als *konkreter Testfall* mit einem Referenzbeispiel durchgeführt. Auch hierbei ist der weitgehende Einsatz von Computerwerkzeugen möglich.

ISO 10303 definiert die Konformität nicht für ein Anwendungsprogramm allein, sondern für ein komplettes System aus Hardware, Betriebssystem und Anwendungsprogramm.

2.2.5 Gemeinsame Ressourcen (Common Resources)

Wie in Abschnitt 2.1.3 erläutert wurde, ermöglichen gemeinsam genutzte Ressourcen den Produktdatenaustausch zwischen Teilmodellen auf niedrigem Niveau. ISO 10303 bietet in den Teilen 41 bis 109 Ressourcen für folgende Daten: geometrische, topologische und visuelle Darstellung, Materialeigenschaften, Toleranzen, Verwaltung von Prozeßstrukturen, technische Zeichnungen und Kinematik. Es wird unterschieden in anwendungsunabhängige Ressourcen, deren Verwendung jedem Produktmodell vorgeschrieben ist, und in anwendungsabhängige Ressourcen. Letztere müssen lediglich von bestimmten Anwendungsgebieten unterstützt werden.

Dieses Konzept wird durch die neueren *Application Interpreted Constructs* (Teile 501 bis 520) und *Application Modules* (Teile 1001 bis 1009) ergänzt, auf die hier jedoch nicht eingegangen werden soll.

2.2.6 Produktmodelle in ISO 10303 (Application Protocols)

Außer den bereits erwähnten Werkzeugen und Regeln zur Definition und zur Anwendung von Produktmodellen im allgemeinen sind in ISO 10303 eigenständige Produktmodelle definiert, die sogenannten *Application Protocols* (APs). Damit sich alle diese Modelle in den Gesamtzusammenhang der Norm einfügen, ist jedem AP eine Reihe formaler Anforderungen vorgeschrieben, deren Funktionen den in Abschnitt 2.1.1 beschriebenen Bestandteilen eines Produktmodells entsprechen.

Der Anwendungsbereich wird im sogenannten *Scoping Statement* definiert. Zur Analyse der darin anfallenden Informationsflüsse und Prozesse ist das sogenannte *Application Activity Model* (AAM) in der Notation IDEF0 vorgeschrieben [88]. Um diese Betrachtungen aus der „realen Welt“ im Hinblick auf die formale Modellierung zu strukturieren, werden sie im sogenannten *Application Reference Model* (ARM), auch konzeptuelles Produktmodell genannt, in Objekte der „Informationswelt“ überführt. Dies geschieht bereits in EXPRESS-G.

Diesen Vorüberlegungen folgend wird das *Application Interpreted Model* (AIM), das eigentliche Datenmodell in EXPRESS, unter Verwendung der vorgeschriebenen gemeinsamen Ressourcen definiert. Die Überführung der Objekte des ARM in solche des AIM muß im AP nachvollziehbar dokumentiert sein.

In den Konformitätsanforderungen werden verschiedene Sichten zur Implementierung des Datenmodells angeboten, und es werden die Regeln zur Implementierung dieser Sichten aufgestellt. Außerdem werden für jedes AP die zur Konformitätsprüfung notwendigen abstrakten Testfälle definiert. Für die Implementierung stehen die in Abschnitt 2.2.3 erläuterten vier Stufen zur Verfügung.

Jedes AP muß ein sehr langwieriges Verfahren von der Einreichung über eine Revision und öffentlicher Einspruchsfrist bis zu seiner endgültigen Anerkennung als Teil der Norm durchlaufen. Derzeit existieren APs für allgemeine technische Zeichnungen, für technische Zeichnungen im Maschinenbau und der Elektrotechnik, für Regelungstechnik, Metallverarbeitung sowie für Bauzeichnungen.

2.3 Datenmodellierung mit XML

Da die neuen Informations- und Kommunikationstechnologien in den letzten Jahren sehr stark an Bedeutung gewonnen haben, wird auch die Forderung nach entsprechenden Formen des Produktdatenaustausches lauter, die über die bloße Nutzung von E-Mail und Internet zum Versenden von Dateien hinausgehen. Diese werden in Kapitel 3 dieser Arbeit noch genauer betrachtet werden; hier soll lediglich die Formatspezifikationsprache XML (*Extensible Markup Language*) erläutert werden.

Der Erfolg des Internets basiert auf der Präsentation von Daten auf der Grundlage von SGML [72], einer sogenannten Formatspezifikations- oder Metasprache. Das Prinzip ist analog zu der in der Produktmodellierung üblichen formalen Trennung von Modelldefinition in Form eines EXPRESS-Schemas und von Austauschdateien: Die in SGML geschriebene *Dokumenttyp-Definition* (DTD) definiert ein Format für Dokumente, die dann als sogenannte *Tagged Documents* instanziiert werden. Im Falle des populären Formates HTML werden diese Dokumente in der Regel von Text-Browsern interpretiert. *Armstrong* und *Lockley* führten jedoch bereits 1994 diese Konstellation zur Definition eines Datenaustauschformates vor [3]. Eine breite Anwendung im Internet scheiterte jedoch bislang an der Komplexität von SGML. Die 1998 veröffentlichte, wesentlich kompaktere Untermenge XML 1.0 von SGML läßt dagegen eine weite Verbreitung erwarten [121].

Um das Prinzip von XML zu verdeutlichen, wird wiederum das einfache Datenmodell aus Abschnitt 2.2 herangezogen. Beispiel 2.3 zeigt die DTD analog zum EXPRESS-Schema aus Beispiel 2.1 mit den XML-Elementen *traeger* und *rechteck_wand*. In Beispiel 2.4 ist die entsprechende Austauschdatei als XML-Dokument instanziiert. Die Attributwerte sind dort jeweils zwischen den in <- und />-Zeichen eingeschlossenen Attributnamen angegeben, den sogenannten *Tags*, von denen sich der bereits oben genannte Name *Tagged Document* ableitet.

Beispiel 2.3: Dokumenttyp-Definition in XML

```
<?xml version="1.0"?>
<xml_beispiel>

  <!ATTLIST traeger
            id          ID          #REQUIRED
            bezeichnung CDATA      #REQUIRED
            position    CDATA      #REQUIRED
            profil      CDATA      #REQUIRED
            laenge      CDATA      #REQUIRED>

  <!ATTLIST rechteck_wand
            id          ID          #REQUIRED
            bezeichnung CDATA      #REQUIRED
            position    CDATA      #REQUIRED
            dicke       CDATA      #REQUIRED
            laenge      CDATA      #REQUIRED
            hoehe       CDATA      #REQUIRED>

</xml_beispiel>
```

Beispiel 2.4: Austauschdatei als XML-Dokument

```
<?xml version="1.0"?>
<xml_beispiel>

  <traeger>
    <id>_1</id>
    <bezeichnung>Deckentraeger</bezeichnung>
    <position>POS1</position>
    <profil>HE300A</profil>
    <laenge>7500.0</laenge>
  </traeger>

  <rechteck_wand>
    <id>_2</id>
    <bezeichnung>Seitenwand</bezeichnung>
    <position>POS2</position>
    <dicke>250.0</dicke>
    <laenge>15000.0</laenge>
    <hoehe>2750.0</hoehe>
  </rechteck_wand>

</xml_beispiel>
```

Beim direkten Vergleich dieser Beispiele mit der ursprünglichen EXPRESS-basierten Lösung fallen folgende Dinge auf: Erstens verfügen die Instanzen im XML-Dokument nicht automatisch über Referenznummern (im Beispiel 2.2 sind dies die #-Nummern). Diese müssen also jeweils als explizite Attribute *id* in den XML-Elementen deklariert werden. Zweitens gibt es in XML keine direkte Entsprechung zur Vererbung. Aus diesem Grunde sind die Attribute des ursprünglichen Obertyps *bauteil* direkt in den Untertypen *traeger* und *rechteck_wand* deklariert worden. Drittens ist XML eine dokumentenorientierte Sprache. Das heißt, daß beispielsweise keine numerischen Basistypen zur Verfügung stehen. Daher sind alle zur Längenangabe verwendeten Attribute als Zeichenketten definiert (Typ *CDATA*). Viertens fällt auf, daß die XML-Datei bereits bei diesem einfachen Beispiel größer ist als die STEP-Datei. Bei komplexeren Modellen wächst der Umfang der XML-basierten Dateien sogar auf die drei- bis fünffache Größe an [82].

Zur Lösung des zweiten und dritten Punktes existiert jedoch bereits der weiterführende Standard *XML-Schema* [122]. Darin wird die DTD durch eine sogenannte *XML-Schema-Definition (XSD)* ersetzt, wodurch Erweiterungen um Datentypen und strukturelle Konzepte möglich sind. Dieses Schema ist konsequenterweise selbst in XML geschrieben. Für die genauere Betrachtung dieses sehr vielversprechenden Konzeptes, die den Rahmen dieser Arbeit sprengen würde, ist im Schrifttum eine Sammlung von Verweisen angegeben [76].

Liebich vergleicht die XML- und die EXPRESS-basierte Lösung im Hinblick auf die Produktdatenmodellierung [82]. Für XML sprechen dabei vor allem die Tatsachen, daß es sich um eine von der Software-Industrie mittlerweile bevorzugte Technologie handelt, daß deshalb viele Programmierwerkzeuge kostengünstig verfügbar sind und daß sich XML leicht in Standard-Software und Browser integrieren läßt. Die Vorteile der EXPRESS-basierten Lösung sind dagegen die im oben angeführten Beispiel aufgezeigte geringere Dateigröße und die größere Übertragungssicherheit bei komplexen Daten. Außerdem sind EXPRESS und seine Implementierungsmethoden speziell auf den Produktdatenaustausch ausgerichtet. Dies betrifft insbeson-

dere die in ISO 10303 ausdrücklich vorgesehene Möglichkeit zur Datenbankanbindung.

Diese Vor- und Nachteile lassen erwarten, daß die Bedeutung von XML für die Produktdatenmodellierung hauptsächlich in der Übertragung eher kleinerer Datenmengen im Internet liegen wird. Dies betrifft zum Beispiel Anwendungen für Ausschreibung, Vergabe und Abrechnung sowie den internetbasierten Handel. Für den klassischen Austausch komplexer Produktmodelldaten ganzer Bauwerke oder -projekte zwischen leistungsfähiger Ingenieursoftware sind die EXPRESS-basierten Lösungen jedoch besser geeignet.

Die Bedeutung von XML wurde auch in der STEP-Bewegung erkannt. Der im Entwurfsstadium befindliche Teil 28 von ISO 10303 bietet Werkzeuge zur nachträglichen Überführung sowohl von EXPRESS-Schemata in XML-Schema-Definitionen als auch von STEP-Austauschdateien in XML-Dateien. Auch *Liebich* berichtet von einem Programm, mit dem diese Übertragung automatisch durchgeführt werden kann [82].

2.4 Bewertung

Um die für die weitere Arbeit am besten geeigneten der betrachteten Verfahren und Werkzeugen zur Produktmodellierung auszuwählen, muß zunächst der Ansatz von ISO 10303 als Ganzes bewertet werden.

ISO 10303 hat zweifellos eine Reihe sehr leistungsfähiger und richtungsweisender Errungenschaften hervorgebracht. Die vorgeschriebenen Regeln zur Modellierung und Implementierung werden weltweit angewandt und haben den Einsatz vielfältiger Programmierwerkzeuge ermöglicht. Das Prinzip der absoluten Unabhängigkeit von Hard- und Software sowie das der konsequenten Trennung der Bestandteile von Produktmodellen haben sich ebenfalls durchgesetzt.

Seit Einführung der Norm haben sich jedoch einige gravierende Nachteile bezüglich der Anwendbarkeit des Gesamtkonzeptes gezeigt. Zuerst ist der Umfang der Norm zu nennen. Derzeit existieren 56 Teile, weitere 37 sind im Entwurfsstadium begriffen [ISO10303b]. Bei der Entwicklung der einzelnen Teile wurde die noch zu Beginn geltende Grundforderung nach der Minimierung der Zahl der Entities [44] offensichtlich nicht mehr berücksichtigt. Außerdem hat der sehr formale Ansatz, der alle Teile der Norm prägt, dazu beigetragen, daß der Aufwand für die Implementierung stark angewachsen ist. Ein weiterer Nachteil ist das unflexible und zeitaufwendige Verfahren zur Etablierung und Weiterentwicklung der Application Protocols, das den rasanten Fortschritten im Bereich der Informationstechnologie und der Software-Entwicklung nicht gerade gerecht wird.

Schließlich muß die Möglichkeit zur Einordnung größerer Teile des Bauwesens in das Gesamtkonzept der Norm eher kritisch beurteilt werden: Es ist kaum möglich, daß sich der ganze Industriezweig in einem einzigen AP darstellen läßt. Die im Bauwesen besonders stark ausgeprägten Abhängigkeiten und Verkettungen aller Prozesse lassen sich jedoch zwischen mehreren APs mit den überwiegend geometriebasierten Ressourcen des Applikationsdomänenmodells ohne wirklichen Modellkern oder Metamodell nur unzureichend darstellen.

Aus diesen Gründen werden folgende Schlüsse für die weitere Arbeit gezogen:

- 1) Die Verwendung der Modellierungswerkzeuge aus ISO 10303 ist für die Definition des Komplettbaumodells sinnvoll, wodurch alle Formen der Implementierung zur Verfügung stehen. Die Alternative, die Modellierung in XML, soll aufgrund ihrer im vorangegangenen Abschnitt genannten Nachteile bei der Produktdatenmodellierung nicht verwendet werden. Falls sich XML als formale Spezifikationssprache in der Produktmodellierung dennoch durchsetzen sollte, kann das EXPRESS-Modell nachträglich in ein XML-Schema überführt werden.
- 2) Das Ziel, das Komplettbaumodell langfristig als Teil von ISO 10303 zu etablieren, wird nicht verfolgt.
- 3) Die in Abschnitt 2.2.6 genannten Bestandteile der Application Protocols enthalten alle zur Definition eines Produktmodells notwendigen Vorüberlegungen. Die formale Art und Weise, die ISO 10303 für deren Formulierung vorschreibt, ist jedoch im Rahmen dieser Arbeit zu aufwendig. Deshalb sollen diese Vorüberlegungen vereinfacht und nicht formal, dafür aber möglichst anschaulich formuliert werden.
- 4) Die Frage, ob die Ressourcen aus ISO 10303 verwendet werden sollen, hängt davon ab, ob das zu entwickelnde Modell sich an anderen, bereits existierenden Produktmodellen für das Bauwesen orientieren soll, und ob diese wiederum konform zu ISO 10303 sind oder nicht. Diese Frage läßt sich deshalb erst nach einem Vergleich der für das Bauwesen relevanten Produktmodelle beantworten, der im nächsten Kapitel durchgeführt wird.
- 5) Eine objektorientierte Modellierung und eine Modellarchitektur mit gemeinsamem Modellkern und Metamodell ist zu bevorzugen. Letztlich hängt diese Entscheidung aber ebenfalls davon ab, ob das zu entwickelnde Modell sich an anderen Produktmodellen orientieren soll.

3 Software und Produktdatenaustausch im Bauwesen

3.1 Allgemeines

Der Erfolg eines Produktmodells hängt von der Akzeptanz durch die Software-Hersteller ab. Dafür ist es wesentlich, daß das Umfeld untersucht wird, in dem das Modell angewandt werden soll. Deshalb werden zunächst die Fähigkeiten der zur Verfügung stehenden Software betrachtet, was aber nicht bedeutet, daß das Modell ausschließlich auf die Belange der Software-Industrie zugeschnitten sein darf. Anschließend werden die bereits existierenden Produktmodelle des Bauwesens im Hinblick auf eine mögliche Harmonisierung untersucht, und danach werden verschiedene andere Modelle zur Unterstützung des Produktdatenaustausches skizziert. Schließlich wird die tatsächliche Nutzung dieser Möglichkeiten in der Baupraxis untersucht.

3.2 Software im Bauwesen

3.2.1 Allgemeines

Mit Software werden in diesem Zusammenhang nicht nur Werkzeuge für die klassischen Ingenieuraufgaben bezeichnet, sondern auch solche Software, die zur Abwicklung von Projekten und auch zur Abwicklung des Datenaustausches selbst eingesetzt wird. Es wird hier jedoch weder auf die zugrundeliegenden Informations- und Kommunikationstechnologien noch auf Anwendungs-Software für Buchhaltung, Textverarbeitung oder Tabellenkalkulation eingegangen. Den Ausführungen liegen unter anderem eine Untersuchung des Deutschen Stahlbau-Verbandes [33] sowie eine Recherche im Rahmen des AiF-Forschungsprojektes „Weiterentwicklung der Produktschnittstelle Stahlbau für den Komplettbau“ zugrunde [57]. Folgende, grundlegende Tendenzen waren in letzter Zeit zu beobachten:

Erstens ist durch die Vorteile moderner, objektorientierter Programmiertechnologien eine große Zahl neuer Produkte und Anbieter am Markt erschienen, oder die Hersteller bestehender Systeme haben deren Funktionsumfang zum Teil erheblich erweitert. Die klassischen Aufgaben des konstruktiven Ingenieurbaus wie Entwurf, Konstruktion, Zeichnung und Berechnung von Tragwerken können für sich alleine genommen heutzutage weitestgehend mit Computerunterstützung bearbeitet werden. Im Vergleich zum Stand von vor einigen Jahren ist jetzt die Situation entstanden, daß sich der Funktionsumfang von vielen Programmen ganz unterschiedlicher Herkunft deutlich überschneidet. Ein Beispiel dafür sind die leistungsfähigen CAD-Systeme für Architektur und Ingenieurkonstruktion.

Zweitens versuchen Softwareanbieter verstärkt, ein immer größeres Aufgabenspektrum mit mehreren kooperierenden Programmen aus einer Hand anzubieten, oder mehrere Anbieter schließen sich zu strategischen Allianzen zusammen. Die Software-Bausteine kommunizieren dabei über EDV-Schnittstellen. Für die derart gewachsenen Programmsysteme stellen sich neue Probleme, die über die Bearbeitung in sich geschlossener Ingenieuraufgaben hinausgehen und zunehmend die ganzheitliche Projektbearbeitung betreffen. Dieser Aspekt hat durch die sich jüngst bietenden

Möglichkeiten der neuen Informations- und Kommunikationstechnologien zusätzlich stark an Bedeutung gewonnen.

Drittens sind moderne Computersysteme durch den Preisrückgang, der gleichzeitig zur Leistungssteigerung stattgefunden hat, auch für kleine Büros und zum Teil sogar für Handwerksunternehmen rentabel geworden. Die Verbreitung interaktiver, graphischer Oberflächen erleichtert zudem das Erlernen der Programme.

3.2.2 Stand der Technik

Im folgenden sind Software-Anwendungen aufgelistet, die dem Bauschaffenden im Bereich des Komplettbaus und den angrenzenden Gebieten des Bauwesens heutzutage zur Verfügung stehen. Die Oberbegriffe sollen nicht als starre Typisierung verstanden werden, da die Grenzen zum Teil fließend sind. Das Hauptinteresse für diese Untersuchung gilt der Frage, welche Produktdaten die verschiedenen Programme im gedachten Falle einer Systemkopplung untereinander austauschen könnten, und welche Stellung die Programme dabei einnehmen würden.

Computer Aided Draughting oder Computer Aided Design (CAD):

In bezug auf den Produktdatenaustausch sind im Gegensatz zu reinen Zeichnungsprogrammen nur solche CAD-Programme interessant, die mit einem „intelligenten“, internen Objektmodell arbeiten: Der Anwender kann zum Beispiel Bauteile verlegen und diesen Querschnitt und Material aus Bibliotheken zuordnen. Die oft benutzte Unterscheidung in ebene und räumliche Programme ist hierfür irrelevant, auch wenn die meisten dieser Objektmodelle räumlich sind. Reine Zeichenprogramme haben jedoch insofern eine Bedeutung, als einige dieser Systeme über moderne Programmierschnittstellen verfügen, durch die intelligente Applikationsprogramme für spezielle Ingenieurzwecke „aufgesetzt“ werden können. Diese aufgesetzten Applikationen sind vergleichsweise preiswert und bieten dem Anwender den Vorteil, daß er die Funktionen und die Oberfläche eines verbreiteten CAD-Systems nutzen kann.

Spezielle CAD-Programme für den Stahlbau werden bereits seit zwanzig Jahren entwickelt. Ihr Nutzen bei Entwurf und Konstruktion im Stahlbaualltag ist allgemein anerkannt und vielfach belegt [103], [80]. Für den Komplettbau von Bedeutung ist die Tatsache, daß einige dieser Systeme in letzter Zeit Funktionen zur Konstruktion von Dach- und Wandbekleidungen sowie den Glasbau erhalten haben. Vergleichbare Systeme existieren für den Holzbau und zur Schal- und Bewehrungsplanung im Stahlbetonbau. Aus den internen Objektmodellen der Programme werden ebene Werkstatt- und Montagepläne automatisch abgeleitet. Außerdem können Stücklisten erzeugt oder Fertigungsmaschinen gesteuert werden.

Im Bereich der Architektur werden mit sogenannten CAAD-Programmen (*Computer Aided Architectural Design*) architektonische Objekte wie Dächer, Wände, Decken, Fenster und Türen in einem räumlichen Gebäudemodell angeordnet. Durch topologische Beziehungen können diese Elemente Gebäudeabschnitte, Geschosse und Räume bilden. Man spricht in diesem Zusammenhang von der *Gebäudetopologie*. Alle Genehmigungs-, Ausführungs- und Werkpläne sowie Massen, Flächen und Volumina zur Kostenschätzung können aus dem Modell abgeleitet werden. Neuere Programme verfügen außerdem zum Teil über Funktionen für den Stahl- und Komplettbau in niedrigem Detaillierungsgrad. Dazu stehen Objekte wie Wände aus Trapez-, Well-, und Sandwichprofilen, Pfosten-Riegel-Konstruktionen, Ganzglasfassaden, Dachgestaltungen, Dachfenster, Lichtbänder oder Lichtkuppeln zur Verfügung. Von allen Computerprogrammen stellen die Objektmodelle von CAD-Systemen ein Bauwerk am umfassendsten dar. Im Falle einer Systemkopplung könnten die mei-

sten der anderen Programme die von ihnen benötigten Eingangsdaten aus diesen CAD-Modellen ableiten, so daß den CAD-Programmen eine zentrale Rolle zukäme. Wenn sie zudem über ein Änderungsmanagement für den Planbestand verfügen, ist eine einfache Form der computerintegrierten Planung auch ohne Einsatz von Projektdatenbanken möglich, wie *Pegels* [90] und *Hahn* [46] berichten.

Steuerungssoftware für Fertigungsmaschinen (NC-Fertigung):

NC-Fertigungsmaschinen (NC – *Numerically Controlled*) werden in Stahl- und Holzbaubetrieben seit Jahren eingesetzt. Ihre Steuerungssoftware kann über EDV-Schnittstellen Daten zur Bauteilbearbeitung von entsprechenden CAD-Programmen übernehmen. Diese Systemkopplung, die ebenfalls seit langem Stand der Technik ist, wird als CAD-CAM-Kette bezeichnet (CAM – *Computer Aided Manufacturing*). Über den klassischen Stahlbau hinausgehend werden NC-Maschinen zur Bearbeitung von Profilen im Fensterbau und für stabförmige, dünnwandige Teile des Stahlleichtbaus eingesetzt. Für die raumbildenden Bauteile von Dach- und Wandbekleidungen hat die NC-Technologie jedoch keine Bedeutung, da hier die überwiegend fertigen Bauprodukte in den benötigten Abmessungen bestellt werden. Der Feinzuschnitt kann gegebenenfalls von Hand auf der Baustelle geschehen.

Software zur Produktionsplanung und -steuerung (PPS):

Für die Produktionsplanung und -steuerung im Stahlbau stehen Programme zur Längenoptimierung und zur Ausarbeitung von Schachtelplänen für die NC-Fertigung zur Verfügung. Im Falle einer Systemkopplung können sie detaillierte Informationen aus dem im CAD-Programm des Fachplaners erzeugten Objektmodell übernehmen. Dies wird über herkömmliche Schnittstellen bereits heutzutage praktiziert. Für Dach- und Wandbekleidungen können spezielle CAD-Programme die Verlegepläne hinsichtlich des Verschnitts und hinsichtlich der Palettierung optimieren.

Computer Aided Facility Management (CAFM):

Das *Facility Management* gewinnt mit der Betrachtung ganzer Lebenszyklen in der Gebäudeplanung zunehmend an Bedeutung. Man bezeichnet damit alle Aspekte der Betreuung des Gebäudezustandes unter technischen, kaufmännischen und organisatorischen Gesichtspunkten, also die Bewirtschaftung, Bauunterhaltung, Umbau, Sanierung, Umzug, Verwaltung und Betriebstechnik betreffend. Ziel des Facility Managements ist die Senkung der Betriebs- und Instandhaltungskosten. CAFM-Software basiert entweder auf Datenbankprogrammen oder CAD-Systemen. Sie kann an CAD-Programme gekoppelt werden, indem sie deren Objektmodelle übernimmt.

Software für bauphysikalische Berechnungen:

Mit bauphysikalischen Programmen können Nachweise der Dampfdiffusion und des Schallschutzes geführt oder der Heizwärmebedarf von Gebäuden, die Wärmedurchgänge in Bauteilen sowie die Wärmeströme an Anschlußdetails berechnet werden. Im Falle einer Systemkopplung benötigen diese Programme detaillierte Informationen über den Aufbau der raumabschließenden Bauteile als Eingangsdaten.

Software für die technische Gebäudeausrüstung:

Der Begriff der technischen Gebäudeausrüstung bezeichnet die Gewerke für Heizung, Sanitär, Lüftung und Elektroinstallation. Software für diese Gewerke sind entweder CAD-Programme für den Entwurf und die Planung der Installationssysteme

me einschließlich des Verlaufs von Versorgungsleitungen oder Berechnungsprogramme zur Dimensionierung dieser Systeme. Derartige CAD-Programme benötigen bei einer Kopplung an Komplettbau-Software die Modelldaten des Rohbaus einschließlich Tragwerk in grobem Detaillierungsgrad. Falls Durchbrüche für den Leitungsverlauf in tragenden Elementen angeordnet werden, müssen diese Daten zurück an den Tragwerksplaner gegeben werden.

Stabwerks- und Finite-Elemente-Programme (FE-Programme):

Stabwerksprogramme zur Berechnung von Verformungen und Schnittgrößenverläufen statischer Systeme gehören zu den am längsten und am häufigsten eingesetzten Computerprogrammen im konstruktiven Ingenieurbau. Mittlerweile bieten fast alle Hersteller Module für räumliche Stabwerke mit interaktiver, graphischer Oberfläche an. Moderne Stabwerksprogramme verfügen außerdem über Funktionen zur automatischen Nachweisführung für Holz-, Stahl- und Stahlbetonbauteile nach unterschiedlichen Normen. *Krüner* hat gezeigt, daß Stabwerksprogramme und CAD-Programme für den Tragwerksentwurf zur Datenübertragung und Modellbildung gekoppelt werden können [79]. Die Erzeugung vollständiger statischer Systeme aus CAD-Konstruktionen, mit der sich *Song* beschäftigt hat [109], ist jedoch nur unter interaktiver Einbeziehung des Tragwerksplaners sinnvoll.

Jünger sind die aus dem Maschinenbau stammenden FE-Programme, mit denen sich zum Beispiel Verformungen und daraus abgeleitete Werte wie Schnittgrößen und Spannungen auch an Flächentragwerken und Körpern berechnen lassen. FE-Programme können aber auch bei wärme- und strömungstechnischen Problemen eingesetzt werden. Anfangs wurden die zunächst schwierig zu bedienenden Programme lediglich für komplizierte Flächentragwerke oder komplexe Anschlußberechnungen eingesetzt, zumeist für Forschungsarbeiten. Mittlerweile ist erschwingliche und einfach zu bedienende Software für spezielle praktische Aufgaben im konstruktiven Ingenieurbau verbreitet: Ebene Flächentragwerke des Stahlbeton- und Glasbaus sowie Rotationsschalen im Behälterbau. Wie *Klinkenberg* gezeigt hat, ist eine Kopplung mit CAD-Systemen ebenfalls möglich, wenn der Anwender interaktiv miteinbezogen wird [78].

Software zur Visualisierung und Animation:

Unter Visualisierung versteht man die photorealistische Darstellung von räumlichen CAD-Modellen durch Ausblenden verdeckter Kanten, Schattenwurf, Arbeiten mit Lichtquellen und Darstellen von Oberflächeneigenschaften verschiedener Materialien. Den größten Stellenwert hat diese Technik in der Architektur als digitales Arbeits- und Präsentationsmodell, auch wenn dies bei Architekten nicht unumstritten ist. Allerdings haben mittlerweile auch die meisten handelsüblichen CAD-Systeme für Ingenieurkonstruktionen, ja sogar einige Stabwerksprogramme grundlegende Visualisierungsfunktionen. Laut *Pegels* trägt dies zur besseren Übersichtlichkeit bei komplexen Konstruktionen und zur Erhöhung der Auftragsklarheit durch Präsentationen beim Bauherrn bei [90].

Für höhere Ansprüche und insbesondere, wenn bewegte Animationen des Gebäudemodells gefragt sind, existieren spezielle Programme. Sie können an CAD-Systeme des Architekten oder des Ingenieurs gekoppelt werden, indem sie deren räumliche Objektmodelle übernehmen.

Software für Ausschreibung, Vergabe und Abrechnung (AVA):

Software für Ausschreibung, Vergabe und Abrechnung wird in Bauunternehmen bereits ebenso lange verwendet wie CAD-Programme. Kernstück eines AVA-Programms ist eine Datenbank, in der Preise, Mengen, Leistungsbeschreibungen und gegebenenfalls Skizzen zu den einzelnen Positionen eines Bauvorhabens abgelegt sind. Aus diesen Daten können Kostenberechnungen, Leistungsverzeichnisse und Rechnungen erstellt werden. Über standardisierte Schnittstellen werden diese Programme bereits heutzutage an CAD-Systeme gekoppelt, aus deren Objektmodellen sie Mengen, Massen, Flächen und Volumina ermitteln.

Projektmanagementsysteme (PMS):

Der Ansatz von Projektmanagementsystemen geht über den der bloßen AVA-Programme hinaus: Nach *Pflugbeil* beherrschen PMS außer den oben beschriebenen Funktionen für Ausschreibung, Vergabe und Abrechnung die gesamte zeitliche und kostenmäßige Abstimmung des Bauvorhabens [92]. Dazu verfügen sie über Analyse-, Optimierungs-, Kontroll- und Dokumentationsmechanismen. Wenn sie an CAD-Systeme gekoppelt werden könnten, würden sie die AVA-Daten aus deren Objektmodellen übernehmen. Zusätzlich dazu müßten sie jedoch Daten über den Planungs- und Fertigungsfortschritt der Gewerke erhalten und könnten Informationen zur aktiven Steuerung dieser Prozesse zurückgeben. Ein Überblick über kommerzielle PMS kann bei der im Schrifttum angegebenen Quelle bezogen werden [93].

Informationsmanagementsysteme (IMS):

Als Informationsmanagement im engeren Sinne³ wird die Ablage, Verwaltung und Verteilung aller in einem Unternehmen anfallenden Daten bezeichnet. Dies betrifft neben Schriftverkehr (also Dokumenten im eigentlichen Sinne) und Zeichnungen auch die Geschäftspartner-, Projekt- und Auftragsverwaltung sowie die Revisionsverwaltung der Dokumente und Zeichnungen. Ein IMS basiert auf einer Datenbank und ist zentral auf einem Server (*Single Point of Information*) installiert. Von jedem Dokument und jeder Zeichnung existiert im Unternehmen nur eine einzige Kopie, die auf diesem Server liegt, und auf die jeder Mitarbeiter über ein Netzwerk gemäß festgelegter Berechtigungen zugreifen kann. Zusätzlich zur Datenhaltung muß individuell für jedes Unternehmen die automatische Verteilung und Zugriffsberechtigung für diese Daten programmiert werden (*Workflow*). Die Dokumente und Zeichnungen müssen dazu in digitaler Form vorliegen; das Format spielt für das IMS jedoch keine Rolle, da es die primären Daten gar nicht auswertet. Ausgewertet werden lediglich die sogenannten Metadaten, die allen in das System eingehenden Dateien angehängt werden.

Dieses Modell der Informationsverteilung kann mit Hilfe des Internets auch auf Projektebene mit Partnern aus mehreren Unternehmen übertragen werden. Entsprechende Systeme verwenden dazu Standardtechnologien, die jedem Anwender den Zugang über seinen Standard-Internetbrowser und die Verwendung seiner angestammten Kommunikationsprogramme (zum Beispiel E-Mail) ermöglichen. Die dadurch gegebene Flexibilität, die vergleichsweise geringe Hemmschwelle für die Erstbenutzung und, nicht zuletzt, die universelle Anwendbarkeit – es werden eben wirklich alle Dateiformate behandelt – haben nach *Suchocki* bereits in kurzer Zeit zu beacht-

³ Man spricht in diesem Fall auch von Dokumentenmanagementsystemen (DMS) oder Engineering Data Management Systems (EDMS).

licher Verbreitung dieser neuen Systeme geführt [114]. *Björk* bezeichnet sie zu Recht als Schlüsseltechnologie für das Bauwesen, die in der computerintegrierten Planung eine zentrale Stellung einnehmen wird [7]. Ihr Nutzen liegt jedoch lediglich in der Verteilung von Informationen oder auch von Produktdaten. Eine Kopplung der unterschiedlichen Anwendungsprogramme wird durch sie jedoch nicht ermöglicht.

Bauinformationssysteme:

Über Bauinformationssysteme können Informationen zu bestimmten Themengebieten recherchiert werden. In Baudatenbanken sind beispielsweise Herstellerinformationen und Standardleistungsbeschreibungen von Bauprodukten abgelegt, so daß Architekten und Fachplaner Produktrecherchen und Online-Ausschreibungen durchführen können. In neuester Zeit sind diese Systeme über das Internet zugänglich. Wie IMS sind Bauinformationssysteme nicht auf standardisierte Formate angewiesen, da sie selber die Informationen nicht auswerten.

Internetbasierter Handel (E-Commerce):

Der Begriff *E-Commerce* wird als Oberbegriff für die Abwicklung von Geschäften zwischen Unternehmen (*B2B = Business to Business*), zwischen Unternehmen und ihren Lieferanten (*E-Business*) sowie zwischen Unternehmen und Endkunden (*B2C = Business to Customer*) verwendet [2]. Dazu wird ein „virtueller Marktplatz“ von einem Dienstleister zur Verfügung gestellt, auf dem sich Anbieter von Produkten oder Dienstleistungen registrieren lassen. Potentielle Kunden können online über diese Angebote verfügen und Geschäftstransaktionen direkt abwickeln.

Die Internet-Euphorie der letzten Jahre wurde auch von der Bauindustrie aufgegriffen: Das B2B-Branchenverzeichnis BizScout (<http://www.bizscout.de>) führte im Februar 2001 bereits etwa siebzig Online-Anbieter von Baustoffen und Bauprodukten für den deutschen Markt. Viele E-Commerce-Dienstleister für das Baugewerbe bieten außerdem Funktionen zur Kommunikation zwischen Projektpartnern sowie für Ausschreibung und Vergabe, so daß die Grenzen zu den oben genannten IMS und Bauinformationssystemen fließend sind.

Schmitt sah bereits vor einiger Zeit die Möglichkeit voraus, das gesamte Ausschreibungs- und Vergabewesen von Aufträgen internetbasiert abzuwickeln oder ganze Bauwerke über das Internet kaufen und verkaufen zu können [105]. Dies ist jedoch, zumindest in Deutschland, bisher nicht bestätigt worden: Im April 2001 wurden im B2B-Bereich in Deutschland lediglich ein Prozent der Bestellungen über das Internet getätigt [2]. Im konservativen Baugewerbe dürfte die Zahl noch wesentlich geringer sein.

3.2.3 Ausblick

Der beschriebene Entwicklungsstand von Bau-Software zeigt deutlich, daß für nahezu alle Bereiche des Bauwesens leistungsfähige Computer-Werkzeuge bereits existieren oder in der nächsten Zeit entwickelt werden.

Natürlich besteht Bedarf für Weiterentwicklungen: *Stein* nennt beispielsweise praxisingerechtere FE-Programme zur Lösung der immer komplexeren mechanischen Probleme im Zusammenhang mit anspruchsvoller Architektur oder mit komplizierten Verbundwerkstoffen [113]. *Fischer* arbeitet an Visualisierungssoftware mit einer zeitlichen Komponente (*4D-Visualisierung*) zur Simulation von Bauabläufen [42]. Für CAD-Programme, die bereits jetzt eine Schlüsselrolle im Planungsprozeß inne haben, ist zu erwarten, daß einige dieser Systeme zukünftig über erweiterte Funktio-

nen zur Projektkoordination verfügen werden, sei es mit oder ohne Verwendung zentraler Projektdatenbanken [90], [46]. Nach *Pegels* fehlen dazu jedoch noch rechnergestützte Bauingenieurmethoden für automatisches Management und automatische Qualitätssicherung bei verteiltem, kooperativem Arbeiten: Flexible Workflow-Techniken und intelligente Software-Agenten⁴ zur Konsistenzsicherung von Projektdatenbanken durch Fortschritts- und Änderungsmanagement, Benachrichtigung, Freigabe- und Revisionsmanagement mit Kennzeichnung und Aktualisierung [90].

Eine noch bedeutendere Tendenz ist die im vorangegangenen Abschnitt bereits ange deutete Auslegung vieler Software-Systeme auf das Arbeiten in Netzwerken. Dem wird im einfachsten Fall mit der Unterstützung internetfähiger Dateiformate Rechnung getragen: Große CAD-Systeme bieten bereits sehr komprimierte Zeichnungsdateien an, die von Internetbrowsern mit beschränktem Funktionsumfang angezeigt, jedoch nicht bearbeitet werden können. Derartige Formate sind besonders nützlich in Zusammenhang mit internetgestützten IMS.

Für bestimmte Anwendungen ist darüber hinaus eine räumliche Verteilung der Systeme selbst möglich. Dies geschieht entweder, indem die Softwarekomponenten selbst verteilt werden, zum Beispiel in der von *Bretschneider* gezeigten Variante mit einer einheitlichen Software-Architektur [8], oder durch Anwendung der Agententechnologie⁴, bei der kleine Software-Einheiten selbständig in Netzwerken arbeiten. *Schnellenbach-Held* benutzt Agenten beispielsweise für Aufgaben in der Ausschreibungs- und Angebotsphase eines Projektes [106]. Die dritte Möglichkeit besteht darin, daß der Anwender statt eigener Software lediglich Dienste von einem Server in Anspruch nimmt, der sich nicht einmal im eigenen Unternehmen befinden muß. Dafür spricht neben dem Wegfall von Anschaffungs- und Wartungskosten vor allem die Tatsache, daß die räumliche Verteilung der Systeme der starken Verteilung der beteiligten Planer im Bauwesen besonders gut gerecht wird. Die tatsächliche Verbreitung solcher Systeme wird jedoch wesentlich davon abhängen, ob sie dem Anwender innerhalb der teilweise festgefügtten Arbeitsabläufe in der Baupraxis unmittelbar und subjektiv sichtbaren Nutzen versprechen. Dies gilt übrigens auch für die im vorangegangenen Abschnitt zitierte Entwicklung im Bereich des internetbasierten Handels.

Vor dem Hintergrund dieser Vielfalt an Programmen und den vorhandenen Möglichkeiten durch die modernen Informations- und Kommunikationstechnologien stellt sich einmal mehr die Frage nach der Integration dieser Systeme. Auf Unternehmens- und Projektebene wird sie zum Teil bereits durch die IMS-Technologie beantwortet, deren Bedeutung zukünftig wohl noch größer werden wird. Dabei werden die Grenzen zwischen IMS, PMS, AVA-Software, Bauinformationssystemen und internetbasiertem Handel verschwimmen. *Björk* prognostiziert, daß sich einige große, multifunktionale Internet-Portale⁵ im Bauwesen durchsetzen werden [7]. Dennoch, ohne eine inhaltliche Standardisierung der auszutauschenden Daten können auch sie eine wirkliche Integration aller Software-Systeme nicht leisten.

⁴ Ein Agent ist eine Software-Einheit, die im Auftrag des Nutzers Anfragen im Netz durchführt und die Antworten bereitstellt.

⁵ Als Portal wird die Einstiegsseite eines Internet-Dienstleisters bezeichnet.

3.3 Produktmodelle des Bauwesens

3.3.1 Allgemeines

Dieser Abschnitt gibt einen Überblick über die Produktmodelle des Bauwesens. Dies soll im Hinblick auf die Vermeidung einer Parallelentwicklung und, wie in Kapitel 2 bereits angesprochen, hinsichtlich der Möglichkeit zur Harmonisierung geschehen. Kriterien für diese Möglichkeit sind in erster Linie die Eigenschaften des Modells, die Dokumentation, Aspekte der Implementierung und Methoden zur Konformitätsprüfung. In zweiter Linie werden dazu aber auch die tatsächliche oder die zu erwartende Akzeptanz durch Software-Hersteller und Anwender betrachtet.

Allen im folgenden aufgeführten Modellen ist gemein, daß sie zumindest die Technologien für Modellierung und Implementierung nach ISO 10303 benutzen. Damit können bei diesen Modellen die Möglichkeiten zur Implementierung genutzt werden, die sich durch moderne Programmierwerkzeuge bieten, und für jedes der Modelle ist eine Datenbankimplementierung grundsätzlich möglich.

3.3.2 Application Protocol 225 (AP 225)

Das *Application Protocol 225: Building Elements Using Explicit Shape Representation* ist als Teil 225 von ISO 10303 definiert [75]. Es ist ein eigenständiges Produktmodell und ermöglicht den Produktdatenaustausch von CAD-Daten für Gebäude und Gebäudeelemente mit *expliziter Geometriedarstellung*. Das bedeutet im Gegensatz zur *parametrischen Darstellung*, daß die Form der Teile ausschließlich über geometrische Grundobjekte wie Scheitelpunkte (kleinste, punktförmige Grundobjekte), Polygonzüge, Flächen und Volumenkörper definiert ist.

AP 225 verfügt neben Sprachmitteln zur Geometriedarstellung auch über solche zur Beschreibung von Gebäudemodellen. Die folgenden Gebäudeelemente stehen zur Verfügung: Balken, Aussteifungen, Stützen, Deckenplatten, Fundamente, tragende Seile, Gebäudekerne und Wände. Sie können Eigenschaften besitzen wie Anerkennungsstatus im Planungs- und Genehmigungsprozeß, Öffnungen oder Materialien. Das gesamte Gebäudemodell kann in Gebäudeabschnitte und Stockwerke gegliedert werden. Es können außerdem allgemeine Gebäudeinformationen wie Beschreibung des Bauvorhabens, Bearbeiter, Bauherr und Bauortbeschreibung angegeben werden.

AP 225 ist als einziges der hier untersuchten Produktmodelle völlig STEP-konform. Das heißt, daß es ausschließlich STEP-Ressourcen für Material- und Geometriedarstellung benutzt und die formalen Anforderungen an ein AP erfüllt. Die Dokumentation liegt in englischer Sprache vor und enthält die nach ISO 10303 geforderten abstrakten Testfälle zur Konformitätsprüfung. Von offiziellen Testbeispielen oder von einer die Konformitätsprüfung durchführenden Institution ist mir nichts bekannt. Das Anwendungsgebiet betrifft in erster Linie CAD- und CAAD-Programme. Von Implementierungen des Modells in kommerzieller Software ist mir ebenfalls nichts bekannt.

Bewertung:

Die Vorzüge von AP 225 sind in erster Linie der Status als Teil der internationalen Norm ISO 10303 und die erschöpfenden Möglichkeiten zur expliziten Geometriedarstellung. Da die Darstellung der Gebäudeelemente jedoch überwiegend geometriebasiert ist, und somit vorrangig auf CAD-Systeme ausgelegt ist, ist die Anwendbarkeit des Modells für andere Planungstätigkeiten als die Konstruktion jedoch eher fraglich. Hinzu kommt für solche Disziplinen des konstruktiven Ingenieurbaus wie

den Holz-, Stahl-, Stahlbetonfertigteile- und den Metallbau, in denen Bauteile vorwiegend parametrisch beschrieben werden, daß diese Art der Darstellung in AP 225 ausdrücklich nicht vorgesehen ist. Es kann nicht davon ausgegangen werden, daß eine entsprechende Erweiterung des Modells beim ISO durchzusetzen wäre. AP 225 ist also als Grundlage für das weitere Vorgehen in dieser Arbeit nicht geeignet.

3.3.3 Industry Foundation Classes (IFC)

Die *Industry Foundation Classes* (IFC) sind ein Produktmodell, das den Produktdatenaustausch zwischen allen weltweit am Bau beteiligten Software-Anwendungen ermöglichen soll. Die IFC werden von der *Internationalen Allianz für Interoperabilität* (IAI) entwickelt und betreut. Im Gegensatz zu AP 225 zielen sie nicht vorrangig auf CAD-Anwendungen ab, sondern schließen in der aktuellen Version *IFC2x-Platform* bereits die Disziplinen Architektur, Baubetrieb, technische Gebäudeausrüstung, Facility Management und die Gebäudeinneneinrichtung mit ein [59]. Weil die Integration weiterer Anwendungsgebiete geplant ist, ist das Datenmodell der IFC von vornherein auf flexible Erweiterung ausgerichtet. Um dies zu fördern, bietet die IAI sogar ein spezielles Handbuch für Modellerweiterungen an [60].

Abbildung 3.1 zeigt den streng hierarchischen, modularen Aufbau in vier funktionalen Schichten, den sogenannten *Layern*. Je weiter unten sich eine betrachtete Schicht in dieser Darstellung befindet, desto abstrakter sind die dort definierten Sprachmittel. Jedes der in der Abbildung dargestellten Elemente entspricht einem EXPRESS-Schema. Daraus läßt sich bereits erahnen, daß das Modell sehr umfangreich ist.

Im *Resource Layer* werden grundlegende Dinge wie Maßeinheiten, Geometrie etc. dargestellt, die von den höheren Schichten benutzt werden müssen. Als nächsthöhere Schicht bildet der *Core Layer* eine über längere Zeit unveränderliche Plattform, in der alle übrigen Modellteile verankert sind. Er muß deshalb ein äußerst breites Spektrum an Sprachmitteln zur Verfügung stellen. Dies beinhaltet neben Objekten auch Eigenschaften und Relationen zwischen den Objekten. Die Modellarchitektur entspricht also gemäß Abschnitt 2.1.3 der Variante mit gemeinsamem Modellkern und Metamodell. Im *Domain Layer* sind Teilmodelle für konkrete technische Anwendungsfelder zu finden. Über eine Zwischenschicht, den *Interoperability Layer*, sind diese Domänenmodelle auf den Modellkern sozusagen aufgesteckt. Dazu müssen sie die generischen Datenstrukturen des Modellkerns aufgreifen und sich an eine hierarchische Vorschrift zur Referenzierung halten. Die Objekte, die innerhalb der Domänen definiert sind, sind Spezialisierungen der allgemeineren Objekte des Kerns, so daß letzterer einen gemeinsamen Nenner zwischen den Domänen bildet. Diese Spezialisierung wird durch Vererbung erreicht.

Die Modularität dieser Architektur hat den Vorteil, daß die Domänenmodelle flexibel und unabhängig voneinander durch kleine, fachkundige Gruppen entwickelt werden können, während der gleichbleibende Kern die Kompatibilität der Domänen sichert. Der Produktdatenaustausch kann auf zwei Ebenen stattfinden: entweder innerhalb einer Domäne zwischen gleichartigen Software-Anwendungen auf vergleichsweise hohem Detaillierungsniveau oder zwischen mehreren Domänen und damit zwischen verschiedenartiger Software auf niedrigerem Detaillierungsniveau.

Die aktuelle Version der IFC verfügt in den unteren Schichten über eine Reihe allgemeingültiger Sprachmittel für Geometrie (diese sind sehr detailliert und überwiegend STEP-konform – es sind sowohl explizite als auch parametrische Darstellungen möglich), Topologie, Visualisierung, Möglichkeiten zur externen Referenzierung, Materialdefinition, Kosten- und Terminplanung, Ausschreibung, Anerkennungsstatus von Objekten und Projektinformationen, Organisationsstrukturen sowie

dynamische Modellerweiterungen. Als dynamische Modellerweiterung wird eine Vereinbarung „außerhalb“ des Datenmodells bezeichnet, die also unabhängig von diesem getroffen und auch verändert werden kann. In den oberen Schichten des Modells können folgende Bauelemente dargestellt werden: Balken, Stützen, Dächer, Wände, Deckenplatten, Treppen, Treppenabsätze und -aufgänge, Türen und Fenster. Ihr niedriger Detaillierungsgrad genügt jedoch lediglich in geometrischer Hinsicht ingenieurtechnischen Ansprüchen. Trotzdem ist ihr Vorhandensein im Hinblick auf die Harmonisierung mit dem Komplettbaumodell wichtig.

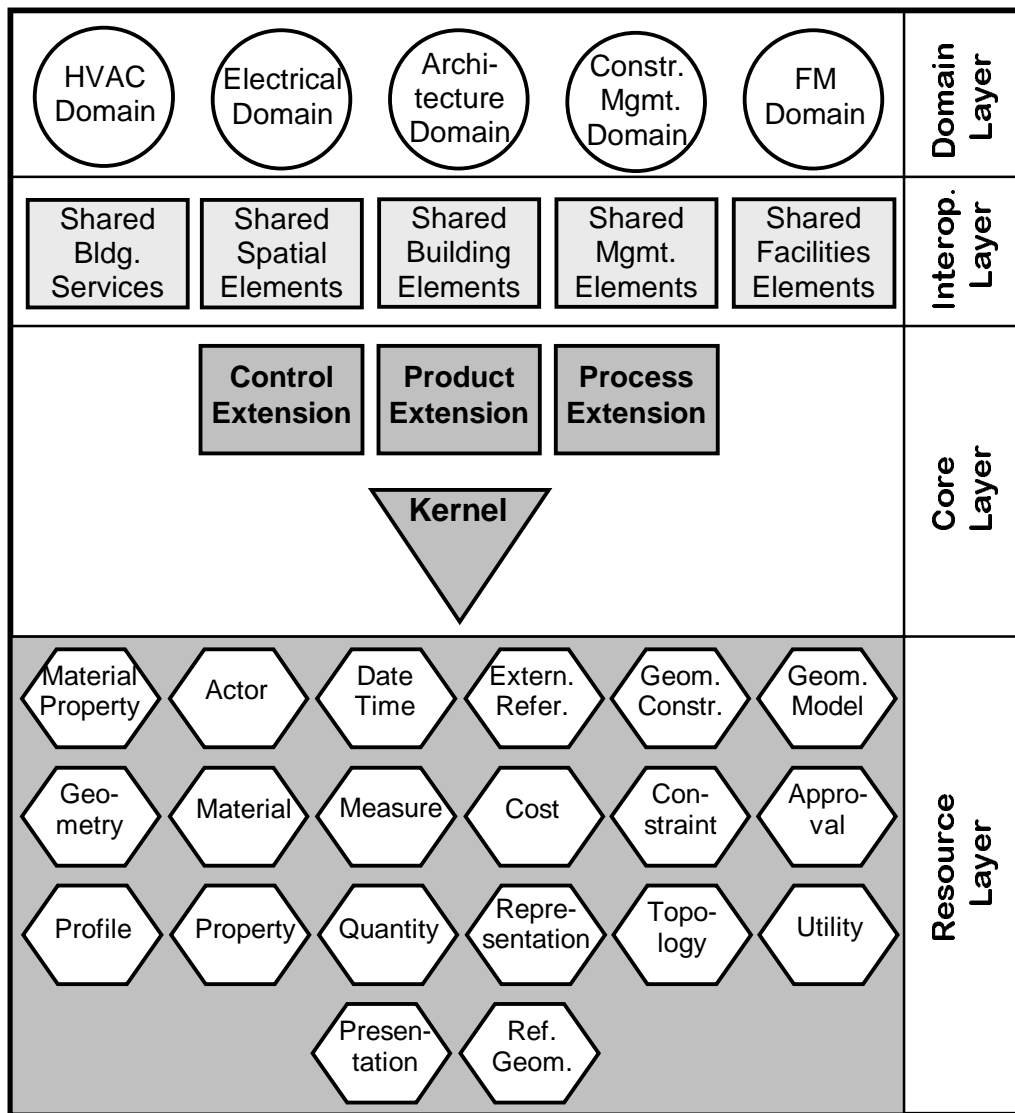


Abbildung 3.1: Modellarchitektur der IFC2x-Plattform

Erwähnt werden sollen auch die XML-Aktivitäten der IAI: Zu dem in EXPRESS definierten Datenmodell existiert ebenfalls eine XML-Variante, das sogenannten *ifcXML* [65]. Aufgrund einer formalen Überführungsvorschrift ist außerdem die maschinelle Übersetzung von zukünftigen, EXPRESS-basierten Erweiterungsmodellen der IFC in *ifcXML* möglich [64].

Die IAI verleiht Zertifikate für Implementierungen der IFC in kommerzielle Software und führt selbst die dazu notwendigen Konformitätsprüfungen durch. Das Verfahren verfolgt einen Kompromiß aus minimalen Kosten und ausreichender Gründlichkeit: Die Prüfung findet für IAI-Mitglieder öffentlich auf sogenannten Zertifizie-

rungs-Workshops statt. Getestet wird nicht die Qualität einer Implementierung, sondern nur deren Fähigkeit, eine hochwertige Austauschdatei zu lesen oder zu schreiben. Es existiert eine Bibliothek mit Testdateien, die bei Bedarf erweitert wird. Abstrakte Testfälle gemäß ISO 10303 gibt es jedoch nicht.

Wie bereits erwähnt ist das Datenmodell der IFC aufgrund seines Anspruches auf Allgemeingültigkeit äußerst umfangreich und komplex. Bezüglich der Implementierbarkeit stehen dieser Komplexität jedoch einige EDV-technische Vorzüge entgegen: Erstens wird die Wiederverwendbarkeit von Software-Komponenten ermöglicht durch die ausgiebige Anwendung des Vererbungsprinzips, durch die generischen Datenstrukturen des Metamodells, die sich in allen anderen Bereichen der IFC wiederfinden, und durch die konsequente objektorientierte Modellierung. Zweitens bietet der sogenannte *Platform Approach* eine gewisse Kontinuität für Implementierer. Dies bedeutet, daß der Modellkern mittelfristig nicht verändert wird. Außerdem liegt die Dokumentation der Komplexität angemessen in Form einer vernetzten html-Beschreibung vor. Sie ist in englischer Sprache und kann unter der im Schrifttum angegebenen Internetadresse kostenlos bezogen werden. Es existieren kommerzielle, zertifizierte Implementierungen der Version IFC1.5.1. Wirkliche praktische Anwendungen sind mir jedoch nicht bekannt.

Bewertung:

Die IFC befinden sich, gemessen an den Möglichkeiten und den Ansprüchen der IAI, erst in einem Anfangsstadium. Abgesehen von der sehr leistungsfähigen geometrischen Darstellung gilt dies insbesondere für den recht geringen Detaillierungsgrad, der einem praktischen Nutzen zumindest im Ingenieurbereich derzeit noch im Wege steht. Als negativ zu bewerten ist der große Aufwand für Einarbeitung und Implementierung des Modells, dem aber die genannten EDV-technischen Vorzüge und die sehr gute Dokumentation entgegenstehen. Positiv ist die Annäherung an ISO 10303 zu bewerten, die sich in der Verwendung von STEP-Ressourcen für Geometrie- und Materialdarstellung zeigt. Von einer wirklichen STEP-Konformität im Sinne eines APs kann jedoch weder in modelltechnischer noch in formaler Hinsicht gesprochen werden.

Die eindeutig größten Stärken der IFC sind ihre Internationalität, ihr ganzheitlicher Ansatz, die große Flexibilität und modulare Erweiterbarkeit sowie die Tatsache, daß an Erweiterungen auch wirklich gearbeitet wird. Die aktuellen Erweiterungsprojekte sind unter der im Schrifttum angegebenen Internet-Adresse aufgelistet [62]. Zu einer zukünftig breiten Akzeptanz durch die Software-Hersteller wird möglicherweise auch das pragmatische Verfahren zur Zertifizierung und die Öffnung für die XML-Entwicklung beitragen.

Diese Gründe sprechen dafür, daß die IFC zukünftig das wichtigste Produktmodell des Bauwesens sein werden, welches für das weitere Vorgehen in dieser Arbeit nicht außer Acht gelassen werden kann.

3.3.4 CIMSteel Integration Standards (CIS)

Ein Ergebnis des 1988 begonnenen EU-Forschungsprojektes Eureka 130 „CIM-Steel“ war das Produktmodell *Logical Product Model* (LPM), das den Produktdatenaustausch für Stahltragwerke ermöglichen sollte. 1995 erschien die erste Version der *CIMSteel Integration Standards* (CIS), deren Kern das LPM/4.0 war. Die Autoren der CIS hatten immer die STEP-Konformität im Auge, einschließlich der Verwendung der STEP-Ressourcen. Es war von Anfang an das erklärte Ziel, die CIS als *Application Protocol 230* (AP 230) zu etablieren, also als einen Teil von ISO 10303.

1996 wurden die CIS/1.0 als Entwurf beim ISO eingereicht. Das Ziel wurde jedoch nie erreicht; das Projekt AP 230 innerhalb der STEP-Entwicklung ist mittlerweile zurückgezogen worden [ISO10303b]. Das CIMSteel-Projekt endete 1998 und wurde dem Britischen *Steel Construction Institute* (SCI) übergeben, unter dessen Regie im Jahre 2000 die CIS/2 erschienen [12]. Sie bestehen aus einer äußerst umfangreichen Dokumentation in englischer Sprache, die sich sehr stark an den formalen Forderungen aus ISO 10303 orientiert.

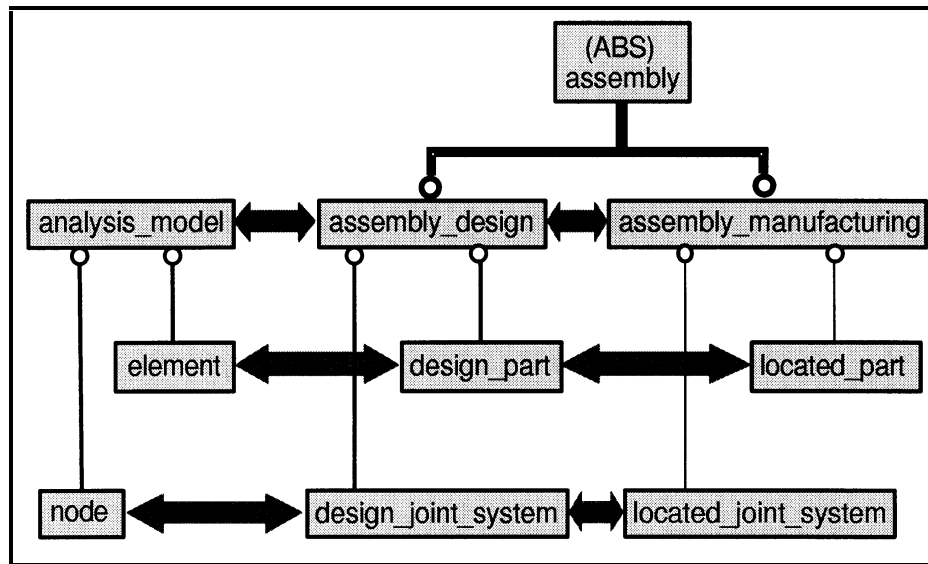


Abbildung 3.2: Übersicht über CIS/2 [12]

Das Datenmodell verfügt in erster Linie über Sprachmittel für den Austausch von Stahlbaudaten auf hohem Detaillierungsniveau. Diese sind Daten für Werkstoffe, parametrische und freie Querschnitte, Raster, „mechanische“ Darstellung von Stabwerken und FE-Modellen, Einwirkungen und Einwirkungskombinationen, Berechnungsergebnisse, Darstellung „physikalischer“ Bauteile, Gruppierung und Bemessung dieser Bauteile, deren Funktion im Gebäude, Verbindungen, Fertigung und Montage. Außerdem können Daten zur Projektplanung und -verwaltung, Dokumentenverwaltung und Qualitätssicherung dargestellt werden. Abbildung 3.2 zeigt – stark vereinfacht – die Einteilung dieser Sprachmittel in drei Sichten (Analysemodell, Entwurfsmodell und Fertigungsmodell).

Nach Angabe der Autoren ist die wichtigste Errungenschaft der CIS/2 die Bereitstellung von Regelungen zur gemeinsamen Datennutzung sowie zur Verwaltung und Versionskontrolle des Datenbestandes in Datenbank-Managementsystemen. Dazu sind fünf aufwärtskompatible Stufen von unterschiedlicher Komplexität und Leistungsfähigkeit vorgesehen, die vom einfachen Dateiaustausch bis zur Implementierung eines sogenannten *Product Model Repositories* reichen.

Derzeit sind neben Absichtserklärungen von Software-Herstellern zur Implementierung bereits Austauschdateien erhältlich, die auf Prototypimplementierungen schließen lassen. Praktische Anwendungen sind mir jedoch weder von CIS/1.0 noch von CIS/2 bekannt. Die CIS/2 enthalten ausführliche Konformitätsanforderungen, abstrakte und konkrete Testfälle sowie Austauschdateien. Eine Konformitätsprüfung wird vom SCI jedoch nicht durchgeführt.

Bewertung:

Die CIMSteel Integration Standards, Release 2, ermöglichen den detaillierten Produktdatenaustausch für Software zur Planung und Bemessung von Stahltragwerken. Das von *Banke* und *Schäufele* in Untersuchungen der Vorgängerversion bemängelte Fehlen wichtiger ingenieurtechnischer Inhalte [5], [101] scheint in den CIS/2 behoben zu sein. Positiv zu bewerten sind auch die Regelungen und Sprachmittel zur Anwendung in Datenbank-Managementsystemen und die zur Verfügung stehenden Mittel zur Konformitätsprüfung. Als sehr hoch muß dagegen der Aufwand für die Einarbeitung in die enorm umfangreiche Dokumentation und für die Implementierung des komplexen Modells eingestuft werden. Dies ist zum Teil auch durch den sehr formalen Charakter der CIS/2 begründet, der obendrein eine flexible Weiterentwicklung des Modells erschwert. Negativ auf die Akzeptanz durch die Software-Hersteller wird sich möglicherweise auch die Vermarktungsstrategie durch das SCI auswirken, das die Dokumentation nicht kostenlos zur Verfügung stellt.

Die Erwartungen, die sich während der Laufzeit an das CIMSteel-Projekt richteten, waren in erster Linie durch die in Aussicht stehende Aufnahme in ISO 10303, die dadurch gewährleistete internationale Akzeptanz sowie die Integration des Stahlbaus in den Gesamtzusammenhang des Bauwesens begründet. Da dieses Ziel nicht erreicht wurde, werden die CIS/2 in dieser Arbeit nicht weiter berücksichtigt.

3.3.5 Produktschnittstelle Stahlbau

Die *Produktschnittstelle Stahlbau* ermöglicht den Produktdatenaustausch in einem Anwendungsgebiet, das dem der CIMSteel Integration Standards sehr ähnlich ist, nämlich der Planung und Fertigung von Stahlkonstruktionen. Dabei wurden jedoch die Besonderheiten des deutschen Stahlbaus berücksichtigt. Das auf [47] basierende Produktmodell [46], [49], [89], [100] wird seit 1996 vom Deutschen Stahlbau-Verband (DStV) zur Anwendung empfohlen. Die aktuelle Version 4/2000 umfaßt Sprachmittel zur Darstellung der Bereiche allgemeine Daten (Querschnitte, Material, Gebäuderaster), Statikdaten, Daten der Rohkonstruktion, detaillierte Konstruktionsdaten, die auch die Anforderungen von NC-Maschinen erfüllen, Oberflächenbeschichtungen sowie beliebige Verbindungen von Bauteilen [32]. Standardteile wie Profile, Schrauben und Schraubengarnituren können allein durch genormte Bezeichnungen nach [29] beschrieben werden. Der sehr hohe Detaillierungsgrad ist Voraussetzung für die Anwendung des Modells in der Praxis.

Abbildung 3.3 zeigt die Einteilung des Modells in sogenannte *Implementierungsbereiche*. Sie ermöglichen es den Software-Herstellern, auch Teile des Modells zu implementieren. Die Implementierungsbereiche sind in Umfangsrichtung des Kreises dargestellt. Die Darstellung läßt außerdem eine grobe Teilung des Modells in das Konstruktionsmodell auf der linken Seite und dessen statischer Abstraktion auf der rechten Seite erkennen.

Verglichen mit den zuvor beschriebenen Produktmodellen ist die Art der Modellierung eher bodenständig. Beispielsweise werden keine Vererbungsstrukturen angewandt, und grundlegende Sachverhalte wie Geometrien werden oft nur implizit definiert oder feature-basiert dargestellt, wie von *Saal* und *Haller* gezeigt [100]. STEP-Ressourcen werden dabei nicht verwendet. Dies alles hat jedoch den Vorteil, daß das Modell kompakt und die Dokumentation sehr übersichtlich ist. Die Dokumentation liegt lediglich in deutscher Sprache vor; sie kann kostenlos von der im Schrifttum angegebenen Internet-Adresse heruntergeladen werden [32].

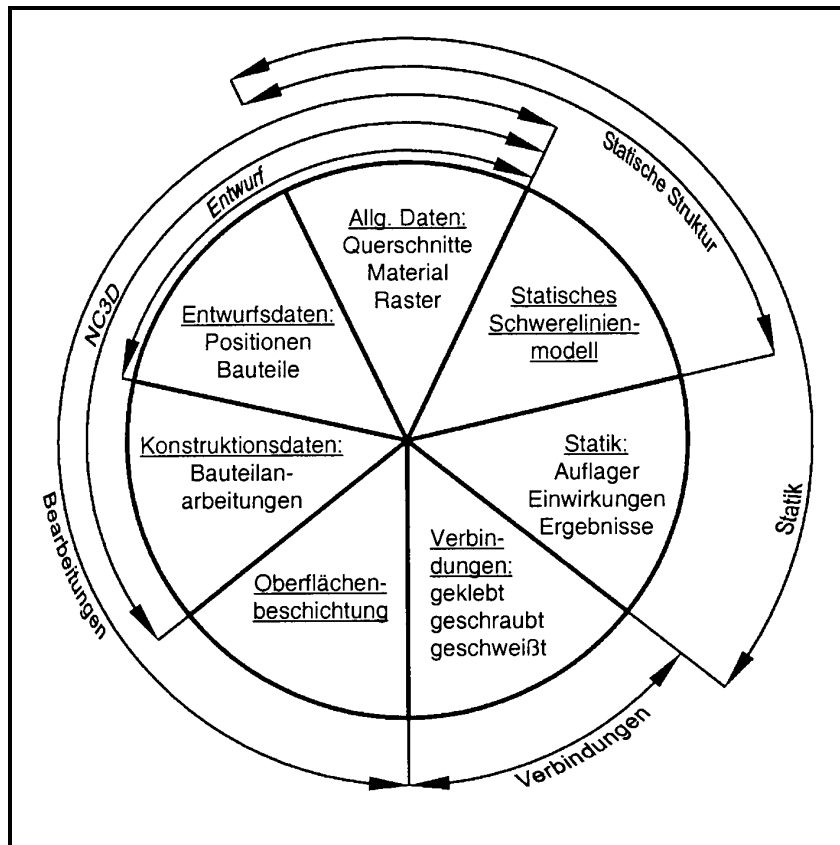


Abbildung 3.3: Modellarchitektur der Produktschnittstelle Stahlbau [32]

Der DStV hat aufgrund des engen Kontaktes sowohl zu Software- als auch zu Bau-firmen im Zusammenhang mit der technischen Weiterentwicklung und der Anwendung der Produktschnittstelle die Notwendigkeit von Konformitätsprüfungen erkannt und ein entsprechendes Verfahren entwickeln lassen [54]. Es orientiert sich in groben Zügen an den Vorgaben von ISO 10303, ist aber weniger formal. Die große Zahl der Testfälle deckt jedoch alle Anwendungsgebiete der Produktschnittstelle ab. Das Testpaket, das aus einer Liste mit abstrakten Testfällen, Kriterien für Bestehen und Nichtbestehen der Tests, Austauschdateien und einer Anleitung zur Durchführung der Prüfung besteht, ist öffentlich zugänglich. Software-Hersteller können damit die Qualität ihrer Schnittstellen verbessern, und Endanwender können die Aussagen ihrer Software-Anbieter stichprobenartig überprüfen. Entgegen seiner ursprünglichen Absicht führt der DStV selbst jedoch weder Prüfungen durch, noch verleiht er Zertifikate für bestandene Tests.

Die Produktschnittstelle Stahlbau ist derzeit das einzige Produktmodell, das in der Baupraxis angewandt wird. Neben Anwendungsbeispielen, bei denen der Software-Anwender den Datenaustausch bewußt vollzieht [55], gibt es verdeckte Anwendungen wie die von *Hahn* beschriebene interne Nutzung der Produktschnittstelle beim Zusammenschluß mehrerer Software-Firmen [46]. Auch wird die Produktschnittstelle in der Forschung angewandt [78], [109], [110].

Bewertung:

Die praktischen Anwendungen der Produktschnittstelle Stahlbau sprechen für sich. Die Gründe dafür sind zuallererst die praxisgerechten Inhalte des Modells und ein Detaillierungsgrad, der ingenieurtechnischen Ansprüchen genügt. Dazu kommt, daß dies in sehr kompakter Form mit übersichtlicher Dokumentation erreicht wird, was sich in vergleichsweise geringem Aufwand für Einarbeitung und Implementierung

auswirkt. Sehr wichtig sind auch die starke Praxisorientierung durch flexible Betreuung und Weiterentwicklung unter Einbeziehung von Bau- und Software-Industrie, das Verfahren zur Konformitätsprüfung sowie die freie Verfügbarkeit der Dokumentation.

Im Hinblick auf eine weitere Verbreitung der Produktschnittstelle ergeben sich jedoch mehrere Probleme: Es hat sich zwar gezeigt, daß durch die Orientierung des Modells an der Denkweise des Ingenieurs das momentane Anwendungsgebiet im Stahlbau sehr gut erfaßt wird, daß aber die Erweiterbarkeit des Modells auf andere Bereiche des Bauwesens durch die eher bodenständige Modellarchitektur zwangsläufig beschränkt ist. Dies gilt besonders für Bereiche außerhalb des konstruktiven Ingenieurbaus. Außerdem steht einer Implementierung durch internationale Software-Hersteller das Fehlen einer englischsprachigen Dokumentation im Wege.

Die Möglichkeit zur Nutzung der Vorzüge der Produktschnittstelle für das in dieser Arbeit zu entwickelnde Produktmodell ist jedoch durch die Entscheidung des DSTV gegeben, die Produktschnittstelle in die IFC zu integrieren. Dadurch wären die oben genannten Probleme gelöst [55]. Die Integration wird derzeit im ST-4 Projekt der IAI durchgeführt. Der Stand des Projektes wird in einer Dokumentation beschrieben, die laufend aktualisiert wird [56].

3.3.6 Produktmodell Datentransfer im Holzbau (DtH)

Die erste Version des damals noch *Produktschnittstelle Holzbau* genannten Produktmodells entstand parallel zur Produktschnittstelle Stahlbau [48]. Daraus wurde an der Universität Cottbus das *Produktmodell Datentransfer im Holzbau* (DtH) entwickelt, dessen aktuelle Version 2/2000 den Ingenieurholzbau, den Zimmermannsbau und den Fertighausbau abdeckt [118]. Es können Elemente des Tragwerkes (CAD- und statisches Modell), raumbildende Elemente (Wände, Dächer, Decken), Öffnungen (Türen, Fenster, Treppenaugen, Schornsteinöffnungen), Fundamente, Schornsteine, Treppen, Bauabschnitte, Etagen sowie Positionen für Ausschreibung, Vergabe und Abrechnung dargestellt werden. Diese Elemente können mit sehr hohem, den Ansprüchen des Ingenieurholzbaus genügendem Detaillierungsgrad dargestellt werden.

Die Art der Modellierung, die Modellarchitektur und die Dokumentation sind denen der Produktschnittstelle Stahlbau sehr ähnlich. Im Gegensatz zu ihr ist die vollständige Dokumentation jedoch auch in englischer Sprache erhältlich. Statt eines mehr oder weniger formalen Verfahrens zur Konformitätsprüfung wird ein Programm angeboten, das die Syntax von Austauschdateien überprüfen und deren Inhalt am Bildschirm darstellen kann. Dokumentation und Programm können kostenlos von der im Schrifttum angegebenen Internet-Adresse heruntergeladen werden [118]. Das DtH ist in mehreren kommerziellen CAD-Systemen für den Holzbau implementiert.

Bewertung:

Da das Produktmodell DtH prinzipiell die gleichen Stärken und Schwächen wie die zuvor beschriebene Produktschnittstelle Stahlbau hat, und da ebenfalls die Integration in die IFC geplant ist, trifft die Bewertung der Produktschnittstelle auch auf das DtH zu.

3.4 Modelle zur Unterstützung des Datenaustausches

3.4.1 Allgemeines

In diesem Abschnitt werden Modelle gezeigt, die oft in Verbindung mit Produktmodellen genannt werden, und die der Unterstützung des Datenaustausches dienen.

3.4.2 Prozeßmodelle

Ein *Prozeßmodell* ist die analytische Beschreibung von Aktivitäten und Informationsflüssen von Planung oder Ausführung. Prozeßmodelle können beispielsweise der Verdeutlichung dieser Informationsflüsse dienen, was in der Produktmodellierung eine wichtige Rolle spielt. In der Bauinformatik wird darüber hinausgehend versucht, das beim Produktdatenaustausch auftretende Problem der Planungskonflikte mit Hilfe von Prozeßmodellen zu lösen. Planungskonflikte treten insbesondere dann auf, wenn mehrere Planer nicht in zeitlich geregelter Abfolge fest definierte Planungsschritte ausführen, sondern iterativ zu Lösungen kommen, so wie es im Bauwesen in der Regel der Fall ist. Diese Konflikte können durch das Produktmodell aufgrund seines rein beschreibenden Charakters nicht gelöst werden. *Scherer und Katranuschkov* stellen dem Produktmodell deshalb ein zusätzliches Prozeßmodell zur Steuerung des Produktdatenaustausches zur Seite [102].

Zwar existieren formale Spezifikationssprachen zur Erstellung von Prozeßmodellen, zum Beispiel *IDEFO* [88], die Prozeßmodelle selbst sind in der Regel jedoch keine allgemeingültigen Standards im Sinne von Produktmodellen. Sie dienen vielmehr der Beschreibung projekt- und firmenspezifischer Abläufe und werden in der Regel in Form von Software zur aktiven Steuerung des Informationsflusses eingesetzt. So liegt im Prinzip jedem PMS oder IMS ein Prozeßmodell zugrunde, das individuell auf spezielle Anwendungen zugeschnitten werden kann.

3.4.3 Zentrale Projektdatenbanken

Die Idee, ein Bauvorhaben über eine zentrale, von Anwendungs-Software unabhängige Projektdatenbank abzuwickeln, ist nicht neu. Bereits *Mitchell* propagierte das zentrale, vollständig integrierte Arbeiten, allerdings für die festgelegten Abläufe bei der Planung von standardisierten Bauobjekten, wie den *Harness-Hospitals* in den USA oder der *Oxford Methods of Building* in England [85]. Analog zum Einsatz objektorientierter Produktmodelle befassen sich neuere Forschungen mit der Entwicklung oder Anwendung objektorientierter Datenbanksysteme [84], [120].

In letzter Konsequenz werden umfassender, verlustfreier Produktdatenaustausch und -haltung in der Tat nur mit Datenbankunterstützung möglich sein. Deshalb wurden Datenbanksysteme auch von Anfang an als eine mögliche Anwendung von Produktmodellen gesehen, etwa in der STEP-Bewegung [73] oder bei der von *Haller* entwickelten Projektdatenbank Stahlbau [47]. Die Datenstruktur des Produktmodells wird dabei vollständig in der Datenbank umgesetzt. Für jedes Projekt existiert ein einziger, zusammenhängender Datenbestand. Die in Forschungsprojekten entwickelten Produktmodell-datenbanken zur Systemintegration im Bauwesen haben jedoch nie einen Umfang erreicht, der an die Ansprüche der Baupraxis herangereicht hätte.

3.4.4 Verteilte Systeme

Mittlerweile wird zunehmend von der Vorstellung abgewichen, ein Projekt in seiner Gänze zentral mit einem einzigen Datenbestand abwickeln zu können. Die ursprüngliche, praxisfremde Meinung war die, daß bei einer integrierten Projektabwicklung wirklich alle anfallenden Daten in einer zentralen Datenbank gespeichert werden müssen, damit das System überhaupt funktioniert. Eine neuere Tendenz für datenbankgestützte Projektbearbeitungen geht dagegen in die Richtung, die Datenbank auf mehrere Server zu verteilen. *Schmitt* berichtet von einer Pilotanwendung mit einer auf den Bauherrn und die Hauptplanungspartner verteilten Projektdatenbank, wodurch nach Angabe des Autors Kosten und Zeit gespart werden [105].

Die Akzeptanz der wesentlich flexibleren verteilten Systeme wird möglicherweise größer als die der zentralen Projektdatenbanken sein. Dies ist auch vor dem Hintergrund zu sehen, daß oftmals die kleineren Projektpartner experimentierfreudiger sind als die großen Firmen. Diese kleinen Büros oder Firmen sind jedoch in der Regel nicht in der Position, eine einheitliche Lösung für ein gesamtes Projekt durchzusetzen, sondern eher für Teilprojekte.

Das Thema der verteilten Systeme ist neu im Bauwesen; entsprechend groß ist noch der Forschungsbedarf. Ergebnisse sind beispielsweise vom laufenden DFG-Schwerpunktprogramm „Vernetzt-kooperative Planungsprozesse im konstruktiven Ingenieurbau“ zu erwarten [15]. In diesem Projekt sollen Planungsprozesse und Kooperationsmodelle analysiert sowie Verfahren und Methoden zum Arbeiten in verteilten Systemen entworfen werden. Das Programm endet im Jahre 2006.

3.4.5 Kommunikation über das Internet

Kommunikation über das Internet bedeutet natürlich auch den Austausch von Daten. Nachdem sich in den letzten Jahren die dazu notwendigen Basisstandards etablieren konnten, wie zum Beispiel XML, beginnt man über den Produktdatenaustausch speziell im Internet nachzudenken. Für den internetbasierten Handel im Bauwesen werden zur Zeit beispielsweise die Standards *aecXML* [63] und *bcXML* [115] entwickelt. Nach *Liebich* haben sich beide Initiativen dazu bekannt, die Inhalte von *ifcXML* [65] zu übernehmen [82].

3.5 Anwendung der Modelle in der Baupraxis

Angesichts der Anstrengungen, die zur Erschaffung von Produktmodellstandards und für Forschung zur Systemintegration unternommen wurden, und angesichts der Tatsache, daß deren möglicher Nutzen seit langer Zeit bekannt ist, ist der in der Baupraxis tatsächlich durchgeführte Datenaustausch verschwindend gering – von einer wirklichen Integration der Planungsschritte ganz zu schweigen.

Selbst der Austausch einfacher, ebener Geometriedaten ist oftmals nicht realisierbar. Ein Grund dafür ist die von *Haas* bereits 1993 kritisierte Abhängigkeit der Anwender vom herstellerdefinierten Austauschformat *dxf* (*Drawing Exchange Format*) des Software-Herstellers AutoDesk, mit dem lediglich geometrische Daten übertragen werden können [45]. Die weiter oben bereits zitierte Umfrage des DStV zeigt, daß *dxf* auch heute noch die meistangewandte Schnittstelle im Stahlbau ist [33]. Ebenfalls als „Schnittstelle“ wird oft das Dateiformat des gemeinsamen Basissystems von aufgesetzten CAD-Programmen bezeichnet. Auch dies ist keine befriedigende Lösung, da hierbei ebenfalls nur reine Geometriedaten übertragen werden können, und man ebenfalls vom Hersteller der zugrundeliegenden Software abhängig ist.

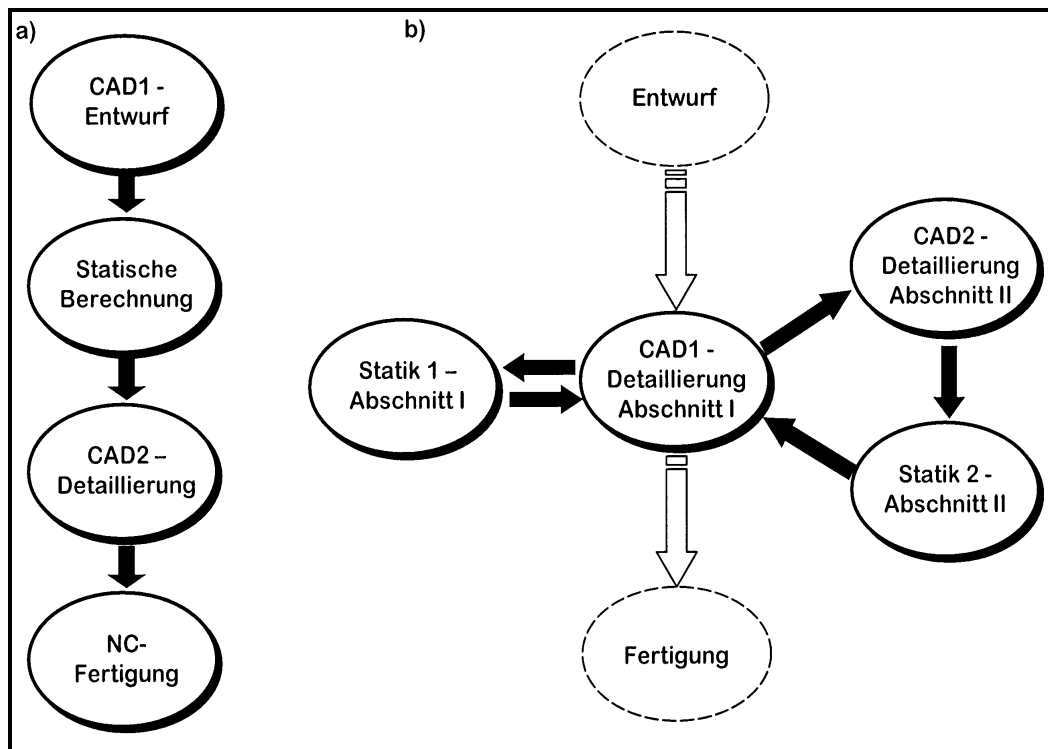


Abbildung 3.4: a) „vertikale“ und b) „horizontale“ Systemkopplung

Produktdaten können dagegen mit den bewährten Schnittstellen des DStV zur Übergabe von CAD-Daten an die NC-Steuerung [30] und Stückliste [31] übertragen werden, was auch praktiziert wird. Diese Schnittstellen sind jedoch im Gegensatz zu den in Abschnitt 3.3 beschriebenen Produktmodellen jeweils auf ganz bestimmte Anwendungen beschränkt. Produktdaten können auch mit dem Standard *GAEB 2000* übertragen werden [43]. Er dient zur Vereinheitlichung des Aufbaus von Leistungsverzeichnissen und als Schnittstelle zur Übergabe von Daten aus dem Objektmodell von CAD-Programmen an AVA-Programme. *GAEB 2000* ist bereits in handelsüblicher Bau-Software implementiert. Produktdatenaustausch im Sinne von verlustfreier, ein größeres Anwendungsgebiet betreffender Kommunikation ist jedoch mit keiner der genannten Schnittstellen möglich.

Das einzige der in diesem Kapitel vorgestellten Produktmodelle, das tatsächlich in der Praxis angewandt wird, ist die Produktschnittstelle Stahlbau [55]. Die Anwendungen beschränken sich jedoch derzeit auf Schritte, die im Planungsprozeß nacheinander ausgeführt werden, wie zum Beispiel die Übernahme einer CAD-Rohkonstruktion in ein Stabwerksprogramm oder umgekehrt. Dies wird als „vertikale“ Systemkopplung bezeichnet, siehe Abbildung 3.4.

Der Datenaustausch in den zitierten Beispielen fand ausschließlich über Austauschdateien statt. Zentrale Datenbanken zur gemeinsamen Datenhaltung wurden nur intern bei Zusammenschlüssen mehrerer Software-Hersteller zwischen ihren eigenen Produkten eingesetzt [46]. Die praktische Anwendung von Projektdatenbanken, die wirklich unabhängig von Anwendungs-Software sind, ist mir dagegen nicht bekannt.

Was sind also die Gründe dafür, daß die in den vorangegangenen Abschnitten genannten Möglichkeiten nur so zögerlich angenommen werden? Erklärungen, die die Schuld jeweils nur einer Seite zuschieben (der Bauindustrie oder den Software-Herstellern), sind zu einfach: *Rezman* argumentiert beispielsweise, daß für den einzelnen Bauunternehmer als Teil einer Ad-hoc-Organisation gar kein Anreiz zur Schaffung eines langfristigen Kommunikationssystems bestünde, da die Kommuni-

kationsstrukturen im Bauwesen eben ad hoc geschaffen werden [95]. Dem muß jedoch entgegengehalten werden, daß ein Bauunternehmen sehr wohl Interesse haben wird, sowohl über interne Kommunikationssysteme als auch über solche zu seinen Subunternehmern zu verfügen. Richtig ist dagegen, daß die mittelständischen deutschen Bauunternehmer gar nicht über die Mittel verfügen, Software und Schnittstellen auf dem heutigen Stand der Technik zu entwickeln.

Umgekehrt lautet der Vorwurf, daß viele Software-Hersteller gar nicht unbedingt daran interessiert sind, Daten mit Produkten konkurrierender Anbieter austauschen zu können, oder gar ihre Systeme zu öffnen. Die „horizontale“ Kopplung gleichartiger Programme betreffend, siehe Abbildung 3.4b, mag dies vielleicht berechtigt sein. Dieses Argument wird jedoch in Zukunft in den Hintergrund treten, wenn zunehmend die Integration mit immer mehr Programmen aus den verschiedenen Bereichen des Komplettbaus oder des gesamten Bauwesens gefragt ist, die nicht in direkter Konkurrenz zueinander stehen.

Die wohl zutreffendste Erklärung des Zögerns ist wohl die, daß derzeit noch keines der Produktmodelle alle Anforderungen in einem umfassenden Anwendungsgebiet erfüllen kann, und daß demzufolge keine Einigkeit darüber besteht, welches der Modelle sich durchsetzen wird. Sobald diese Einigkeit jedoch herrscht, werden die Anwender die Implementierung dieses Modells von ihren Software-Herstellern fordern können, und diese werden dann auch dazu bereit sein.

3.6 Fazit

Die in diesem Kapitel angestellten Betrachtungen zeigen, daß die Kopplung der Software-Systeme aus allen Bereichen der Architektur und des Bauingenieurwesens ein immer wichtigeres Thema werden wird. Die Voraussetzungen dafür sind einerseits durch den in den letzten Jahren stark angewachsenen Leistungsumfang der Anwendungsprogramme gegeben, so daß sich die Anwendungsgebiete der unterschiedlichen Systeme mittlerweile fast lückenlos ergänzen, und andererseits durch die modernen Informations- und Kommunikationstechnologien. Dennoch muß festgestellt werden, daß die Kopplung trotz der großen Anstrengungen der Vergangenheit noch nicht vollzogen wurde. Der Grund dafür ist nach wie vor das Fehlen eines wirklich universellen, sowohl von Anwendern als auch von Software-Herstellern anerkannten Produktmodellstandards. Auch die sehr vielversprechenden IMS und die weiter oben skizzierten Modelle zur Unterstützung des Datenaustausches vermögen eine wirkliche Integration ohne diesen Produktmodellstandard nicht zu leisten, da ihre Aufgabe ausschließlich die Verteilung und Verwaltung der Informationen ist.

Für einzelne Anwendungsgebiete existieren zwar gute Ansätze, aber keines der untersuchten Produktmodelle wird den Anforderungen des gesamten Bauwesens oder auch nur größerer Sparten gerecht. Im Gegenteil, die bloße Existenz verschiedener Standards steht einer Verbesserung der Situation sogar im Wege. Deshalb soll im weiteren Verlauf dieser Arbeit kein neues Produktmodell entwickelt werden, sondern es soll ein Erweiterungsmodell zu den IFC2x definiert werden, die von den betrachteten Produktmodellen eindeutig das zukunftssträchigste sind. Für die IFC sprechen neben den Vorzügen des Modells selbst der sich zusätzlich erschließende Anwendungsbereich, die Praxiserfahrungen und die Unterstützung seitens der Bau- und Software-Industrie, die durch die Integration der Produktschnittstelle Stahlbau und des DtH demnächst zur Verfügung stehen werden.

4 Ingenieurgemäße Anforderungen

4.1 Allgemeines

In diesem Kapitel sollen alle aus der Sicht des Ingenieurs notwendigen Vorüberlegungen zur Spezifikation des Datenmodells getroffen werden. Um die beabsichtigten Informationsflüsse im Planungsablauf des Komplettbaus zu beschreiben, werden zunächst die Planungsschritte in Form sogenannter *Anwendungsprozesse* dargestellt. In erster Linie bestimmen sie die Datenarten und -mengen, die das Datenmodell abdecken muß. Die Darstellung der Anwendungsprozesse ist im Vergleich zur herkömmlichen, zeichnungsorientierten Arbeitsweise im Bauwesen modellorientiert, so daß hiermit bereits die Grundzüge der Modellstruktur festgelegt werden. Um die genauen Inhalte der Modellierung einschließlich der zu erzielenden Detaillierungen festzulegen, werden die einzelnen Gewerke anschließend aus technischer Sicht beschrieben.

Da das Produktmodell als Erweiterung der IFC entwickelt werden soll, wird im folgenden der Begriff des *Bauelementes* im Sinne der IFC verwendet: Ein Bauelement ist ein Teil eines Bauwerks mit globaler Funktion, zum Beispiel ein Dach als Ganzes. Teile, die lediglich für den Fachplaner von Bedeutung sind, werden als *Bauteile* bezeichnet. Verlegetafeln innerhalb eines Daches sind beispielsweise Bauteile.

4.2 Anwendungsprozesse

4.2.1 Allgemeines

Die Anwendungsprozesse dieses Abschnittes zeigen die Informationsflüsse zwischen den Planungsschritten im Komplettbau, die über das Produktmodell abgewickelt werden sollen, sowie die dabei transportierten Dateninhalte. Sie zeigen nicht, wie etwa die Vielzahl von Pfeilen in den Abbildungen suggerieren könnte, eine Reihe von Einzelschnittstellen.

Die Prozesse stellen den realistischen Fall dar, daß in allen Planungsschritten zumindest ein leistungsfähiges CAD-System eingesetzt wird, das mit einem Objektmodell arbeitet. Teilprozesse können jedoch auch ohne Computerunterstützung durchgeführt werden; die Ergebnisse müssen dann von Hand in die Software des nächsten Planers eingegeben werden.

Eine allgemeingültige und zugleich detaillierte Beschreibung aller Informationsflüsse im Bauwesen ist weder effektiv noch sinnvoll. Aus diesem Grunde werden hier zum Teil starke Vereinfachungen getroffen, und es werden nur Standardfälle betrachtet.

Zunächst wird eine gewerkeübergreifende Betrachtung angestellt, die die Abläufe mit den klassischen Begriffen des Bauwesens beschreibt. Diese Betrachtung übernimmt also in sehr komprimierter Form die Funktion des AAM nach ISO 10303. Die Planung für jedes Gewerk wird dabei als jeweils ein Prozeß dargestellt. Aus Gründen der Übersichtlichkeit sind lediglich die Planungsprozesse abgebildet, die sich innerhalb des Anwendungsgebietes des Produktmodells befinden. Dies sind die Architekturplanung und die Fachplanungen der Gewerke *Glasbau*, *Stahlbetonbau*, *Stahltragwerk*, Dach- und Außenwandbekleidung (im folgenden einfach *Dach und Wand* genannt) sowie der Ausbau von *Dach- und Wandöffnungen* (in den Abbildun-

gen verkürzt Öffnungen genannt). Diese Öffnungen werden als ein separater Prozeß betrachtet, so daß sie von allen anderen Gewerken benutzt werden können.

Diese Betrachtung wird anschließend durch gewerkespezifische Prozesse verfeinert, die sich bereits der Begriffe des Datenmodells bedienen, wie die oben erwähnten Bauelemente und Bauteile. Dies erfordert zunächst ein Umdenken auf seiten des Lesers, ist jedoch für die spätere Modellierung äußerst hilfreich: Diese Darstellung legt nämlich unmittelbar das Konzept für das spätere Datenmodell im Sinne des ARM nach ISO 10303 fest. Neben den Prozessen für die eigentlichen Gewerke des Produktmodells, die in den Abbildungen grau hinterlegt sind, werden auch periphere, für Planung und Ausführung jedoch wichtige Prozesse berücksichtigt (statische und bauphysikalische Bemessung, Materialbestellung, Fertigung und Montage). Für das Stahltragwerk braucht kein gewerkespezifischer Anwendungsprozeß betrachtet zu werden, da ein entsprechendes Erweiterungsmodell der IFC ja im bereits erwähnten ST-4 Projekt der IAI entwickelt wird [56].

In den Abbildungen 4.1 bis 4.6 wird zwischen primären und sekundären Datenwegen unterschieden. Primär werden diejenigen Datenwege genannt, die für den (unrealistischen) Fall notwendig sind, daß die Planer sich nicht gegenseitig beeinflussen, daß also die einzelnen Schritte lediglich vertikal gekoppelt sind. Für eine darüber hinausgehende, horizontale Kopplung werden auch sekundäre Datenwege benötigt, die mit gestrichelten Pfeilen dargestellt sind.

Die Untersuchungen dieses Abschnittes basieren auf der im vorangegangenen Kapitel bereits erwähnten Recherche, die im Rahmen des AiF-Forschungsprojektes „Weiterentwicklung der Produktschnittstelle Stahlbau für den Komplettbau“ durchgeführt wurde [57].

4.2.2 Gewerkeübergreifende Betrachtung

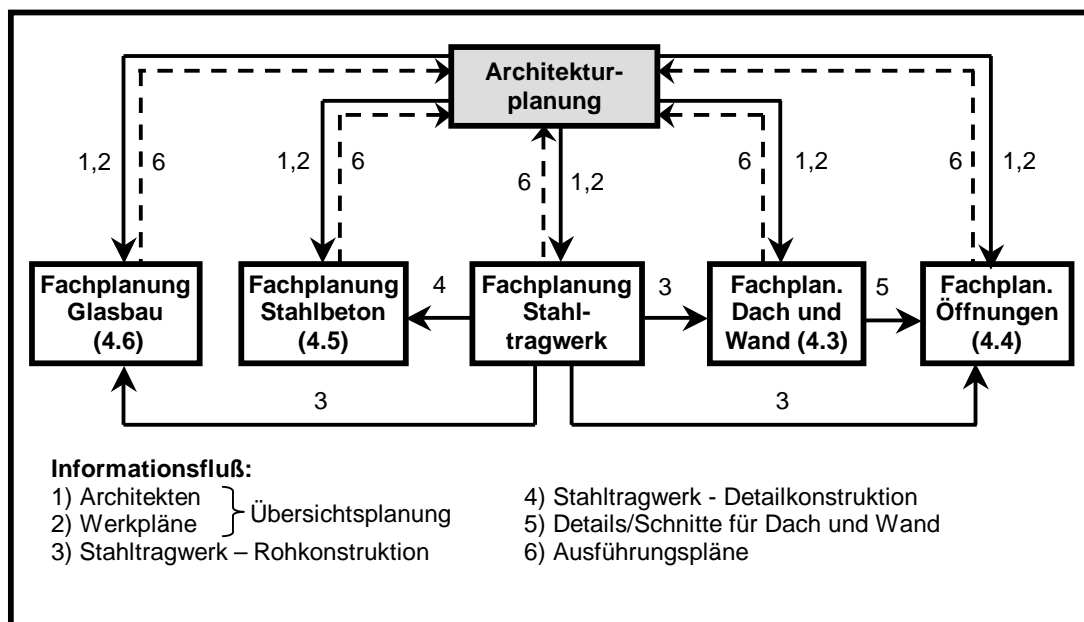


Abbildung 4.1: Informationsfluß bei koordinierender Funktion des Architekten

Abbildung 4.1 zeigt in stark vereinfachter Form die klassische Variante des globalen Planungsablaufs, bei der der Architekt den Entwurf des Bauwerks erstellt und die einzelnen Gewerke an die Fachplaner vergibt. Die Integration der Fachplanungen verbleibt beim Architekten, der Rückmeldungen oder Änderungen erhält. Die Koor-

dination kann im zentralen Objektmodell des CAAD-Systems des Architekten stattfinden.

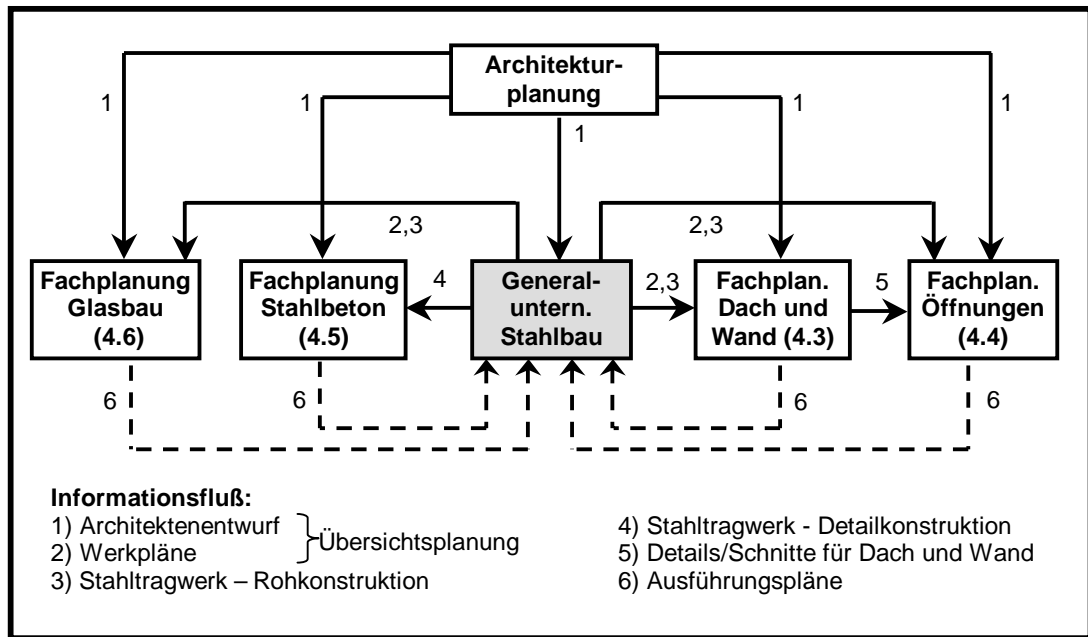


Abbildung 4.2: Informationsfluß bei koordinierender Funktion des Stahlbauers

Abbildung 4.2 zeigt die im Industriebau übliche Variante, daß nämlich das Stahlbauunternehmen die zentrale, koordinierende Rolle übernimmt. Es vergibt die einzelnen Gewerke an Nachunternehmer oder Partnerunternehmen und erhält die Ergebnisse der Fachplaner zur Kontrolle oder zur nachträglichen Anpassung der Stahlkonstruktion zurück. Das zentrale Objektmodell wird hier im Stahlbau-CAD-System erzeugt. Im allgemeinen erhalten die Fachplaner jedoch zusätzlich die Entwürfe des Architekten.

In beiden Abbildungen ist die Weitergabe der Daten aus der Ausführungsplanung an Fertigung und Montage aus Gründen der Übersichtlichkeit nicht eingezeichnet. Diese Informationsflüsse werden jedoch in den folgenden Abschnitten berücksichtigt.

4.2.3 Dach und Wand

Abbildung 4.3 zeigt die typische Planung eines Gewerkes für Dach oder Wand. Der dargestellte Prozeß ist aus den gewerkeübergreifenden Betrachtungen der Abbildungen 4.1 und 4.2 herausgelöst und wurde weiter verfeinert. Diejenigen Datenwege, die aus den übergeordneten Diagrammen übernommen wurden, sind mit den entsprechenden Zahlen gekennzeichnet.

In der Übersichtsplanung werden Raster und Bauelemente mit CAD- oder CAAD-Programmen erzeugt. Dies beinhaltet die groben Abmessungen der Bauelemente samt Öffnungen, die Vorgabe der Bauart sowie besondere Anforderungen, zum Beispiel an Wärme- und Brandschutz.

In der Phase der Detailplanung legt der Fachplaner im CAD-System daraufhin die genaue Bauart der Bauelemente einschließlich aller verwendeter Profile und Bauprodukte für Dämmung und Dichtung fest. In den meisten Fällen können die Produkte mit den Tragfähigkeitsangaben des Herstellers ausgewählt werden. In komplizierteren Fällen muß statt dessen eine statische Berechnung durchgeführt werden. Dasselbe gilt für die bauphysikalische Bemessung. Oft werden in diesem Stadium auch die Anschlußdetails der Gebäudehülle an Teile der Öffnungsgewerke oder an

das Stahltragwerk geplant. Das Stahltragwerk muß deshalb zumindest als Rohkonstruktion bekannt sein.

Auf Basis dieser Vorgaben werden in der Verlegeplanung mit speziellen CAD-Systemen die Form und die Anordnung der einzelnen Bauteile, deren Befestigungen, gegebenenfalls besondere Montagevorschriften sowie die Palettierung festgelegt. Hierbei müssen natürlich die Bauteile der Unterkonstruktion und die Öffnungswerke berücksichtigt werden.

Die mittlerweile in hohem Detaillierungsgrad vorliegenden Modelldaten werden schließlich zur Koordination an den Generalplaner zurückgegeben und als Grundlage für Materialbestellung und Abrechnung in AVA-Programmen genutzt. Außerdem werden sie als Montagezeichnungen für die Baustelle geplottet.

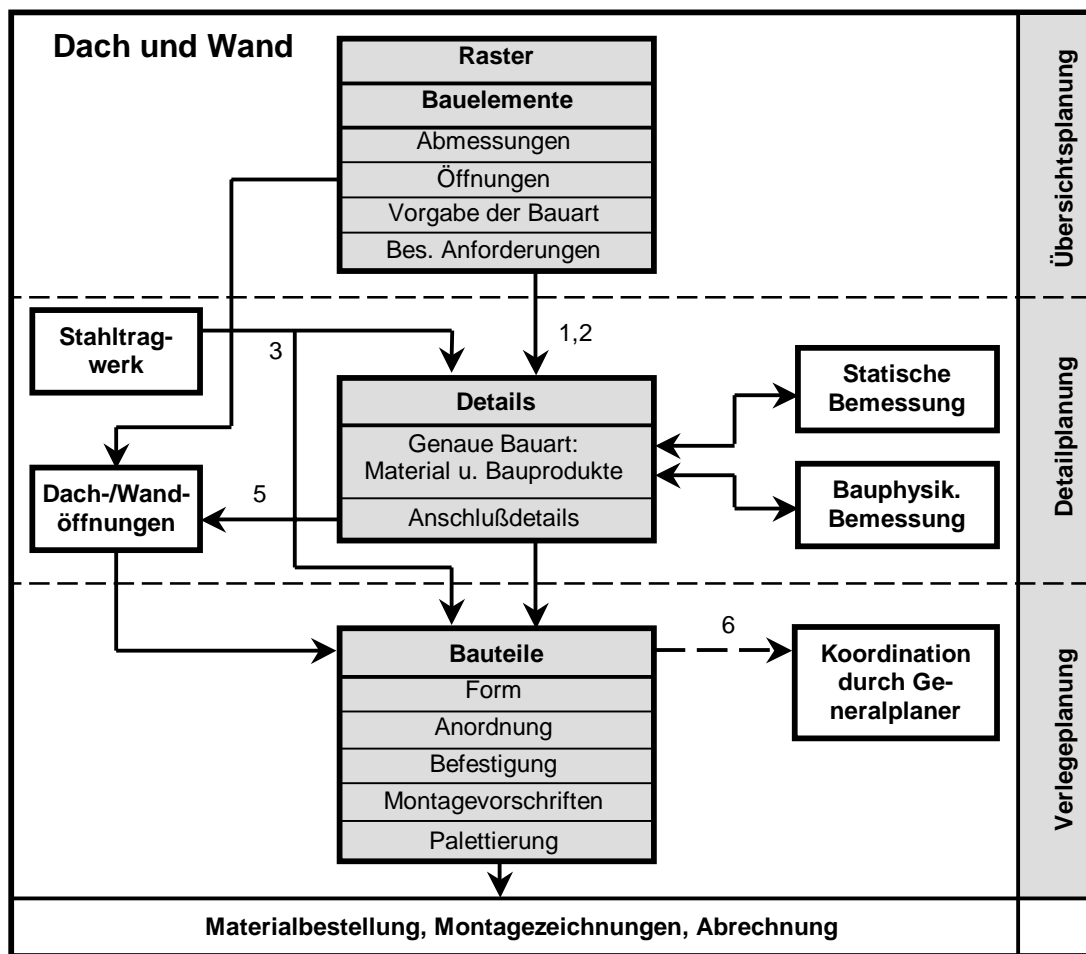


Abbildung 4.3: Planung von Dach und Wand

4.2.4 Dach- und Wandöffnungen

Die Gewerke für die Öffnungen von Dach und Wand müssen an die übrigen Gewerke angepaßt werden und können deshalb in der Regel zeitlich erst nach diesen geplant werden. Entsprechend einfach ist der Informationsfluß, siehe Abbildung 4.4. Üblicherweise erhält zum Beispiel ein Fensterbauer neben den Daten der Übersichtsplanung (grobe Abmessung und gewünschte Bauart der Fenster sowie besondere Anforderungen) Daten aus der Detailplanung von Dach und Wand (genaue Bauart der Gebäudehülle sowie gegebenenfalls die Vorgabe von Anschlußdetails).

Das Stahltragwerk kann ebenfalls von Bedeutung sein, wenn es sich um Auswechslungen für größere Öffnungen handelt, zum Beispiel bei Lichtbändern. Nach diesen Vorgaben legt der Fachplaner im CAD-Programm die Bauteile des Öffnungsgewerkes fest. Der in Abbildung 4.4 gestrichelt eingezeichnete Datenweg in Richtung der Verlegeplanung der Gebäudehülle erübrigt sich meistens, da die genaue Anpassung der Öffnungen oft direkt auf der Baustelle vorgenommen wird. Die detaillierten Daten der Konstruktion gehen anschließend zur Koordination an den Generalplaner zurück sowie an die Materialbestellung, Fertigung, Montage und Abrechnung.

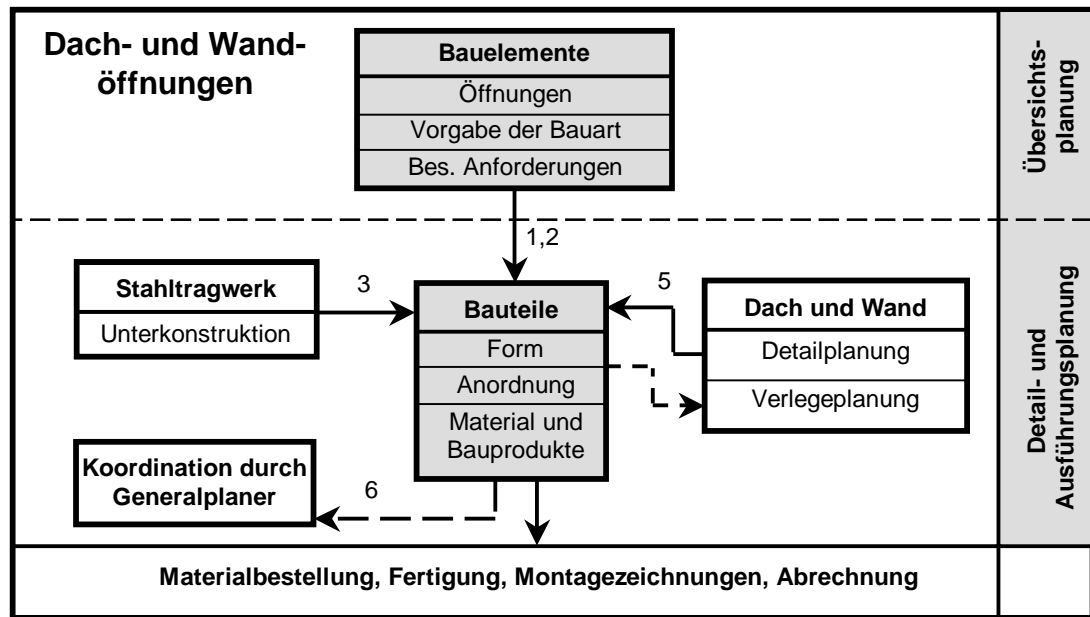


Abbildung 4.4: Planung von Dach- und Wandöffnungen

4.2.5 Stahlbetonbau

Die Planungsabläufe des Stahlbetonbaus sind nicht weniger komplex als die der übrigen Bauweisen. Innerhalb der IAI wird der Stahlbetonbau bereits in einem speziellen Erweiterungsprojekt für die IFC behandelt [112], von dem allerdings noch keine Ergebnisse vorliegen. Aus diesem Grunde wird für das Produktmodell im Rahmen dieser Arbeit eine möglichst einfache Darstellung angestrebt, die lediglich die dringend benötigten Bauelemente wie Wände, Decken und Fundamente abdeckt. Das sehr komplizierte Thema der Bewehrungsplanung wird gar nicht behandelt. Entsprechenden CAD-Programmen zur Bewehrungsplanung stehen also lediglich die Formen der Bauelemente zur Verfügung. Abbildung 4.5 zeigt den somit sehr einfachen Informationsfluß für ein tragendes Stahlbetonteil.

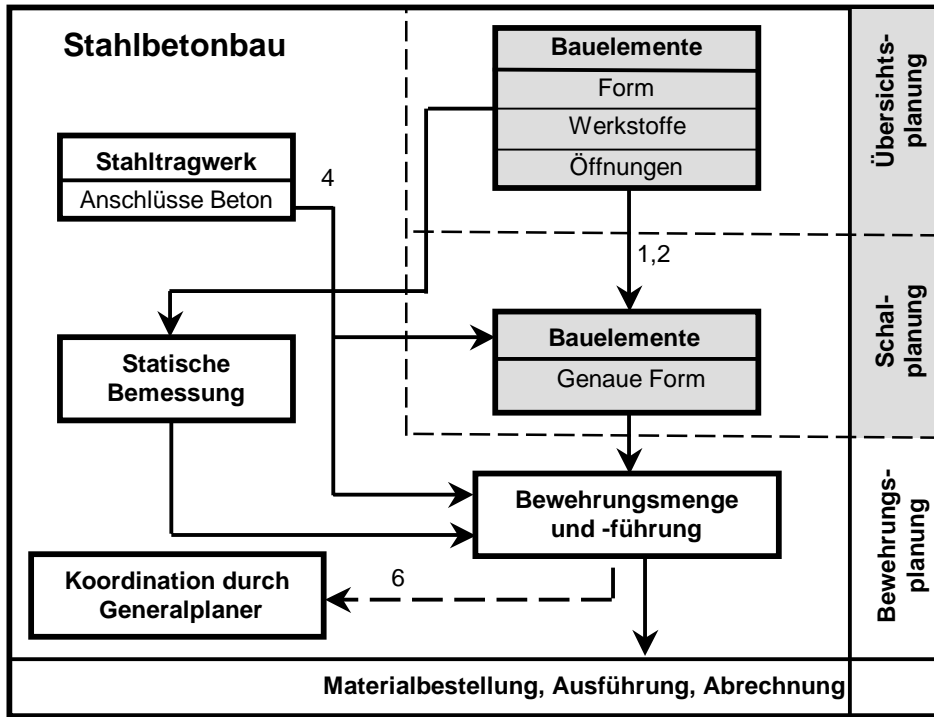


Abbildung 4.5: Planung des Stahlbetonbaus

4.2.6 Glasbau

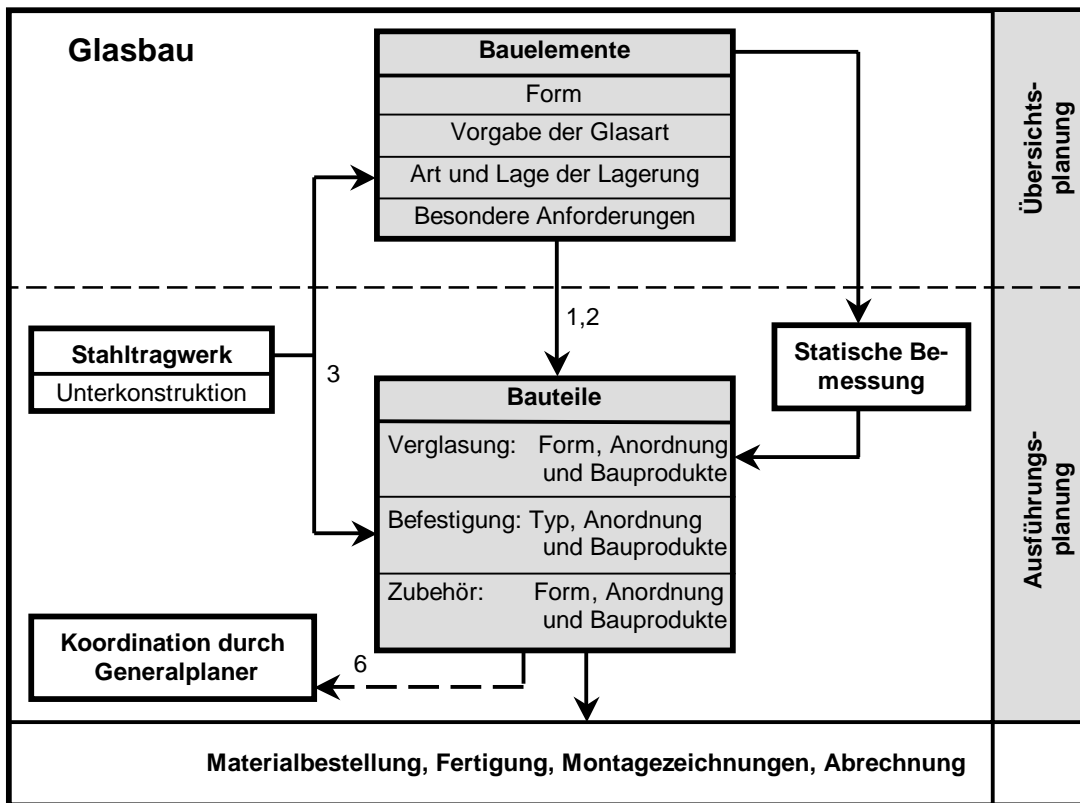


Abbildung 4.6: Planung des Glasbaus

Abbildung 4.6 zeigt die Planungsschritte für ein Gewerk des Glasbaus. Die Übersichtsplanung liefert relativ grobe Vorgaben bezüglich der Abmessungen der Bauelemente, der Art und Lagerung der Verglasung sowie eventuell besondere Anforderungen, zum Beispiel für das Resttragverhalten oder den Brandschutz. Aus der statischen Berechnung und Glasbemessung, die in der Regel mit FE-Programmen durchgeführt wird, ergibt sich die erforderliche Glasdicke. In der Ausführungsplanung legt der Fachplaner mit einem CAD-System daraufhin die Art, Form, Anordnung und die zu verwendenden Bauprodukte für Verglasung, Befestigung und Zubehör fest.

4.2.7 Besondere Anforderungen

In den vorangegangenen Abschnitten wurde wiederholt der Begriff der *besonderen Anforderung* genannt. In dieser Arbeit ist damit immer eine Anforderung gemeint, die an ein Bauelement im Hinblick auf die weitere Fachplanung gestellt wird. Das folgende Beispiel soll dies anhand eines Wärmeschutznachweises veranschaulichen, siehe Abbildung 4.7.

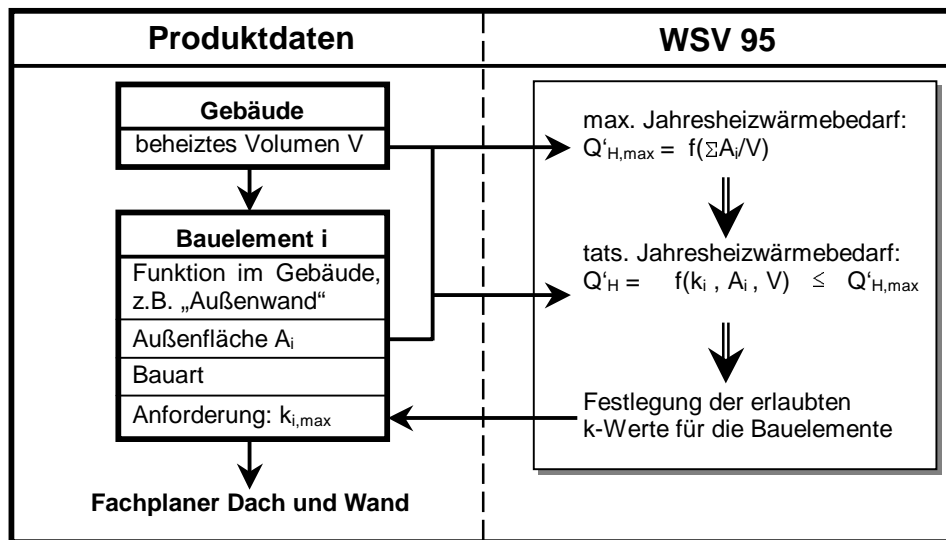


Abbildung 4.7: Beispiel für eine Anforderung im Wärmeschutz

Die Produktdaten auf der linken Seite der Abbildung stellen ein Gebäude in Form einer Instanz für das Gebäude selbst und mehrerer Instanzen für Bauelemente dar. Aus diesen Daten können die für den Nachweis gemäß Wärmeschutzverordnung 95 [WSV 95] erforderlichen Eingangswerte herausgelesen werden. Aus dem Nachweis, der auf der rechten Seite der Abbildung skizziert ist, ergeben sich für die Bauelemente maximal erlaubte Wärmedurchgangskoeffizienten, die als Produktdaten den Instanzen der Bauelemente hinzugefügt werden ($k_{i,max}$). Da die Erfüllung dieser Werte letztlich von der späteren Fachplanung abhängt, soll an dieser Stelle nicht von tatsächlichen Eigenschaften gesprochen werden, sondern lediglich von Anforderungen.

Weitere, typische Anforderungen in diesem Sinne ergeben sich aus korrosions-, schall- und brandschutztechnischen Gründen.

4.3 Technische Beschreibung der Gewerke

4.3.1 Allgemeines

Die folgende Beschreibung der Gewerke aus technischer Sicht legt die Inhalte für die Modellierung fest. Aus diesem Grunde sind die folgenden Erläuterungen ausführlich gehalten. Bei sehr großer Produktvielfalt wird jedoch auf entsprechendes Schrifttum verwiesen. Es wird primär nach grundlegenden Bauarten und erst sekundär nach Bauteilen unterschieden, die verwendet werden. Die Gewerke sollen so detailliert dargestellt werden, wie es gemäß der Anwendungsszenarien für den Bauplaner relevant ist. Bauprodukte werden als Einheiten behandelt. Das heißt, daß ihre Herstellung nicht von Bedeutung ist.

Dieser Abschnitt ist das Ergebnis einer umfassenden Marktsichtung, die anhand eines Bauinformationssystems durchgeführt wurde [53]. Bei der Typisierung der gesichteten Bauprodukte wurden soweit wie möglich die einschlägigen technischen Normen berücksichtigt.

4.3.2 Gewerkeübergreifendes Modell

Um die in den folgenden Abschnitten beschriebenen Gewerke im Datenmodell zu verankern, muß es ein gewerkeübergreifendes Teilmodell geben, das modelltechnisch denselben Stellenwert hat wie die gewerkespezifischen Teilmodelle. Dieses gewerkeübergreifende Modell muß als kleinsten gemeinsamen Nenner die Bauelemente aller Gewerke in grober Detaillierung darstellen können. Dieses sind Dächer, Decken, Außenwände, Träger, Stützen, Baugruppen des Stahlbaus, Glasfassaden, Einzel- und Streifenfundamente, Bodenplatten, Türen, Fenster, Industrietore, Dachbelichtungselemente sowie Rauch- und Wärmeabzugsanlagen (RWA). Ihre Darstellung muß Form und grobe Abmessungen, grundsätzliche Bauart, Funktion innerhalb des Gebäudes (zum Beispiel tragende oder raumbildende Aufgaben) sowie die besonderen Anforderungen beinhalten.

Neben der Funktion zur Integration der gewerkespezifischen Modelle bietet es sich an, das gewerkeübergreifende Modell auch für den Produktdatenaustausch auf niedrigem Detaillierungsniveau zu nutzen. Es handelt sich jedoch nicht um ein „Architekturmodell“ und berücksichtigt weder architektonische noch ästhetische Gesichtspunkte.

4.3.3 Dach und Wand

4.3.3.1 Bauarten und Anwendungen

Das Gewerk *Dach und Wand* umfaßt Dach- und Außenwandbekleidungen in Elementbauart. Die Bauelemente dieses Gewerkes haben in der Regel keine primäre Tragfunktion im Gebäude; sie leiten lediglich die Wirkungen von Schnee, Wind und ihr eigenes Gewicht in das Tragwerk weiter. Der klassische Stahlbeton- und Stahlbetonfertigteilbau oder der Mauerwerksbau gehören nicht zum Gewerk Dach und Wand.

Der wichtigste Bereich von Dach und Wand in diesem Sinne ist der Leichtbau, besonders der Stahlleichtbau. Für ihn ist im Gegensatz zur im klassischen Stahlbau vorherrschenden Verwendung genormter Profilformen die sehr große Vielfalt an Bauprodukten kennzeichnend. Das Gewerk Dach und Wand beinhaltet außerdem die Bauart „Gasbeton“ und die ebenfalls sehr große Palette von Anbau- und Sonderteilen. Nicht eingeschlossen sind dagegen Trennwände in Schichtaufbau (zum Beispiel

mit Gipskartonplatten) und die im Geschößbau verbreiteten Verbunddecken mit Trapezprofilen. Die Bauelemente für Dach und Wand werden bezüglich der tragenden Bauteile in vier Bauarten unterschieden.

a) Trapezprofilbauart (Leichtbau):

Raumabschließende Bauteile in Trapezprofilbauart bestehen in der Regel aus mehreren Schichten mit getrennten Funktionen für Tragwirkung, Dampfsperre, Wärmedämmung und Abdichtung. Die tragenden Trapezprofile können aufgrund ihrer Wasser- und Dampfdichtigkeit jedoch auch die Funktionen anderer Schichten übernehmen, die dann wegfallen. Die Schichten wiederum sind in Verlegeebene in Tafeln, Paneele oder Bahnen aufgelöst. Die Bezeichnung Trapezprofil soll auch Sonderformen wie Well-, Klemmrippen- oder Stehfalzprofile mit einschließen.

b) Stahlkassettenbauart (Leichtbau):

Kassettenprofile werden hauptsächlich in Außenwänden, seltener auch in Dächern eingesetzt. Die Anordnung der Kassettenstege ist in Wänden stets horizontal. Der Aufbau ist immer zweischalig, da die tragende Kassettschicht nicht wasserdicht ist. Typische Deckschichten bestehen aus Trapez- oder Wellprofilen. Bei horizontaler Spannrichtung der Trapez- oder Wellprofile werden konstruktive Ergänzungen erforderlich. Die Wärmedämmung kann vor den Kassettenstegen hängend ausgeführt werden, oder die Dämmelemente können zwischen den Stegen eingebaut werden. Im zweiten Fall sollten zur Entschärfung von Wärmebrücken thermische Trennstreifen zwischen Trag- und Deckschicht angebracht werden.

c) Sandwichbauart (Leichtbau):

Sandwichelemente werden gleichermaßen für Dächer und Außenwände von Industriebauten verwendet. Im Gegensatz zu den beiden zuvor beschriebenen Bauarten gibt es jedoch nur eine einzige Schicht, die alle Funktionen vereint.

d) Gasbetonbauart:

Da bei dieser Bauart vorgefertigte Montagebauteile eingesetzt werden, wird sie dem hier betrachteten Gewerk zugeordnet, und nicht etwa dem Stahlbetonbau. Die Gasbetonmontagebauteile bilden dabei in der Regel einen einschichtigen Wandaufbau, da sie über sehr gute Eigenschaften bezüglich Wärmedämmung sowie Brand-, Schall- und Feuchtigkeitsschutz verfügen. Sie werden liegend oder stehend angeordnet, mit speziellen Laschen, Schlaufen oder Ankerbolzen befestigt, verfugt, eventuell vergossen und zum Wetterschutz beschichtet. Alternativ zum einschichtigen Aufbau kann eine Metallfassade vorgehängt werden, zum Beispiel aus Trapezprofilen oder speziellen Fassadentafeln.

Gute und knappe Übersichten zu allen vier Bauarten sind in den „Stahlbau-Arbeitshilfen“ zu finden [6]. Detailliertere Informationen zu den Bauarten a) bis c) bieten außerdem *Baehre* [4] sowie die Schriftenreihe „IFBS-Info“ des Industrieverbandes zur Förderung des Bauens mit Stahlblech e.V. [66] bis [71]. Einen umfassenden Überblick über die Anwendung von Sandwichelementen gibt eine spezielle Empfehlung der Europäischen Konvention für Stahlbau [37]. Die Verwendung hinterlüfteter Außenwandbekleidungen ist in DIN 18516 geregelt [26].

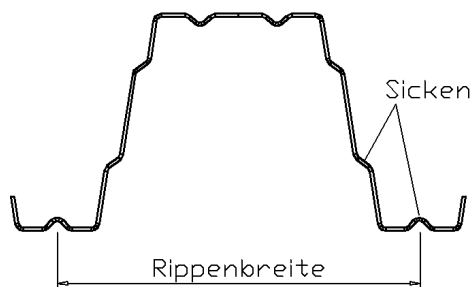
4.3.3.2 Bauteile

Trapezprofile:

Trapezprofile sind Tafeln, die in Längsrichtung gerippt sind. Diese *Makroprofilierung* gibt dem Bauteil eine definierte Tragrichtung. Zusätzlich können die Profile Längssicken aufweisen (*Mikroprofilierung*), die als Beulversteifungen dienen. Trapezprofile werden meist aus verzinktem Stahlblech, seltener auch aus Aluminium hergestellt. Sowohl Stahl- als auch Aluminiumtafeln können für erhöhten Korrosionsschutz oder zur Farbgebung zusätzlich zur Verzinkung ein- oder beidseitig organisch beschichtet sein. Einen Überblick über Herstellungsformen von Stahltrapezprofilen gibt *IFBS-Info 3.02* [68]. Die Länge der Tafeln ist nur aus Transportgründen begrenzt, üblicherweise werden Längen bis 24 m angeboten. Die Tafeln sind in der Regel eben, beispielsweise für Bogendächer werden sie jedoch auch in Längsrichtung gekrümmt hergestellt.

Abbildung 4.8 zeigt Querschnitt und Gesamtansicht einer typischen Trapezprofiltafel. Um Dichtheit bei Verwendung als wasserführende Schicht zu erreichen, werden die Tafeln überlappend gestoßen. Die Entwässerungs- und die Hauptwetterrichtung bestimmen daher die beiden Montagerichtungen. Durch ausreichende Überlappungen und Befestigungen können Querstöße biegesteif ausgeführt werden. Trapezprofiltafeln sind in Positivlage, wenn der gestoßene Gurt unten ist, und in Negativlage, wenn er oben ist. Diese Definition entspricht den gängigen Normen. Beim Verlegen können die Trapezprofiltafeln in Querrichtung geringfügig gedehnt oder gestaucht werden.

a) Querschnitt:



b) Verlegetafel (Positivlage):

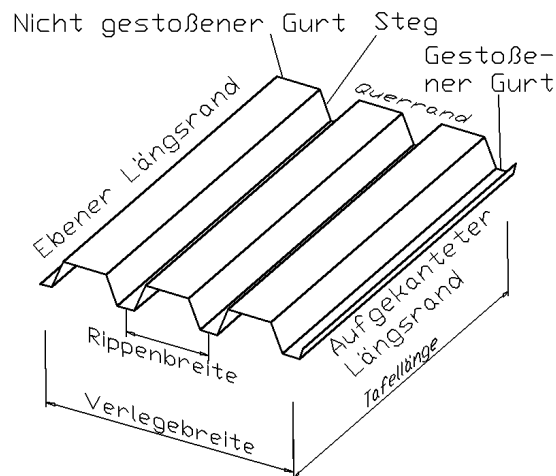


Abbildung 4.8: Trapezprofiltafel

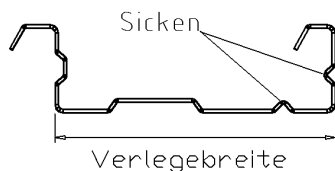
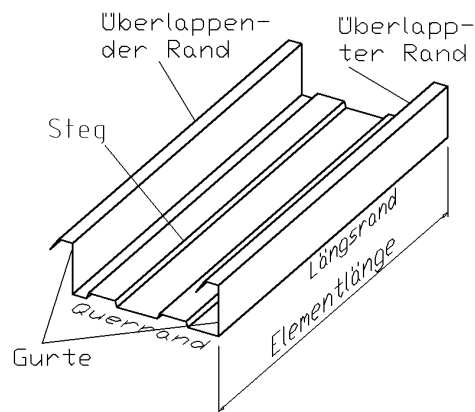
Die deutschen und europäischen Normen für Trapezprofile sind im Schrifttum angegeben [28], [34], [35], [36], [39], [40], [94].

Trapezähnliche Profile:

Insbesondere für die Außenschichten von Dächern existieren Sonderprofile, die sich in Form und Einbau von den Trapezprofilen im eigentlichen Sinne unterscheiden. Dies sind zum Beispiel Fassadenpaneele, Pfannenbleche, Well-, Klipp-, Falz- und Steckfalzprofile. Sie werden aus verzinktem Stahlblech, Aluminium, Zink, Kupfer, Faserzement oder transparenten Kunststoffen hergestellt. *IFBS-Info 3.04* gibt einen Überblick über die üblichen Herstellungsformen trapezähnlicher Stahlprofile [70].

Kassettenprofile:

Kassettenprofile sind langgestreckte Bauteile aus verzinktem und gegebenenfalls zusätzlich organisch beschichtetem Stahlblech, die durch seitliches Aneinanderfügen flächige Bauelemente mit definierter Tragrichtung bilden. Der C-förmige Querschnitt ist für zusätzliche Beulsteifigkeit in der Regel längsgesickt (Mikroprofilierung). Die Längsränder sind in der Regel unterschiedlich ausgebildet, so daß sie an den Stößen überlappen. Die Querstöße werden dagegen nicht überlappend ausgeführt. Im Gegensatz zu Trapezprofilen gibt es keine Positiv- oder Negativlage: Die Elemente werden immer mit dem Steg zur Unterkonstruktion eingebaut. Die Länge der Elemente ist nur aus Transportgründen begrenzt. Ein Überblick über Herstellungsformen von Stahlkassettenprofilen ist in *IFBS-Info 3.01* gegeben [67]. Für Kassettenprofile gilt die gleiche Normung wie für Trapezprofile. Abbildung 4.9 zeigt ein typisches Kassettenelement in Querschnitt und Gesamtansicht.

a) Querschnitt:**b) Kassettenelement:****Abbildung 4.9: Kassettenelement****Sandwichelemente:**

Sandwichelemente bestehen aus zwei metallischen Deckschichten, die über einen Kern schubsteif verbunden sind. Am weitesten verbreitet sind Elemente mit Deckschichten aus verzinktem Stahlblech und einem PUR-Schaumkern. Seltener wird für beide oder nur eine Deckschicht Aluminium verwendet, oder als Kernmaterial Mineral- oder Glaswolle. In [37] werden als Kernmaterialien außerdem PIR-, PS- oder PF-Kunststoffe angeführt. Die Deckschichten können zusätzlich zur Verzinkung organisch beschichtet sein.

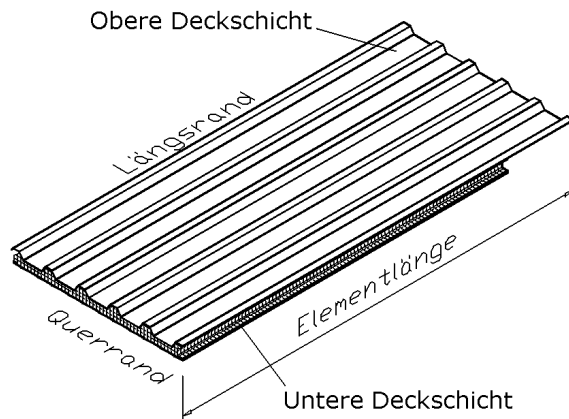
Die Deckschichten können zur Erhöhung der Tragfähigkeit wie Trapezprofile gerippt sein (Makroprofilierung). Dies ist bei Dachelementen wegen des Schubkriechens des Kunststoffkerns unter Langzeitbelastung üblich. Üblicherweise wird nur die äußere Schicht profiliert ausgeführt, da eine ebene innere Schicht einen besseren Raumabschluß ermöglicht. Wandelemente haben in der Regel zwei ebene Deckschichten.

Beide Deckschichten können aus optischen Gründen zusätzlich genutet oder mikroliniert sein. Einen umfassenden Überblick über Profilformen von Sandwichelementen gibt *IFBS-Info 3.03* [69]. Sandwichelemente werden endlos gefertigt, die Elementlänge ist nur aus Transportgründen begrenzt.

Da Sandwichelemente fast immer als Außenhaut eingesetzt werden, müssen alle Stöße wasserdicht sein. Die Längsränder der Elemente werden in der Regel ab Werk zueinander passend ausgebildet, zum Beispiel als Nut und Feder, mit überlappender oberer Deckschicht oder als Kombination von beiden. Stöße an Querrändern werden

entweder mit überlappender Deckschicht oder als Stumpfstoß mit Deckblechen ausgeführt. In beiden Fällen werden Dichtungsbänder eingelegt. Eine ausführlichere Typisierung von Stößen ist in [37] zu finden. Abbildung 4.10 zeigt typische Sandwichelemente für Dach und Wand.

a) Dachelement:



b) Wandelemente:

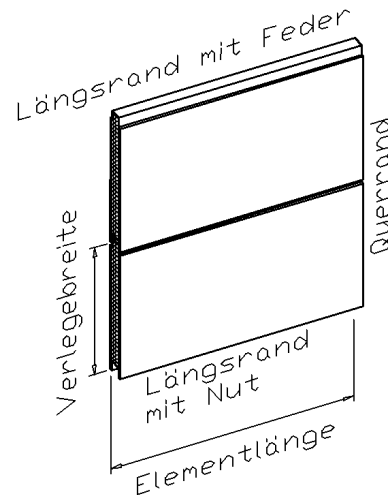


Abbildung 4.10: Sandwichelemente für Dach und Wand

Gasbetonsteine:

Montagebauteile aus dampfgehärtetem Gasbeton sind entweder als Wandplatten, die Nut und Feder haben können, als Sturzwandplatten über Tor und Fensteröffnungen oder als geschoßhohe Wandtafeln verfügbar.

Fassadentafeln zur Außenwandbekleidung:

Vorgehängte Fassadentafeln haben im Rahmen des hier betrachteten Gewerkes Dach und Wand nur in Verbindung mit Gasbetonwänden eine Bedeutung. Metallische Tafeln aus Stahl- oder Aluminiumblech können entweder als ebene Bauteile ausgebildet sein, die über Profilschienen oder einzelne Wandanker an der Tragschicht befestigt werden, oder als sogenannte kofferförmige Bauteile mit abgekanteten Rändern und eingeschnittenen Haken, siehe Abbildung 4.11. Diese Haken können in eine entsprechende Unterkonstruktion eingehängt werden.

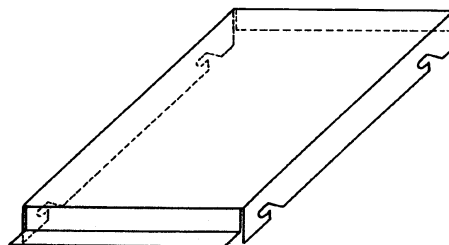


Abbildung 4.11: Kofferförmige Fassadentafel [38]

Dämmelemente:

Zur Wärmedämmung werden plattenförmige Materialien wie Leichtbauplatten, Polystyrol-Partikelschaum, Korkplatten oder bahnenförmige Werkstoffe wie Mineralfaserwolle verwendet. Sie werden zum Teil auf darunterliegende Schichten aufgeklebt. Bei Kassettenwänden oder mehrschaligen Bauteilen mit Abstandsprofilen

werden Dämmelemente oft zwischen die Profildurte gelegt, so daß die Dämmschicht nicht durchgängig ist.

Dachhaut, Dichtungsschicht:

Sofern die äußerste Schicht eines Bauelementes nicht als wasserdichte Trapez- oder Wellprofilschicht ausgebildet ist, werden zur Abdichtung Bitumendachbahnen (in der Regel mehrlagig) oder Hochpolymerbahnen verwendet. In DIN 18531 werden alle üblichen Werkstoffe aufgeführt [27].

Schalungen, Verblendungen:

Schalungen dienen als tragender Untergrund zum Beispiel für Stehfalzdächer. Als Verblendungen bezeichnet man die innersten Schichten beispielsweise bei abgehängten Decken. Verwendung finden Blechprofile, Gips- oder Gipskarton-, Sperrholz-, Preß- oder Faserplatten.

Abstandhalter:

Linienförmige Abstandhalter bestehen aus leichten Kaltprofilen oder Kanthölzern. Sie werden bei mehrschaligen Dächern und Wänden zum Beispiel zwischen zwei Trapezprofilschichten verlegt. Mehrere Lagen aus Abstandprofilen können sich kreuzen.

Dichtungsbänder, Trennstreifen:

Dichtungsschnüre werden zur Abdichtung von Fugen in Kassettenwänden verwendet, Dichtungsbänder bei Überlappstößen von Sandwichelementen. Thermische Trennstreifen aus PE, PVC, Mineralwolle oder Hartfaser werden zur Entschärfung von Wärmebrücken bei direktem Kontakt von metallischen Innen- und Außenschalen eingesetzt, zum Beispiel bei Kassettenwänden.

Linienförmige Zubehörteile:

Linienförmige Anbauteile aus Kaltprofilen werden für An- und Abschlußprofile, Eckausbildungen, Tropfkanten, Dachrinnen, Trauf- und Firstbleche, Attikakappen, Randversteifungsbleche, Randwinkel, Verblendungen u.v.a.m. eingesetzt. Besonders erwähnt werden sollen tragende Bauteile für Auswechslungen an Durchbrüchen. Hier werden neben Kaltprofilen häufig Kanthölzer verwendet.

Sonstige Zubehörteile:

Bei kleinen Durchbrüchen genügt nach DIN 18807 die Verwendung von Verstärkungsblechen statt einer Auswechslung. Weitere Teile für Aufbauten, Abhängungen, Abläufe, Regenfalleitungen und Dachrinnen sowie deren Befestigung sollen hier außer acht gelassen werden.

Befestigungselemente:

Für die Baustellenverwendung existieren folgende punktförmige Verbindungsmittel zur Befestigung von Blechen untereinander und auf der Unterkonstruktion: gewindefurchende Schrauben, gewindeschneidende Schrauben, Blechtreibschrauben, Bohrschrauben, Holzschrauben, Blindniete und Setzbolzen. In *IFBS-Info 7.01* sind alle in Deutschland bauaufsichtlich zugelassenen Verbindungselemente angegeben [71]. Im Schrifttum aufgeführt, jedoch in der Praxis bei flächigen Bauelementen selten verwendet, werden Punktschweißungen, Rollnahtschweißungen, Clinchverbindungen und Klebungen.

Bei Trapezprofilen in Negativlage (Verbindung in der Hochrippe) kommen sogenannte Kalotten zum Einsatz. Stehfalzdächer werden mit sogenannten Haften befestigt. Speziell für Sandwichelemente existieren sogenannte *stand-off screws* mit jeweils einem Gewinde für äußere Deckschicht und Unterkonstruktion.

4.3.3.3 Bemessung

Die oben aufgeführten tragenden Bauelemente werden überwiegend durch rechtwinklig zu ihrer Ebene wirkende Lasten beansprucht. Durch die Profilierung weisen sie eine eindeutige Tragrichtung auf, weshalb Schnittgrößen im allgemeinen nach der Balkentheorie ermittelt werden. Auf Elemente in steilen Dächern oder Wänden können zusätzlich Normalkräfte wirken. Außerdem können die flächigen Bauteile bei ausreichender Befestigung Schubfelder bilden und beispielsweise Dachverbände ersetzen.

Die Bemessung von Leichtbauelementen ist im allgemeinen kompliziert, da einerseits dünnwandige Querschnittsteile infolge Beulens planmäßig ausfallen, siehe DASt-Richtlinie 016 [14] und ENV 1993-1-3 [40], und andererseits zum Beispiel Sandwichelemente ein zeitabhängiges Tragverhalten aufweisen. Für bauaufsichtlich zugelassene Bauteile werden jedoch Bemessungstabellen vom Hersteller angeboten, die dem Statiker die Arbeit erleichtern. Dies gilt auch für deren Verbindungen.

4.3.3.4 Besondere Anforderungen

Besondere Anforderungen für raumabschließende Bauteile können an Korrosionsschutz, Brandschutz, Schalldämmung und -schluckung gestellt werden. Verbesserte Korrosionsbeständigkeit von Stahl- oder Aluminiumblech – DIN 18807 unterscheidet die Klassen I bis III – kann durch zusätzliche organische Beschichtungen erzielt werden. Um höhere Feuerwiderstandsklassen zu erreichen, müssen geeignete Materialien in ausreichender Dicke oder feuerhemmende Bauteile verwendet werden. Zur besseren Schalldämmung können außer größeren Bauteildicken zum Beispiel Kies-schüttungen auf Dächern beitragen. Für erhöhte Schallabsorption werden gelochte Trapez- und Kassettenprofile verwendet, die zusätzlich mit schallabsorbierendem Material gefüllt werden können.

Die Einhaltung der Forderungen bezüglich Wärme- und Tauwasserschutz ist zwar selbstverständlich, die entsprechenden Nachweise sind aber besonders bei Kassettenwänden wegen der Wirkung der Kassettenurte als Wärmebrücken nicht trivial. An der Lösung dieses Problems wird noch gearbeitet [13], [83a], [83b], [91].

4.3.4 Dach- und Wandöffnungen

4.3.4.1 Allgemeines

Die genaue Ausführung und der Zusammenbau für die verschiedenen Öffnungsgewerke ist zum größten Teil fest vorgegeben, da in der Regel komplette Produkte angeboten werden. Aus diesem Grunde ist es in der Praxis üblich und für das Produktmodell ausreichend, daß die Bauteile lediglich mit den für die Gebäudehülle maßgebenden lichten Abmessungen, gegebenenfalls zusätzlich erforderlichen Unterkonstruktionen, geeigneten Befestigungsmaßnahmen und mit Angabe der Herstellerbezeichnung dargestellt werden. Eine darüber hinausgehende Detaillierung ist nicht notwendig. Dementsprechend vereinfacht werden im folgenden lediglich die verschiedenen Typen der Öffnungsgewerke beschrieben. Eine Ausnahme stellt der Fensterbau dar, bei dem die verwendeten Profile individuell kombiniert werden können. Die Fenstergröße ist somit innerhalb bestimmter Grenzen frei wählbar.

4.3.4.2 Fenster und Fenstertüren

DIN 12519 teilt Fenster in Einfachfenster, Verbundfenster und Doppelfenster (Kastenfenster) ein [21]. Weitergehend kann nach der Anzahl der Flügel und der Öffnungsart unterschieden werden, siehe ebenfalls DIN 12519. Rahmen und Zargen für Fenster und Fenstertüren werden in der Regel aus speziellen Profilen aus Holz, Kunststoff, Aluminium oder Stahl individuell angefertigt. Zur Verglasung stehen Einfach-, Isolier- und Sicherheitsverglasungen zur Verfügung. Als Zubehör werden neben Dichtungen und Beschlägen auch Rolläden, Rollädenkästen und Fensterbänke verwendet.

Die maßgebenden Konturen eines Fensters sind die lichte Fensteröffnung, die Zargenaußenkontur sowie die Rohbaulichte im Hinblick auf den Einbau des Fensters in Dach und Wand. Bei Industriehallen und Bürogebäuden werden oft mehrere Fenster in eine Wandöffnung eingebaut, so daß eine sogenannte *Fensteranordnung* entsteht, wie es in Abbildung 4.12 zu sehen ist. Fensterwände gemäß DIN 18056 werden dagegen dem Gewerk des Glasbaus zugeordnet [23].



Abbildung 4.12: Fensterreihen

Spezielle Anforderungen können an Wärmedurchgangswiderstand, Schalldämmung, Einbruchswiderstand (zum Beispiel nach DIN 1627 [19]) Fugendurchlässigkeit und Schlagregendichtigkeit (zum Beispiel nach DIN 18055 [22]) gestellt werden.

4.3.4.3 Türen

Für Türen gilt im wesentlichen das zu Fenstern gesagte. In DIN 12519 werden Türen in erster Linie nach der Funktion in Fenster-, Innen-, Außen-, Wohnungseingangs- und Brandschutztüren eingeteilt [21]. Eine feinere Unterscheidung ergibt sich

aus der Öffnungsart, siehe ebenfalls DIN 12519. Für Türen gelten die gleichen Anforderungen wie für Fenster, zusätzlich können spezielle Brand- und Rauchschutzeigenschaften nach DIN 18095 gefordert werden.

4.3.4.4 Industrietore

Für Industrietore ist der seltene Fall gegeben, daß eine umfassende Typisierung und Begriffsbestimmung in Form von DIN 12433 existiert [20]. Die Norm unterscheidet Tore für Industrie- und Hallenbau nach ihrer Bewegungsart und nach der Zahl der Torflügel. Ein weiteres Unterscheidungskriterium ist die Betätigungsart des Tores (Handbetätigung, Kraftbetätigung, Betätigung durch Schwerkraft, Energiespeichereinrichtung, automatischer Betrieb, Notbetätigung). Maßgebende Konturen sind die Gebäudeöffnung und die nutzbare Öffnung. Besondere Anforderungen werden gegebenenfalls an Brand- oder Strahlenschutzture gestellt.

4.3.4.5 Dachbelichtung

Dachbelichtungen sind spezielle Bauelemente zum Durchlassen von Tageslicht in Dächern von Industriebauten. Im folgenden wird zwischen Lichtkuppeln, Lichtpyramiden, Tonnengewölbe- und Satteldachlichtbändern sowie selbsttragenden Lichtbändern unterschieden.

Lichtkuppeln haben einen kreisförmigen oder polygonalen Grundriß und eine lichtdurchlässige, kuppel- oder pyramidenförmige Abdeckung, die einteilig ist und in der Regel auf einem Aufsetzkranz montiert wird. Lichtpyramiden haben einen pyramidenförmigen Aufbau, der aus Profilen und Glasscheiben besteht. Alle genannten Aufbauten können starr oder zum Öffnen ausgebildet sein. Abbildung 4.13 zeigt Lichtkuppeln und -pyramiden.

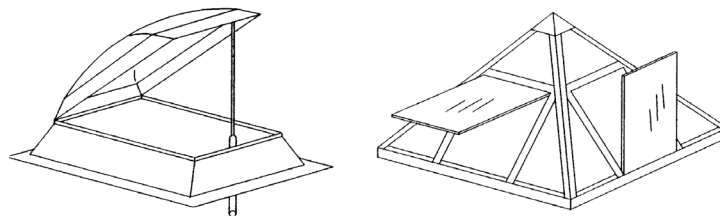


Abbildung 4.13: Lichtkuppel und -pyramide

Lichtbänder sind im Gegensatz zu Kuppeln und Pyramiden in ihrer Länge nicht begrenzt. Beim Tonnengewölbe-Lichtband tragen kreisförmig gebogene Metallsprossen, meist Aluminiumstrangpressprofile, die äußeren Lasten in Querrichtung ab. Auf diese Sprossen sind Hohlkammerplatten aus lichtdurchlässigem Polycarbonat aufgeklemmt oder aufgeschraubt, wodurch sie in die charakteristische Form des Lichtbandes gezwungen werden. Die Sprossen selbst sind auf einer Zargenkonstruktion gelagert, die in der Regel über Zugbänder in Querrichtung ausgesteift wird. Einzelne Elemente zwischen diesen Sprossen können geöffnet werden.

Satteldachlichtbänder haben einen dreieckigen Querschnitt. Sie bestehen aus einem Sprossenfachwerk, das die lichtdurchlässigen Elemente trägt. Dieses können Glasscheiben oder Polycarbonat-Hohlkammerplatten sein. In die schrägen Flächen können Öffnungen oder Zusatzgeräte eingebaut sein, beispielsweise Lüftungsgeräte.

Einen Sonderfall stellen die selbsttragenden Lichtbänder dar, die einen gewölbeförmigen Querschnitt aufweisen, bei denen die Lasten jedoch ohne Quersprossen in die

Zargenkonstruktion abgetragen werden. Sie sind nur für schmale Öffnungen geeignet. Abbildung 4.14 zeigt die drei Lichtbandtypen.

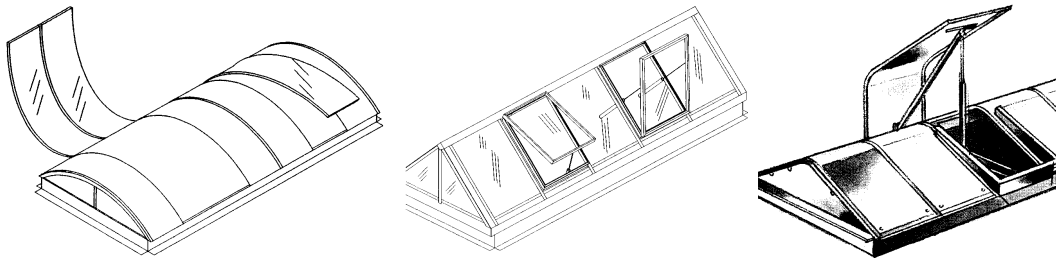


Abbildung 4.14: Tonnengewölbe-, Satteldach- und selbsttragendes Lichtband

Für alle Dachbelichtungselemente wird ein großes Spektrum an Zubehör angeboten, wie Verdunkelungen, Antriebe zum Öffnen, Ventilatoren u.a.. Weitere, eigentlich auch zum Thema Dachbelichtung gehörende Bauteile, wie lichtdurchlässige, profilierte Kunststoffplatten nach DIN 1013 [18] oder Dachflächenfenster, werden hier nicht behandelt. Die genannten Dachbelichtungselemente werden in der Regel als komplette Produkte mit allen Zubehörteilen angeboten. Deshalb sind für den Planer des Bauwerks nur der Typ der Dachbelichtung, die maßgebenden Konturen der Öffnungen (die lichte Öffnung sowie die Rohbauöffnung im Dach), die überschlägigen Abmessungen, besondere Anforderungen und eventuelles Sonderzubehör relevant. Alle weiteren Eigenschaften können über die Herstellerbezeichnung zugeordnet werden.

Für Dachbelichtungen sind durch die großen Öffnungen in den Tragelementen von Dächern in Leichtbauweise in der Regel Auswechselungen notwendig; diese werden jedoch dem Gewerk Dach und Wand zugeordnet.

4.3.4.6 Rauch- und Wärmeabzugsanlagen (RWA)

Rauch- und Wärmeabzugsanlagen sind Geräte, die im Brandfall Rauch und Wärme aus einem Gebäude abführen. DIN 18232 unterscheidet zwischen Rauch- und Wärmeabzügen und zwangsbelüfteten maschinellen Rauchabzügen [25]. Neben diesen speziellen Geräten können auch Fenster oder Elemente der Dachbelichtung die Funktion von Rauchabzügen übernehmen, sofern sie über Vorrichtungen zum selbsttätigen Öffnen verfügen. Bei der Dimensionierung der Anlagen ist der entscheidende Wert die aerodynamisch wirksame Öffnungsfläche A_w .

4.3.5 Stahlbetonbau

Die für den Industrie- und Hallenbau wichtigen Bauelemente aus Stahlbeton sind Hallenböden, Einzel- und Streifenfundamente, Innen- und Außenwände sowie Decken. Wie bereits in Abschnitt 4.2.5 begründet wurde, wird der Bereich der Bewehrung von Stahlbetonteilen im Produktmodell dieser Arbeit nicht dargestellt. Die somit verbleibenden Sachverhalte dieser Bauweise werden als allgemein bekannt vorausgesetzt. Für Hallenböden sei auf die Stahlbau-Arbeitshilfe 42.1 verwiesen [6]. Die relevanten Daten der Bauelemente sind deren Abmessungen, die verwendete Betonfestigkeitsklasse und die Bewehrungsstahlsorte. Letztere wird oft bereits in der Phase der Übersichtsplanung festgelegt, so daß sie auch ohne Kenntnis der eigentlichen Bewehrungsplanung von Bedeutung ist.

4.3.6 Glasbau

4.3.6.1 Allgemeines

Das Gewerk des Glasbaus beinhaltet im Gegensatz zum Fensterbau, der bei den Dach- und Wandöffnungen behandelt wurde, großflächige, raumabschließende Elemente des Bauwerks. Für das Produktmodell von Interesse sind Glasfassaden und Überkopfverglasungen bei Kuppeln, Vordächern und Dächern. Bauelemente können dabei vollständig verglast sein, wie die in Abbildung 4.15 gezeigte Glasfassade, oder lediglich teilweise, wie das in Abbildung 4.16 gezeigte alltägliche Beispiel aus dem Industriebau. Nicht Teil dieser Arbeit sind begehbare, betretbare und absturzsichernde Verglasungen oder Biegeträger und Treppenstufen aus Glas.

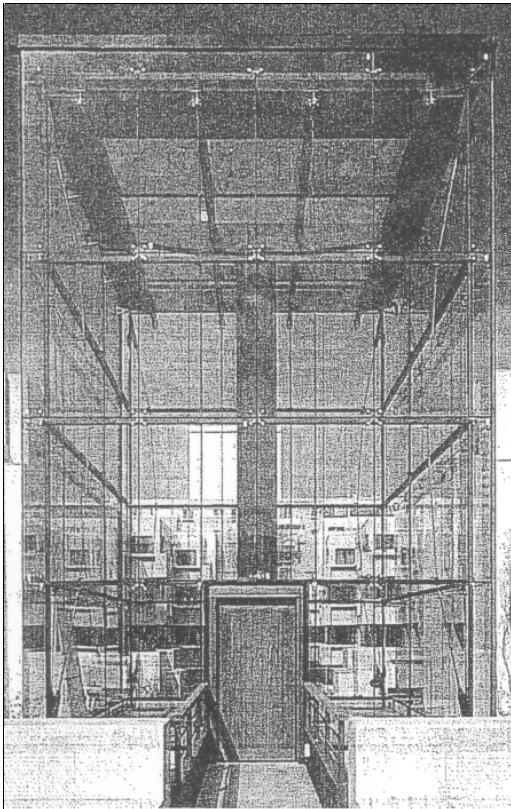


Abbildung 4.15: Glasfassade [98]



Abbildung 4.16: Verglaste Fläche

Ein geeignetes Kriterium für die Typisierung von Glaskonstruktionen ist die Art der Lagerung der Verglasungsbauteile, weil dadurch sowohl die zu verwendenden Bauteile als auch die Berechnungsarten und die anzuwendenden technischen Vorschriften bestimmt werden. Es kann unterschieden werden in punktförmige Lagerung, ideale Punkthalterung, mechanische Linienlagerung oder geklebte Linienlagerung (*Structural Sealant Glazing - SSG*). Nicht behandelt werden an dieser Stelle veraltete Bauarten wie Bleiverglasung oder „Falz mit Kittfuge“.

4.3.6.2 Bauteile

Verglasungsbauteile:

Im Bauwesen sind die Glasarten Floatglas (Spiegelglas), Gußglas (Ornamentglas), Borosilikatglas mit erhöhter Temperaturwechselbeständigkeit, durch Vorspannen erzeugtes Einscheiben-Sicherheits-Glas (ESG), teilvorgespanntes Glas (TVG) und Drahtglas (Gußglas mit Drahtnetz) gebräuchlich. Sie werden in den folgenden Erzeugnisformen angeboten:

- 1) Einfachverglasung aus Floatglas, Gußglas, Borosilikatglas oder Drahtglas
- 2) TVG aus Floatglas, Gußglas oder Borosilikatglas
- 3) ESG aus Floatglas, Gußglas oder Borosilikatglas
- 4) ESG wie oben mit Folie
- 5) Verbund-Sicherheits-Glas (VSG) mit PVB-Folienverbund aus Floatglas, ESG, TVG oder deren Kombination
- 6) Verbundglas (VG) mit Gießharzverbund
- 7) Isolierglas aus mehreren der oben genannten Einzelgläser mit evakuiertem Zwischenraum
- 8) Wärmeschutzglas: Isolierglas mit im Zwischenraum beschichteten Oberflächen

Die Glasarten Ziehglas (Fensterglas), Glaskeramik und die Herstellungsformen Drahtornamentglas, Profilbauglas und Glassteine sind im Industriebau von untergeordneter Bedeutung. In den allermeisten Fällen werden ebene Glastafeln verwendet, es existieren jedoch auch einfach und sogar doppelt gekrümmte Schalen. Die Tafeln können zwar beliebige Umrisse haben, sind jedoch meist rechteckig. Sie können gebohrt sein, die Löcher selbst können gefast oder für Senkungen vorbereitet sein. Glasoberflächen können emailliert, geätzt oder sandgestrahlt sein, Kanten können geschliffen, poliert oder bei ESG gebrochen sein. Für eine ausführliche Beschreibung der Glasarten und -erzeugnisse wird auf das Schrifttum verwiesen [9], [104], [108].

Befestigungen:

Mit Befestigungen werden hier diejenigen Bauteile bezeichnet, die die Verglasung und die Tragkonstruktion verbinden. Der Kontakt zur Verglasung ist in der Regel mittelbar über Dichtungen oder Verklotzungen. Die Vielfalt der angebotenen Befestigungsbauteile ist sehr groß.

1) Punktförmige Halterungen:

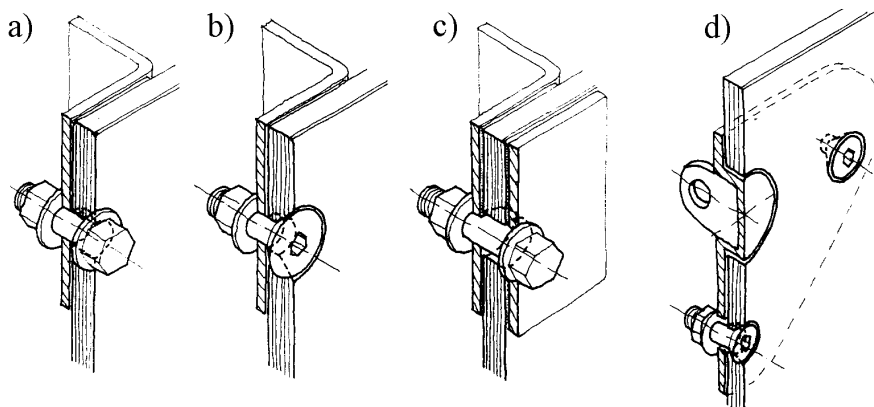


Abbildung 4.17: Punktförmige Halterungen [98]

Punktförmige Halterungen sind im Vergleich zum Verglasungselement kleine Bauteile, jedoch ohne die besonderen Eigenschaften des idealen Punkthalters. Abbildung 4.17 zeigt verschiedene Ausführungen: a) Halterung mit Schraube im Bohrloch, b) mit Senkschraube, c) Klemmplattenhalterung und d) als sogenannte *stud assembly*.

2) Ideale Punkthalter:

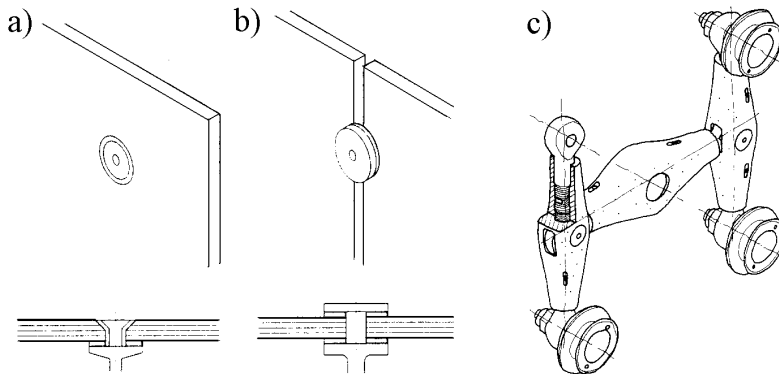


Abbildung 4.18: Ideale Punkthalter a) im Bohrloch, b) in der Fuge [104], c) Mehrfachhalter [98]

Ideale Punkthalter sind speziell für diesen Zweck entwickelte, zumeist rotations-symmetrische Bauteile, die ein ideales punktförmiges Auflager bilden, siehe Abbildung 4.18. Man unterscheidet in gelenkige und nicht gelenkige Halter: Zuerst genannte übertragen planmäßig nur Kräfte, letztere auch Momente. Weiter wird unterschieden nach der Anordnung in den Fugen zwischen den Verglasungstafeln oder in Bohrlochern sowie nach der Ausführung in flächenbündige und nicht flächenbündige Halter. Ideale Punkthalter werden oft in Mehrfachform zur Befestigung mehrerer benachbarter Verglasungstafeln hergestellt.

3) Linienlager:

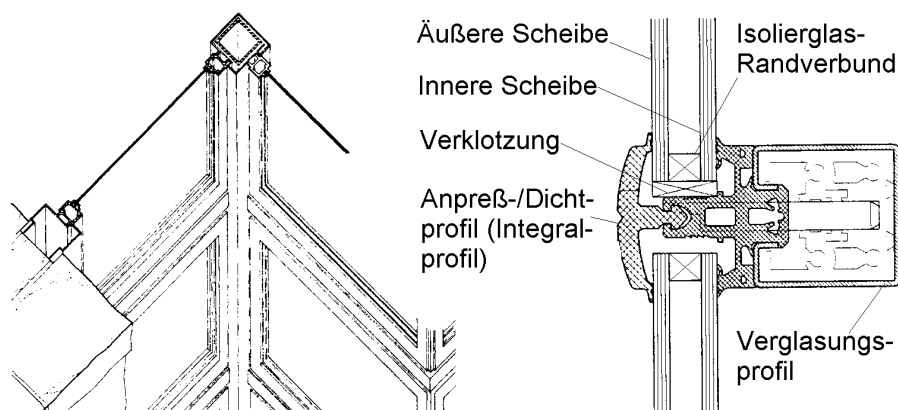


Abbildung 4.19: Linienlager [104]

Linienlager werden statisch als gehaltene Ränder der Glasplatte behandelt. Die im Glasbau verwendeten Linienlager, siehe Abbildung 4.19, bestehen aus einem Verglasungsprofil aus Stahl oder stranggepresstem Aluminium, in das die Verglasung mittels einer angeschraubten Preßleiste festgeklemmt ist. Das Verglasungsprofil bildet oft die Pfosten und Riegel einer Fassade. Innerhalb der Profile sind Dichtun-

gen und Verklotzungen zur Vermeidung eines unmittelbaren Kontaktes des Glases mit dem Metall verlegt.

4) Structural Sealant Glazing (SSG):

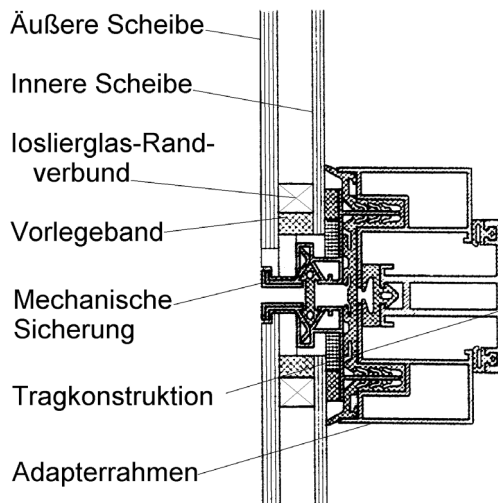


Abbildung 4.20: Structural Sealant Glazing [104]

Eine Besonderheit ist das *Structural Sealant Glazing*, bei dem die Verglasung in einen Adapterrahmen geklebt ist, der wiederum in die Tragkonstruktion eingehängt wird, siehe Abbildung 4.20. Die Klebefuge übernimmt dabei lastabtragende und dichtende Funktion. Zusätzlich dazu wird die Verglasung in der Regel noch mechanisch gesichert. Für diese Lagerungsart existiert eine spezielle EOTA-Leitlinie [41]. Die Befestigungsarten Schweißen von Glas an Glas und Lötten von Metall an Glas sind im Bauwesen ohne Bedeutung.

Dichtungen:

Dichtungen zwischen Verglasungsbauteil und Befestigung sind entweder Dichtringe in den Bohrlöchern von idealen Punkthaltern oder linienförmige Dichtungen bei Linienlagern, z.B. dauerelastische Dichtprofile aus EPDM/APTK oder Silikon.

Verfugung:

Als Verfugung werden linienförmige dichtende Teile zwischen freien Rändern von Verglasungsbauteilen bezeichnet, zum Beispiel bei Fugen und rahmenlosen Ecken punktgelagerter Verglasungsbauteile. Es werden Profile aus EPDM/APTK oder extrudiertem Silikon verwendet.

Verklotzung:

Sobek verwendet den Begriff der Verklotzung für untergelegte Teile, die das Glasgewicht in die Glashalteleisten oder Verglasungsprofile abtragen [104]. Die Klötze sind in der Regel aus Holz oder Kunststoff.

Sonstige Bauteile

Sonstige Bauteile sind Abstandshalter, Anschluß- oder Begrenzungsprofile von Fassaden.

Tragkonstruktion:

Im Gegensatz zur Befestigung hat die Tragkonstruktion keinen Kontakt zum Verglasungsbauteil, auch nicht mittelbar über Dichtungs- oder Verklotzungselemente. Sie wird im Rahmen dieser Arbeit deshalb nicht dem Gewerk „Glasbau“ zugeordnet.

4.3.6.3 Bemessung

Charakteristisch für das Tragverhalten von Glas ist dessen Sprödigkeit und die dadurch bedingte Herabsetzung der Zugfestigkeit in Folge selbst kleinster Imperfektionen auf der Oberfläche. Dies bestimmt sowohl die Berechnungsmethoden als auch die vergleichsweise hohen Sicherheitsanforderungen. Die Tragfähigkeit von biegebeanspruchten Gläsern kann durch Nachbearbeitungen wie Vorspannen erheblich gesteigert werden. Die Berechnung der Schnittgrößen für die Haupttragwirkung von Verglasungsbauteilen als Platte kann in Standardfällen tabellarisch erfolgen. Bei komplizierteren Geometrien und bei lokalen Kräfteinleitungen, zum Beispiel bei Punkthaltern, muß auf FE-Programme zurückgegriffen werden. Üblicherweise wird heutzutage nach dem Konzept der zulässigen Spannungen bemessen. *Sobek* nennt als weitere, dem Werkstoff besser gerecht werdende Bemessungsverfahren das Konzept der Bruchwahrscheinlichkeit und das Konzept der Grenzzustände [104].

Das Tragverhalten und die Bemessung von Glas ist im Schrifttum ausführlich erläutert, siehe beispielsweise *Sobek* [104] oder *Bucak* [9]. In der Normung sind diese neuesten technischen Entwicklungen zum Teil noch nicht erfaßt, so daß sie nur Teilgebiete der Bemessung von Tragelementen aus Glas abdeckt. Die technischen Regeln des Deutschen Instituts für Bautechnik (DIBt) decken zum Beispiel lediglich bestimmte Glasarten mit Linienlagerung ab [16], [17]. *Charlier* gibt eine Bestandsaufnahme der deutschen Normung im Glasbau [10].

4.3.6.4 Besondere Anforderungen

Wegen des im Bruchzustand zu erwartenden plötzlichen Versagens werden für bestimmte Einsatzgebiete von Glas besondere Forderungen zum Schutz von Personen gestellt. Für Überkopfverglasungen zum Beispiel müssen Sicherheitsgläser verwendet werden.

Brandschutzverglasungen müssen eine ausreichende Wechseltemperaturfestigkeit aufweisen, die durch die Verwendung von Borosilikatgläsern erreicht werden kann. Es wird in Deutschland in die Feuerwiderstandsklassen G30 bis G120 gegen Flammen- und Brandgasdurchtritt und F30 bis F120 gegen Durchtritt von Hitzestrahlung eingeteilt.

Weitere Anforderungen können bezüglich Färbung, Lichtdurchlässigkeit, Schallschutz, Schlagregendichtigkeit (zum Beispiel nach DIN 18055 [22]), Wärme- oder Energiedurchlässigkeit gestellt werden.

4.4 Zum Nutzen eines mechanischen Teilmodells

In den vorangegangenen Abschnitten wurde neben der Konstruktion der unterschiedlichen Gewerke auch deren Bemessung behandelt. Es stellt sich somit die Frage, in wieweit ein „mechanisches“ Teilmodell zur Darstellung von statischen Systemen, Einwirkungen, Berechnungsergebnissen oder sogar der Bemessung sinnvoll ist, und ob entsprechende Systeme aus dem Konstruktionsmodell erzeugt werden können. Erfahrungen mit der Anwendung der Produktschnittstelle Stahlbau [32] haben gezeigt, daß diese automatische Erzeugung sehr aufwendig zu programmieren ist. Im Gegensatz zu der dortigen Anwendung im Zusammenhang mit dem Stahltragwerk hat die Statik im Komplettbau einen viel geringeren Stellenwert, so daß dieser Aufwand nicht ohne weiteres gerechtfertigt ist.

Exemplarisch wird die Schnittgrößenermittlung einer Wand aus Sandwich-Elementen betrachtet, siehe Abbildung 4.21. Das System wird im maßgebenden Schnitt auf einen Durchlaufträger reduziert. Der Einsatz eines Stabwerksprogramms ist dadurch gerechtfertigt, daß es sich um ein statisch unbestimmtes System mit ungleichmäßigen Stützweiten handelt.

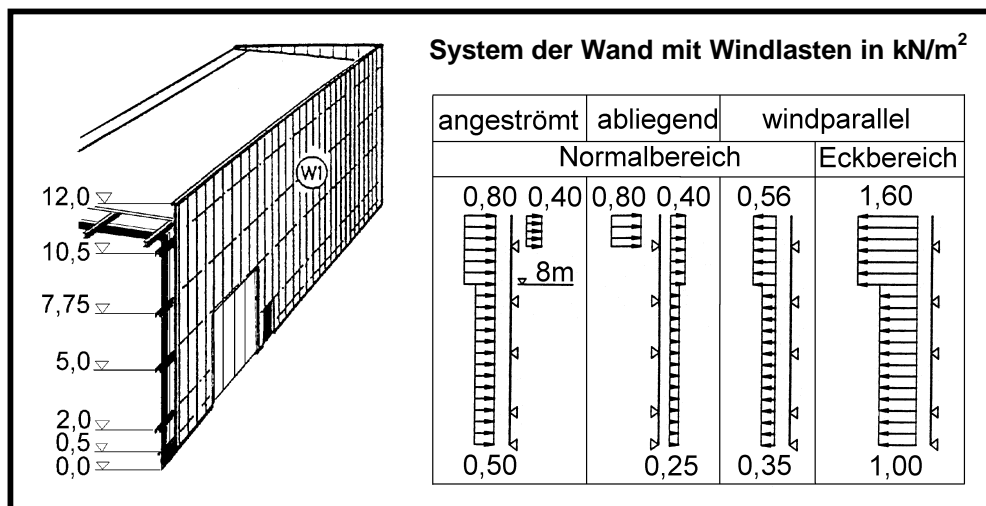


Abbildung 4.21: Wandkonstruktion aus Sandwich-Elementen [77]

Bei einer Kopplung des Stabwerksprogramms an ein CAD-System könnten aus dessen Objektmodell die Querschnittswerte und die Gesamtlänge der Sandwichelemente übernommen werden. Es ist dagegen fast unmöglich, automatisch die Positionen des maßgebenden Schnittes und die der Auflager aus den Wandriegeln abzuleiten. Die Einwirkungen müßten ohnehin aufwendig von Hand eingegeben werden. Die durch eine Systemkopplung erreichte Arbeitersparnis wäre im Vergleich zu den dennoch notwendigen, manuellen Nacharbeiten also sehr gering.

Ähnlich verhält es sich bei den anderen Bauarten für Dach und Wand und im Glasbau. Für die Bauelemente des Stahlbetonbaus stellt sich das Problem im Rahmen dieser Arbeit dagegen erst gar nicht, da deren Bemessung hier nicht behandelt wird. Der Nutzen eines mechanischen Teilmodells innerhalb des Produktmodells für den Komplettbau wäre also sehr gering.

4.5 Fazit

Durch die Anwendungsprozesse in Abschnitt 4.2 wird eine Modellarchitektur mit vier Teilmodellen für das Komplettbaumodell vorgegeben. In einem gewerkeübergreifenden Teilmodell werden als kleinster gemeinsamer Nenner aller Gewerke sämtliche Bauelemente des Komplettbaus in niedrigem Detaillierungsgrad dargestellt. Die gewerkespezifischen Teilmodelle sind somit im gewerkeübergreifenden Modell verankert. Es muß jeweils ein gewerkespezifisches Teilmodell für Dach und Wand, für den Ausbau von Dach- und Wandöffnungen und für den Glasbau geben. Das Modell für Dach- und Wandöffnungen muß eigenständig sein, damit es von allen anderen Teilmodellen genutzt werden kann.

Für den Stahlbetonbau ist kein spezielles Modell notwendig, da Stahlbetonelemente im Produktmodell dieser Arbeit lediglich mit niedrigem Detaillierungsgrad darzustellen sind, der im gewerkeübergreifenden Modell erreicht werden kann. Für das Stahltragwerk, das im Komplettbau eine wichtige Rolle spielt, braucht ebenfalls kein eigenes Modell entwickelt zu werden, da dies bereits im ST-4 Projekt der IAI geschieht [56].

Die Anwendungsprozesse haben außerdem gezeigt, daß der Produktdatenaustausch in mehreren Detaillierungsstufen möglich sein muß. Die niedrigste dieser Stufen, die der Übersichtsplanung entspricht, soll im gewerkeübergreifenden Teilmodell angesiedelt sein. Im Teilmodell für Dach und Wand soll es zwei weitere Stufen für Detail- und Verlegeplanung geben. In den beiden Teilmodellen für Dach- und Wandöffnungen sowie für den Glasbau ist jeweils eine weitere Detaillierungsstufe ausreichend. Der Umfang der Teilmodelle ist durch die technischen Beschreibungen in Abschnitt 4.3 vorgegeben.

Alle angesprochenen Teilmodelle betreffen die Konstruktion der Gewerke. In Abschnitt 4.4 wurde gezeigt, daß ein zusätzliches mechanisches Modell für den Komplettbau nur von geringem Nutzen wäre. Ein entsprechendes Modell wird derzeit ohnehin im ST-4 Projekt entwickelt. Dieses kann genutzt werden, falls später doch Bedarf dafür bestehen sollte. In dieser Arbeit sollen jedoch keine Anstrengungen zur Kopplung von Konstruktionsmodellen und mechanischem Modell unternommen werden.

5 Das Datenmodell der Industry Foundation Classes

5.1 Allgemeines

In Kapitel 3 dieser Arbeit wurden die Grundzüge der IFC2x sowie deren Modellar-chitektur erläutert, und es wurde die Notwendigkeit aufgezeigt, das Komplettbau-modell als Erweiterung der IFC zu definieren. Um zu gewährleisten, daß vorhandene Datenstrukturen der IFC genutzt werden und das Erweiterungsmodell somit optimal im Modellkern verankert ist, empfiehlt die IAI folgendes Vorgehen [60].

- 1) Die für den Komplettbau relevante Untermenge der IFC wird festgelegt. Dazu werden aus den konkreten Klassen des Interoperability Layers zunächst diejen-igen Bauelemente ausgewählt, die für den Komplettbau von Bedeutung sind. Durch Zurückverfolgen der Verweis- und Vererbungshierarchien werden an-schließend alle dazu verwendeten Datenstrukturen im Core und Resource Layer der IFC ausfindig gemacht.
- 2) Bestehende Klassen aus dieser Untermenge werden um Eigenschaften erweitert, die die IFC nicht abdecken.
- 3) Neue Klassen werden für Dateninhalte definiert, die die IFC nicht abdecken.

Aus Platzgründen wird die Beschreibung der relevanten Untermenge in diesem Ka-pitel mit einer Erläuterung der darin enthaltenen Datenstrukturen verbunden. Der Verständlichkeit halber wird dies in umgekehrter Reihenfolge getan als in Schritt 1) vorgeschlagen, also beginnend mit den allgemeingültigen Strukturen. Aus Platz-gründen wird die Herleitung der Untermenge nicht dokumentiert. Sie ist zum Ver-ständnis auch nicht unbedingt notwendig.

Da das Datenmodell der IFC sehr umfangreich und komplex ist, können die Be-schreibungen nicht bis ins kleinste Detail gehen. Für weitergehende Erläuterungen wird auf den *Model Implementation Guide* und die Online-Dokumentation der IFC2x verwiesen [59], [61]. Die Schritte 2) und 3) des oben genannten Vorgehens werden erst in Kapitel 6 durchgeführt werden.

5.2 Vereinbarungen

5.2.1 Terminologie

Zur präzisen Formulierung wird die Terminologie der IFC verwendet, auch wenn dies nicht immer dem allgemeinen Sprachgebrauch oder dem Sprachgebrauch nach ISO 10303 entspricht. Dies betrifft die folgenden Begriffe:

Der Begriff *Klasse* wird anstelle des Begriffs *Entity* aus ISO 10303 verwendet. Sämtliche Klassennamen beginnen mit den Buchstaben „Ifc“, gefolgt von einem aussagekräftigen englischen Namen, zum Beispiel *IfcObject*. Diejenigen Klassen, die am Ende eines Astes in der Vererbungshierarchie stehen, werden dem Bild des Vererbungsbaums entsprechend als *Leaf Node Classes* bezeichnet. Im Gegensatz zu ihren abstrakten Oberklassen werden sie tatsächlich in die Austauschdateien ge-schrieben. Die Datensätze in den Austauschdateien heißen wie in ISO 10303 *Instan-zen*. Der Begriff des *Bauelementes* wurde bereits in Kapitel 4 angesprochen. Er leitet

sich von der Klasse *IfcBuildingElement* ab, durch die Gebäudeteile mit globaler Funktion dargestellt werden. Der Begriff der *Datenstruktur* wird für Konstrukte aus beliebigen Klassen im Datenmodell verwendet.

5.2.2 Top-Down-Ansatz

Die Darstellung von Bauwerken in den IFC geht vom Bauwerk als Ganzem aus, das in seine Bestandteile, zum Beispiel die Bauelemente, zerlegt wird. Die Bestandteile können ihrerseits immer weiter in Komponenten aufgelöst werden. Der hierzu verwendete Mechanismus wird *Auflösung* oder *Dekomposition* genannt. Mit fortschreitender Auflösung nimmt der Detaillierungsgrad, aber auch die Datenmenge der Darstellung zu.

5.2.3 Vererbungsregeln

Die *generischen Datenstrukturen* aus dem Modellkern werden an die Klassen der Domänenmodelle weitervererbt und dort spezialisiert. Da an beliebigen Punkten im Vererbungsbaum angeknüpft werden kann, ist die Vererbung ein wichtiges Prinzip zur Erweiterung der IFC. Die Richtung der Vererbung darf jedoch nicht entgegen der Hierarchie der funktionalen Schichten stattfinden. Die IFC verwenden lediglich einfache Vererbung, das heißt, daß eine Unterklasse nicht mehrere unmittelbare Oberklassen haben darf. Außerdem ist festgelegt, daß in einer Unterklasse keine von den Oberklassen geerbten Attribute umdefiniert werden dürfen.

Mit fortschreitender Vererbung nimmt die Zahl der Attribute und Eigenschaften einer Klasse zu. Dadurch wird die mögliche Informationsdichte und der mögliche Detaillierungsgrad für die Darstellung größer; diese Vielzahl an Möglichkeiten muß für jede spezielle Anwendung jedoch konkretisiert und in der Regel wieder eingeschränkt werden. Dies kann in der Definition der Leaf Node Classes oder in der Definition der Sichten geschehen.

5.2.4 Verweisregeln

Auch die Verweise zwischen den Klassen unterliegen in den IFC bestimmten Regeln, die durch die Hierarchie der funktionalen Schichten begründet sind. Diese Regeln besagen, daß Klassen lediglich von anderen Klassen aus derselben oder aus höheren Schichten referenziert werden dürfen. In Domänen definierte Klassen dürfen nur Klassen in tieferliegenden Schichten referenzieren, und nicht umgekehrt. Dadurch soll einerseits gesichert werden, daß die generischen Strukturen des Modellkerns durchgehend genutzt und somit die Domänenmodelle möglichst stark im Kern der IFC verankert werden. Andererseits bleibt so die Konsistenz des Modellkerns unabhängig von der Entwicklung einzelner Domänen.

5.2.5 Notation

Das Datenmodell der IFC ist gemäß ISO 10303 in der formalen Sprache EXPRESS definiert. Zusätzlich dazu wird es in der Online-Dokumentation vollständig in Form von EXPRESS-G-Diagrammen dargestellt. Zur Verdeutlichung der Zusammenhänge zwischen den Klassen und den verschiedenen Modellteilen wird in dieser Arbeit ebenfalls mit EXPRESS-G gearbeitet. Die wichtigsten Elemente von EXPRESS-G gemäß ISO 10303 sind deshalb in den folgenden Punkten erläutert und in Abbildung 5.1 dargestellt.

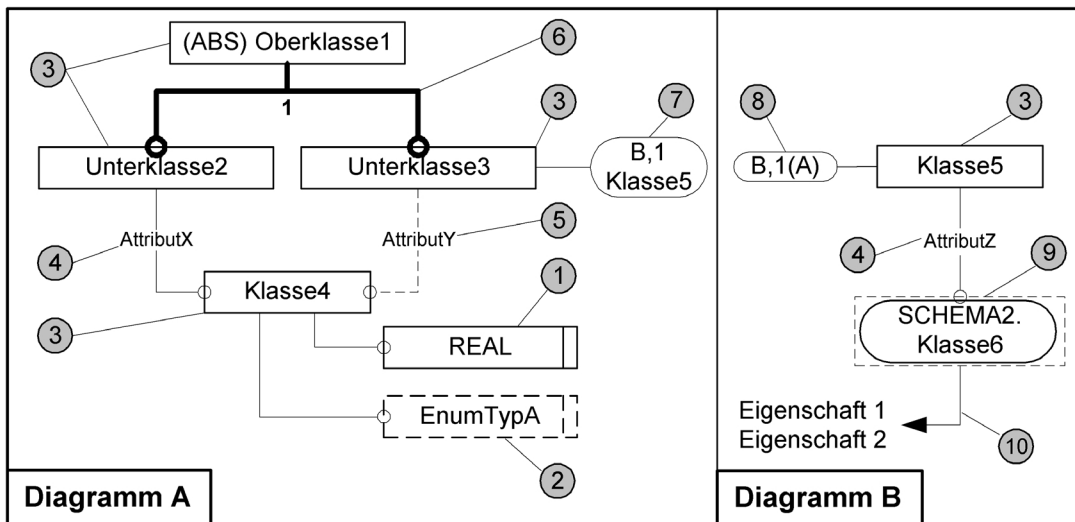


Abbildung 5.1: Elemente von EXPRESS-G

- 1) Ein eckiges Kästchen mit doppelter Linie am rechten Rand definiert einen einfachen Datentyp (BOOLEAN, INTEGER, REAL oder STRING).
- 2) Ein eckiges, gestricheltes Kästchen mit doppelter Linie am rechten Rand definiert einen Aufzählungstyp. Innerhalb des Kästchens steht der Name des Typs.
- 3) Ein eckiges, mit durchgehenden Linien gerahmtes Kästchen definiert eine Klasse. Der Klassenname steht im Inneren des Kästchens. Sofern es sich um eine abstrakte Oberklasse handelt, ist dem Namen das Kürzel „(ABS)“ vorangestellt.
- 4) Eine dünne, durchgehende Linie definiert einen Verweis zwischen Klassen durch ein explizites Attribut. Der Kreis steht auf der Seite der referenzierten Klasse; der Name des Attributes steht an der Linie.
- 5) Eine dünne, gestrichelte Linie definiert wie Punkt 4) einen Verweis. Er ist jedoch optional.
- 6) Eine dicke, durchgehende Linie definiert eine Vererbung zwischen Klassen. Der Kreis steht auf der Seite der Unterklasse. Die Ziffer 1 an einer Verzweigung von Vererbungslinien bedeutet, daß es sich um einfache Vererbung handelt.
- 7) Ein abgerundetes Kästchen mit zwei, durch ein Komma getrennte Referenzen und einem Klassennamen im Inneren definiert einen Verweis oder eine Vererbung in Richtung einer Klasse, die sich in einem anderen EXPRESS-G-Diagramm befindet. Die erste Referenz bezeichnet das referenzierte Diagramm, die zweite den Anknüpfungspunkt innerhalb des referenzierten Diagramms.
- 8) Ein abgerundetes Kästchen mit zwei, durch ein Komma getrennte Referenzen und einer dritten Referenz in Klammern definiert einen Anknüpfungspunkt für einen Verweis oder eine Vererbung aus einem anderen Diagramm. Die ersten beiden Referenzen sind identisch mit denen des Anknüpfungspunktes, von dem aus verwiesen wird (siehe Punkt 7)). Die Referenz in Klammern ist die des Diagramms, von dem aus verwiesen wird.
- 9) Ein eckiges, gestricheltes Kästchen mit einem abgerundeten Kästchen im Inneren definiert eine Klasse, auf die von einem anderen EXPRESS-Schema aus verwiesen oder „vererbt“ wird. Durch einen Punkt getrennt werden darin sowohl der Schema- als auch der Klassenname angegeben.
- 10) Teilweise sind die Klassen in den Diagrammen mit allgemeinverständlichen Kommentaren versehen, auf die mit Pfeilen verwiesen wird. Dies entspricht nicht ISO 10303. Die Verfechter einer strengen, formalen Linie mögen dies jedoch durch den Gewinn an Anschaulichkeit und Kompaktheit entschuldigen.

Die Mechanismen 7) bis 9) sind für die IFC besonders wichtig, da deren komplexes Datenmodell aus einer Vielzahl von EXPRESS-G-Diagrammen besteht und mehrere EXPRESS-Schemata umfaßt. Die in diesem und im nächsten Kapitel aufgeführten EXPRESS-G-Diagramme sind deshalb durchlaufend nummeriert und verweisen aufeinander. Die Diagramme in dieser Arbeit erheben jedoch nicht den Anspruch auf Vollständigkeit und sind deshalb zum Teil stark vereinfacht.

5.3 Ressourcen

5.3.1 Allgemeines

Ressourcen sind in der untersten funktionalen Schicht der IFC definiert, dem *Resource Layer*. Eine Ressource definiert eine allgemeingültige und grundlegende Datenstruktur, die gemäß der oben zitierten Verweisregeln von anderen Strukturen in den höheren funktionalen Schichten benutzt werden kann. Die Nutzung gemeinsamer Ressourcen sichert die niedrigste Stufe der Kompatibilität innerhalb des Datenmodells. Ein einfaches Beispiel hierfür ist die Benutzung eines einheitlichen Konzeptes für Maßeinheiten.

Für jede Ressource existiert ein eigenes Ressource-Schema. Im folgenden werden lediglich die für den Komplettbau relevanten der insgesamt zwanzig IFC-Ressourcen beschrieben.

5.3.2 Maßeinheiten

Das Konzept für Maßeinheiten ist im Schema *IFCMEASURERESOURCE* definiert. Der Grundgedanke dabei ist, daß Attribute in den Klassen der IFC nicht direkt als einfache EXPRESS-Datentypen zu definieren sind, sondern auf die Datentypen dieser Ressource zu verweisen haben. Ein Attribut, das eine Länge definiert, verweist beispielsweise auf den Typ *IfcLengthMeasure*. Anstatt die dafür zu verwendende Maßeinheit in der entsprechenden Klasse festzulegen, wird dies einheitlich für die Längenangaben aller Klassen getan, jedoch individuell für jede Austauschdatei. Dadurch wird die projektspezifische Festlegung von Maßeinheiten möglich, was eine Voraussetzung für den Einsatz der IFC in unterschiedlichen nationalen Maßeinheitskonzepten ist.

5.3.3 Geometrie

Die IFC2x bieten sehr umfangreiche und hochentwickelte Sprachmittel zur Darstellung von geometrischen Elementen und deren Einfügung im Raum. Außer den üblichen geometrischen Grundelementen (explizit und parametrisch) stehen räumliche Oberflächen- und Volumenkörpergeometrien zur Verfügung. Letztere können auf drei verschiedene Arten dargestellt werden: mit der Klasse *IfcSweptAreaSolid* durch Extrusion ebener, geschlossener Profile (Extrusionskörper), mit *IfcManifoldSolidBrep* über seine Begrenzungsflächen (BREP-Geometrie) oder mit *IfcCsgSolid* durch boole'sche Operationen (CSG-Geometrie).

Die entsprechenden Klassen sind in fünf Ressource-Schemata definiert, deren Erläuterung an dieser Stelle zu weit führen würde. Im Bezug auf Objekte, die eine Lage im Raum oder eine geometrische Form besitzen, wird deshalb im folgenden einfach von „Einfügekordinatensystem“ und „geometrischer Darstellung“ gesprochen.

Für die Anwendung dieser Geometrien gelten in den IFC zwei Besonderheiten: Erstens darf ein Objekt im allgemeinen mehr als nur eine einzige geometrische Dar-

stellung besitzen, also beispielsweise eine vereinfachte und eine detaillierte. Zweitens soll eine geometrische Darstellung immer sich selbst genügen. Das heißt einerseits, daß sie ausschließlich geometrische Informationen beinhalten darf, und andererseits, daß sie immer vollständig sein muß, daß also zum Beispiel die ausführliche Profildarstellung nicht durch eine standardisierte Bezeichnung ersetzt werden kann. Letzteres hat zwar den Nachteil, daß die Austauschdateien sehr groß werden, gewährleistet aber, daß beim Datenaustausch auch zwischen sehr unterschiedlichen Software-Applikationen zumindest die Geometrie als kleinster gemeinsamer Nenner übertragen werden kann.

5.3.4 Materialdefinition

Die Definition von Materialien in den IFC erfolgt im Schema *IFCMATERIALRESOURCE*, siehe Abbildung 5.2. Im Standardfall wird ein einzelnes Material in der Klasse *IfcMaterial* nur durch eine Bezeichnung im Attribut *Name* beschrieben. Die Klasse *IfcMaterialLayerSet* definiert geschichtete Materialien, indem sie mehrere Schichten (*IfcMaterialLayer*) zusammenfaßt. Jede Schicht enthält außer der erwähnten Bezeichnung eine Dicke und die Information, ob sie hinterlüftet ist oder nicht. Die Dicke der gesamten geschichteten Materialdefinition ergibt sich aus der Summe der Schichtdicken. Auf diese Art können Schichtaufbauten von Dächern oder Wänden definiert werden, einschließlich vorgehängter, hinterlüfteter Fassaden. *IfcMaterialList* definiert dagegen eine nicht weiter spezifizierte Zusammenstellung von Materialien, zum Beispiel Beton, Baustahl und Bewehrungsstahl für ein Stahlbetonteil.

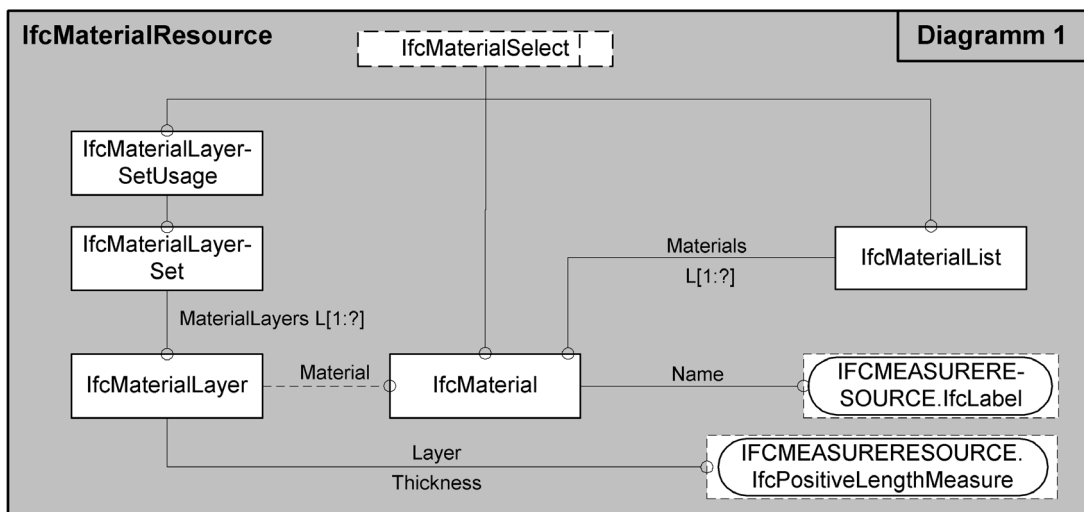


Abbildung 5.2: Material-Ressource (EXPRESS-G)

5.4 Generische Datenstrukturen

5.4.1 Allgemeines

Der Modellkern der IFC ist im Schema *IFCKERNEL* definiert, das im Core Layer angesiedelt ist. In ihm sind alle Datenstrukturen der IFC, außer denen der Ressourcen, verwurzelt. Da diese anwendungsunabhängig sind, wird von *generischen* Strukturen gesprochen. Dies wirkt sich unter anderem darin aus, daß fast alle der hier definierten Klassen abstrakt sind.

Abbildung 5.3 zeigt die drei generischen Schlüsselstrukturen der IFC: das *Objekt*, die *Relation* und die *Eigenschaftsdefinition*. Sie alle leiten sich durch Vererbung von der gemeinsamen Oberklasse *IfcRoot* ab, die unter anderem die Attribute *GlobalId* für die *Objektidentität* und *OwnerHistory* für Besitzstatus und -geschichte beisteuert. Unter Objektidentität versteht man eine durch die Anwendungs-Software vergebene, weltweit einzigartige Zeichenkette, anhand derer ein Objekt jederzeit wiedererkannt werden kann. Der Besitzstatus (Klasse *IfcOwnerHistory*) eines Objektes gibt dessen Besitz- und Benutzerinformationen an. Beide Attribute sind nicht optional, so daß alle im weiteren Verlauf dieses Kapitels aufgeführten Klassen der IFC diese Informationen besitzen.

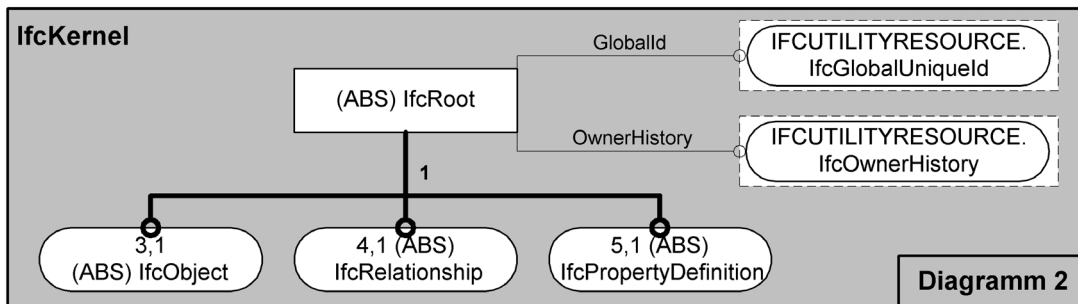


Abbildung 5.3: Schlüsselstrukturen (EXPRESS-G)

5.4.2 Objekte

Abbildung 5.4 zeigt die Klassenstruktur der für diese Arbeit relevanten Objekte in den Schemata *IFCKERNEL* und *IFCPRODUCTEXTENSION*, die sich beide im Core Layer der IFC befinden. Der Begriff des Objektes wird in der weiteren Arbeit ausschließlich im Sinne von *IfcObject* verwendet.

IfcObject:

Objekte sind nach Definition der IFC Prozesse, Kontrollmechanismen, Ressourcen im Sinne von Ausrüstung und Baumaterial, Personen, Projekte, greifbare Objekte und Gruppierungsmechanismen. *IfcObject* ist die abstrakte Oberklasse für alle diese Objekte, die sich durch verschiedene, grundlegende Funktionen unterscheiden. *IfcObject* selbst verfügt über Möglichkeiten zur Materialzuordnung, zur Dekomposition und zur Einfügung von Typdefinitionen (der Typmechanismus wird erst in Abschnitt 5.4.4 erklärt).

IfcProject:

Die Klasse *IfcProject* dient zur Darstellung eines Projektes, das in einer IFC-Austauschdatei genau einmal vorkommt und den äußeren Rahmen für alle übrigen Daten bildet.

IfcGroup:

Die Klasse *IfcGroup* faßt beliebige Objekte zu Gruppen zusammen. Diese Gruppierungen haben keine geometrischen oder topologischen Bedeutungen. Gruppen können beliebig verschachtelt sein.

IfcProduct:

Abstrakte Oberklasse aller greifbaren Objekte. *IfcProduct* verfügt über ein Einfügekoordinatensystem und kann eine oder mehrere geometrische Darstellungen besitzen.

Ein Sonderfall ist die Unterklasse *IfcProxy*, die im Gegensatz zu *IfcProduct* keine geometrische Darstellung hat. Dies kann zum Beispiel sinnvoll sein, um ein Werknormteil nur durch eine standardisierte Bezeichnung darzustellen, also ohne Geometrie. Dies ist kein Widerspruch zu der in Abschnitt 5.3.3 genannten Regel, daß eine Geometrie immer vollständig sein muß, da hier gar keine existiert.

IfcElement:

Abstrakte Oberklasse aller Komponenten eines Bauwerks. Dieses können neben physikalisch existierenden Bau- und Ausrüstungsteilen auch Löcher oder Aussparungen sein. Ein Element kann mit anderen Elementen verbunden werden.

IfcSpatialStructureElement:

Abstrakte Oberklasse zur Darstellung aller nichtgegenständlichen Elemente einer räumlichen Struktur. Dieses sind Gebäude, Geschosse, Räume und Bauplätze.

IfcBuildingElement:

Abstrakte Oberklasse aller Bauelemente. Bauelemente können über raumbildende Eigenschaften verfügen. Das heißt, daß beispielsweise Wände und Decken einen Raum einschließen und dadurch dessen Ausdehnung definieren. Außerdem kann ein Bauelement Öffnungen (*IfcOpeningElement*) besitzen; umgekehrt kann ihm explizit die Eigenschaft zugewiesen werden, eine Öffnungen auszufüllen.

IfcOpeningElement:

Die Klasse *IfcOpeningElement* stellt Öffnungen in Bauelementen dar. Für eine Software-Applikation wird dadurch eine CSG-Operation impliziert, bei der die Geometrie der Öffnung von der des Bauelementes abgezogen wird.

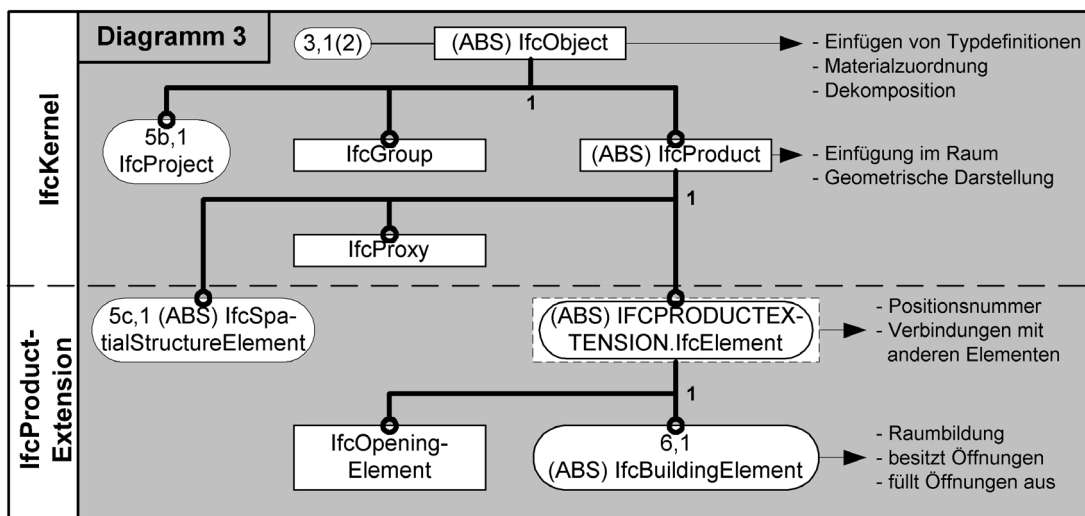


Abbildung 5.4: Objekte (EXPRESS-G)

5.4.3 Relationen

Die zweite Schlüsselstruktur der IFC ist die der Relation. Sie beruht darauf, daß Beziehungen zwischen Objekten anstatt mit einfachen Verweisen durch eigene Klassen dargestellt werden. In den IFC wird deshalb von *Objectified Relationships* gesprochen. Der Sinn dieses Konzeptes ist es, die Semantik der Relationen von den Attributen der Objekte zu trennen. Abbildung 5.5 zeigt die Klassenstruktur der für diese Arbeit wichtigen Relationen, die in den Schemata *IFCKERNEL* und *IFCPRODUCTEXTENSION* definiert sind.

IfcRelationship:

Dies ist die abstrakte Oberklasse für alle Arten von Relationen.

IfcRelAssociates:

Abstrakte Oberklasse zur Definition von Referenzen zu Informationsquellen außerhalb des IFC-Modells. Diese Informationsquellen können Klassifizierungen (*IfcRelAssociatesClassification*), Dokumente (*IfcRelAssociatesDocument*) oder Bibliotheken (*IfcRelAssociatesLibrary*) sein. Ein Beispiel ist die Referenz eines Objektes auf eine Norm, die mit *Ifc-RelAssociatesDocument* dargestellt werden kann.

Eine weitere wichtige Unterklasse ist *IfcRelAssociatesMaterial*, mit der einem Objekt einzelne Materialien, geschichtete Materialdefinitionen oder Materiallisten zugeordnet werden können.

IfcRelAssigns:

Abstrakte Oberklasse zur Zuordnung eines Objektes oder einer Gruppe von Objekten zu einem anderen Objekt. Die Unterklasse *IfcRelAssignsToGroup* definiert zum Beispiel die Zuordnung einer beliebigen Zahl von Objekten zu einer Gruppe (*Ifc-Group*).

IfcRelConnects:

Abstrakte Oberklasse zur Darstellung einer Verbindung zwischen mehreren Objekten. Eine wichtige Unterklasse ist *IfcRelConnectsElements*, mit der Verbindungen zwischen jeweils genau zwei Elementen (*IfcElement*) definiert werden. Ein Beispiel hierfür ist die Verbindung zweier Wände. Diese Verbindung kann eine logische sein, das heißt, daß die Geometrie der Verbindung sich aus den Geometrien der Wände ergibt, oder die Verbindungsgeometrie ist explizit angegeben, zum Beispiel als Berührungsfläche der Wände.

Weitere wichtige Unterklassen sind *IfcRelVoidsElement*, mit der eine Öffnung (*Ifc-OpeningElement*) in einem Element definiert wird, und *IfcRelFillsElement*, mit der ein Element in eine Öffnung eingesetzt wird.

IfcRelDecomposes:

Wie bereits erwähnt, ist die Dekomposition ein grundlegendes Prinzip der IFC, das durch die abstrakte Oberklasse *IfcRelDecomposes* definiert ist. Damit kann ein Objekt in beliebig viele andere Objekte aufgelöst werden. Diese Objekte können durch Verschachtelung mehrerer Dekompositionen wiederum selbst aufgelöst werden. Für den Komplettbau ist nur die Unterklasse *IfcRelAggregates* von Bedeutung, die unterschiedliche Klassen als Komponenten der Dekomposition haben darf. Ein Beispiel für eine Aggregation ist ein Dach, das aus Dachflächen, Sparren und Pfetten besteht. Das aufgelöste Dach darf keine eigene geometrische Darstellung haben. Diese ergibt sich vielmehr aus der Summe der Geometrien seiner Komponenten.

IfcRelDefines:

Die abstrakte Oberklasse *IfcRelDefines* hat zwei Arten von Relationen als Unterklassen: Durch *IfcRelDefinesByProperties* werden einem Objekt Sätze von Eigenschaften (*IfcPropertySetDefinition*) zugeordnet, die das Objekt auf semantische Art und Weise beschreiben. Durch *IfcRelDefinesByType* wird dem Objekt dagegen eine sogenannte Typdefinition (*IfcTypeObject*) zugeordnet, die mehrere Eigenschaftssätze zusammenfaßt. Eine einzige Typdefinition kann beliebig vielen Objekten zugeordnet und dadurch mehrfach eingefügt werden.

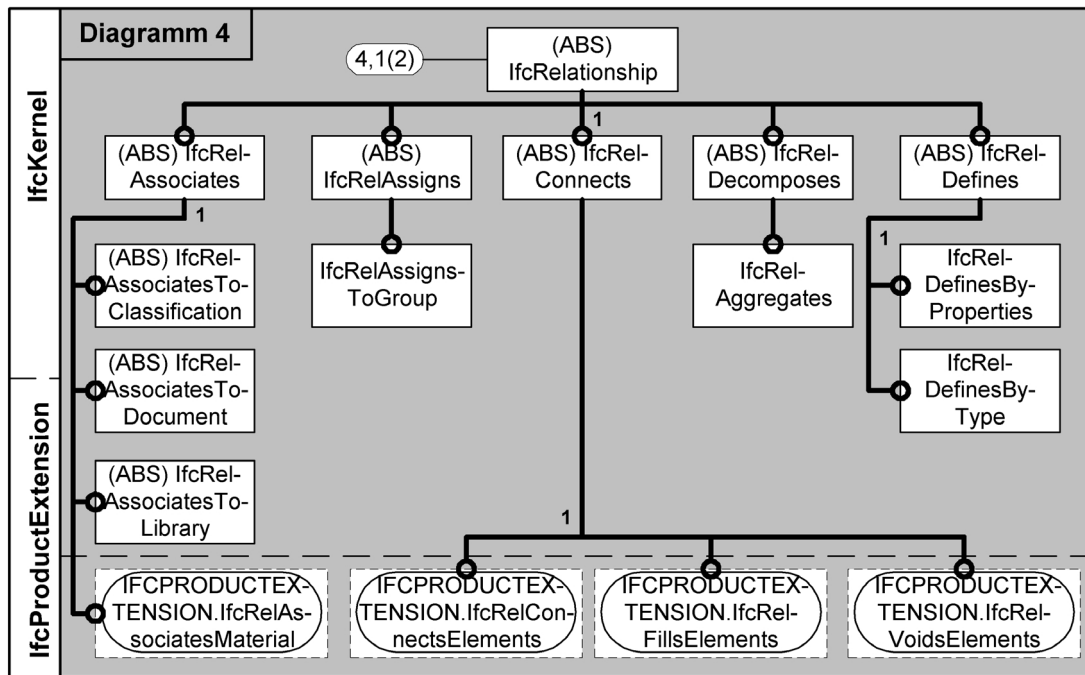


Abbildung 5.5: Relationen (EXPRESS-G)

5.4.4 Eigenschaftsdefinitionen

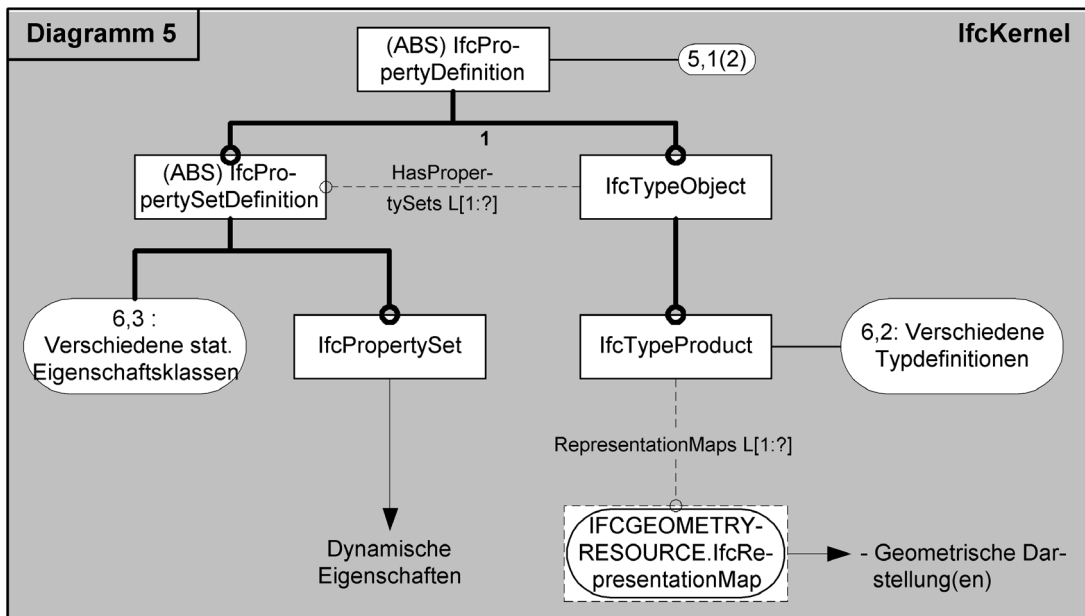


Abbildung 5.6: Eigenschaftsdefinitionen (EXPRESS-G)

Die dritte Schlüsselstruktur der IFC ist die der Beschreibung von Objekten durch spezielle Eigenschaftsdefinitionen. Diese Eigenschaften haben semantischen Charakter - im Gegensatz zu den in Abschnitt 5.4.2 beschriebenen, durch Vererbung weitergegebenen „Grundfunktionen“ der Objekte. Die semantischen Eigenschaften werden nicht direkt als Attribute der zu beschreibenden Objekte definiert, sondern in eigenständigen Klassen. Einem Objekt können durch die bereits erläuterte Relation *IfcRelDefines* eine beliebige Anzahl von Eigenschaftsdefinitionen zugeordnet werden. Dieses Konzept ist für Erweiterungsmodelle der IFC besonders interessant, da

jeder bestehenden, allgemeinverständlichen Klasse weitere Informationen einfach angehängt werden können, ohne die Klasse zu verändern. Abbildung 5.6 zeigt die in dieser Arbeit benötigten Klassen für Eigenschaftsdefinitionen im Schema *IFCKER-NEL*.

IfcPropertyDefinition:

Abstrakte Oberklasse für alle Eigenschaftsdefinitionen.

IfcPropertySetDefinition:

Abstrakte Oberklasse zur Definition von Sätzen aus Eigenschaften, von denen zwei verschiedene Arten existieren.

- a) Statische Eigenschaften: Statische Eigenschaften sind innerhalb des Datenmodells definiert. Sie sind, wie alle anderen Klassen, in einem EXPRESS-Schema der IFC spezifiziert, und zwar als unmittelbare Unterklassen von *IfcPropertySetDefinition*. Die Eigenschaften dieser Klassen werden durch explizite Attribute ausgedrückt, deren semantische Bedeutungen innerhalb der Klassendefinition erklärt sind. Der Begriff „statisch“ bezieht sich darauf, daß diese Bedeutungen somit unveränderlich sind; er bezieht sich nicht etwa auf statische Berechnungen.

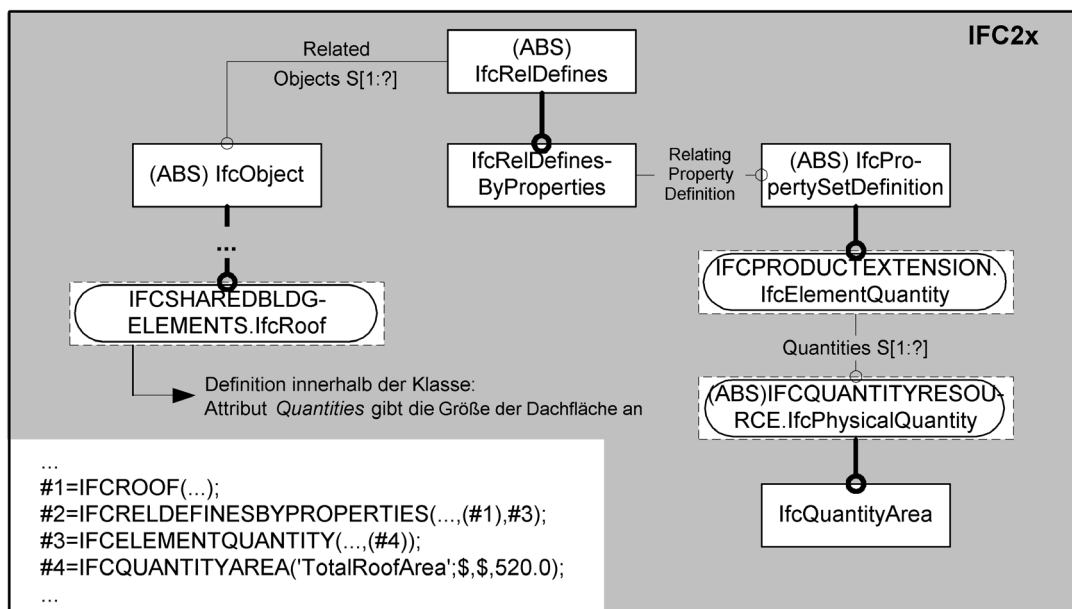


Abbildung 5.7: Beispiel für eine statische Eigenschaft

Abbildung 5.7 zeigt die vereinfachte Klassenstruktur sowie als Anschauungsbeispiel ein Fragment einer Austauschdatei für ein Bauelement „Dach“. In der Klassendefinition für das Dach (*IfcRoof*) wird festgelegt, daß der Zahlenwert für die Größe der Dachfläche im Attribut *Quantities* der statischen Eigenschaftsklasse *IfcElementQuantity* anzugeben ist. Die Zuordnung der Eigenschaft zum Bauelement erfolgt durch *IfcRelDefinesByProperties*.

- b) Dynamische Eigenschaften: Dynamische Eigenschaften sind dagegen außerhalb des Datenmodells definiert. „Dynamisch“ bedeutet, daß das Modell immer bei Bedarf durch zusätzliche Vereinbarungen ergänzt wird. Die IFC2x stellen dazu die Klasse *IfcPropertySet* zur Verfügung, die den syntaktischen Rahmen für solche Erweiterungen im EXPRESS-Schema festlegt. Somit muß lediglich die semantische Bedeutung der Werte durch zusätzliche Vereinbarungen festgelegt werden. Dazu werden sogenannte *PSETs* in der eigens dazu entwickelten, XML-

basierten Sprache PSD definiert [81]. Der Anschaulichkeit halber werden diese PSETs in dieser Arbeit jedoch als Tabellen dargestellt, so wie PSD auch von Standard-Internet-Browsern angezeigt wird.

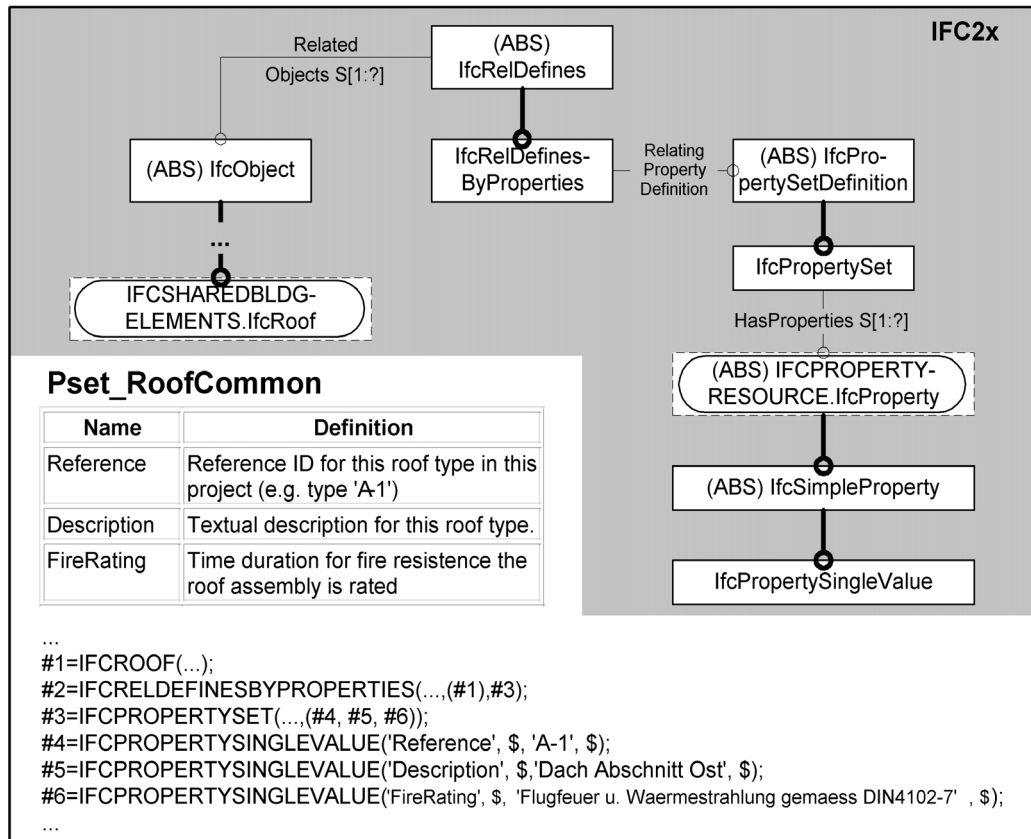


Abbildung 5.8: Beispiel für dynamische Eigenschaften

Abbildung 5.8 zeigt das Beispiel der dynamischen Modellerweiterung um die Eigenschaften „Referenz, Beschreibung und Feuerwiderstand“ für das Bauelement „Dach“. Die Eigenschaften werden jeweils in einer Instanz der Klasse *IfcPropertySingleValue* angegeben, die durch eine Instanz *PropertySet* zusammengefaßt und dem Dach über *IfcRelDefinesByProperties* zugeordnet werden. Die Bedeutung dieser Werte ist in der abgebildeten Tabelle *Pset_RoofCommon* festgelegt.

Im allgemeinen sind statisch definierte Eigenschaften zu bevorzugen, da in diesem Fall alle Definitionen durch das Produktmodell ja allgemein anerkannt und in formaler Art und Weise definiert sind. Die dynamische Variante ist dagegen dann von Vorteil, wenn Eigenschaften nur für einen begrenzten Kreis von Beteiligten von Bedeutung sind, wenn Sachverhalte gar nicht einheitlich darstellbar sind (zum Beispiel bedingt durch unterschiedliche Normen) oder für Erweiterungen, die im Hinblick auf ihre spätere statische Integration in das Modell zunächst einmal dynamisch durchgeführt werden.

IfcTypeObject:

Mit der Klasse *IfcTypeObject* werden sogenannte *Typdefinitionen* dargestellt. Eine Typdefinition kann beliebige semantische Eigenschaften besitzen. Eine einzige Instanz dieser Klasse kann von mehreren Objektinstanzen eingefügt werden, die dann *Typeeinfügungen* heißen.

Von besonderer Bedeutung ist die Unterklasse *IfcTypeProduct*, die außer semantischen Eigenschaften eine oder mehrere vollständige Geometrien besitzen kann. Diese Geometrien sind in der Klasse *IfcRepresentationMap* gekapselt. Die Einfügung erfolgt über die Klasse *IfcMappedItem*, die einen sogenannten Transformationsoperator besitzt: Damit kann die eingefügte Geometrie im Koordinatensystem der Blockeinfügung noch verschoben, gedreht und bezüglich aller drei Koordinatenachsen einheitlich skaliert werden. Dieser Sonderfall der Typdefinition wird auch als *Blockdefinition* bezeichnet. Sie kann von jeder Unterklasse von *IfcProduct* eingefügt werden, die ein Einfügekoordinatensystem besitzt. Die einfügende Klasse wird entsprechend als *Blockeinfügung* bezeichnet. Abbildung 5.9 zeigt die Klassenstruktur für diesen Blockmechanismus am Beispiel eines Bauelementes „Tür“, bestehend aus Blockeinfügung (*IfcDoor*) und Blockdefinition (*IfcDoorStyle*). Eine einzige Blockdefinition, die Form und Eigenschaften der Tür beinhaltet, kann von beliebig vielen Instanzen der Klasse *IfcDoor* beliebig oft im Bauwerk angeordnet werden.

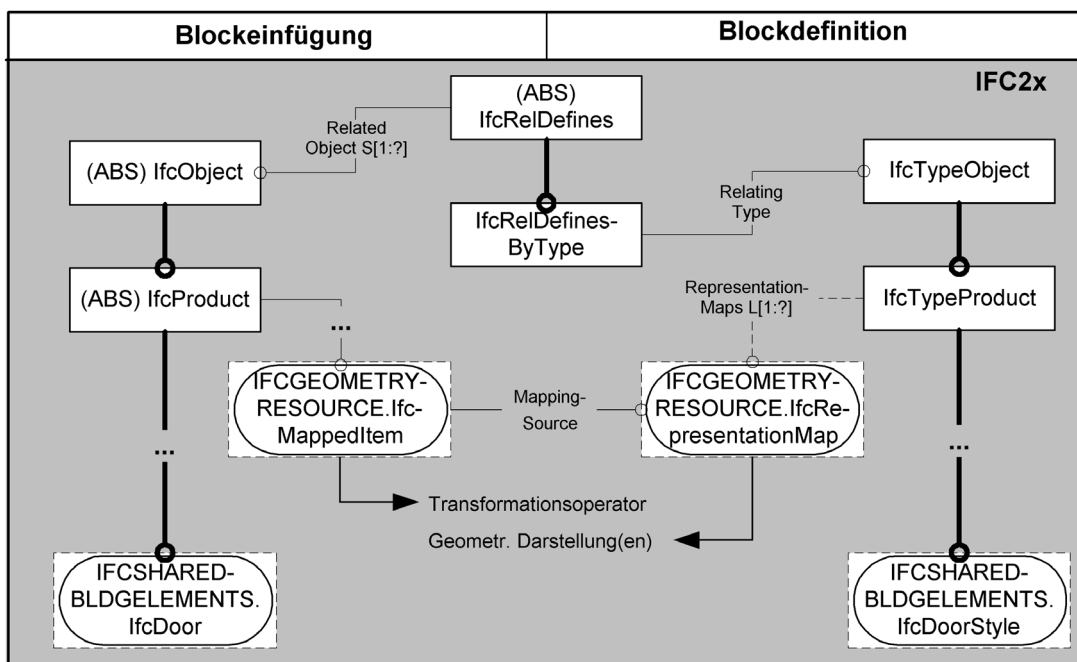


Abbildung 5.9: Blockmechanismus (EXPRESS-G)

Der Blockmechanismus ist also bei vielfach vorkommenden, identischen Bauteilen sinnvoll. Der Nachteil dieser Struktur ist ihre Komplexität, die sich aus der vereinfachten Darstellung in Abbildung 5.9 lediglich erahnen läßt.

5.5 Konkrete Datenstrukturen

5.5.1 Allgemeines

In diesem Abschnitt werden die *Leaf Node Classes* der IFC2x betrachtet, sofern sie für die weitere Arbeit von Bedeutung sind. Diese Klassen sind nicht abstrakt, werden also in den Austauschdateien instanziiert. Aus diesem Grunde wird von *konkreten Datenstrukturen* gesprochen.

5.5.2 Projektstruktur

Die größte Einheit innerhalb einer IFC-Austauschdatei ist das Projekt, das pro Datei genau einmal vorkommt. Dieser inhaltliche Rahmen für alle weiteren Daten der Datei wird im folgenden als *Projektstruktur* bezeichnet. Abbildung 5.10 zeigt die dazu verwendeten Klassen.

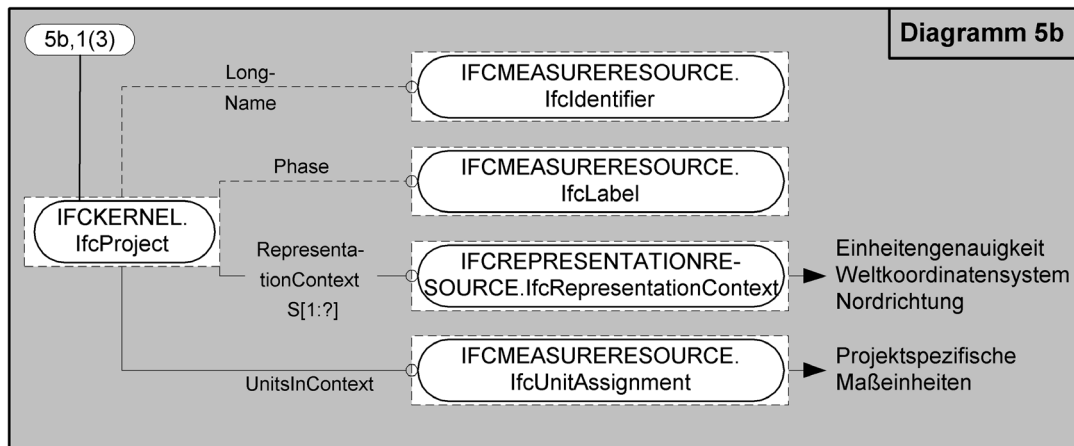


Abbildung 5.10: Klassen der Projektstruktur (EXPRESS-G)

Ein Projekt wird durch die Klasse *IfcProject* dargestellt. Es verfügt neben Attributen zur Angabe des Projektnamens und der aktuellen Projektphase über einen Verweis auf mindestens eine Instanz der Klasse *IfcRepresentationContext*, die die Randbedingungen für geometrische Darstellungen festlegt. In der außerdem referenzierten Instanz der Klasse *IfcUnitAssignment* werden die projektspezifischen Maßeinheiten festgelegt.

5.5.3 Bauwerkstruktur

Gemäß des Top-Down-Ansatzes der IFC wird ein Bauwerk als Ganzes beschrieben und immer weiter in seine Bestandteile aufgelöst. Abbildung 5.11 zeigt die Klassen zur Darstellung derjenigen Bestandteile, die nichtgegenständlich sind: *IfcSite* (Bauplatz), *IfcBuilding* (Gebäude oder Gebäudeabschnitt), *IfcBuildingStorey* (Stockwerk) und *IfcSpace* (Raum). Sie sind im Schema *IFCPRODUCTEXTENSION* der IFC definiert.

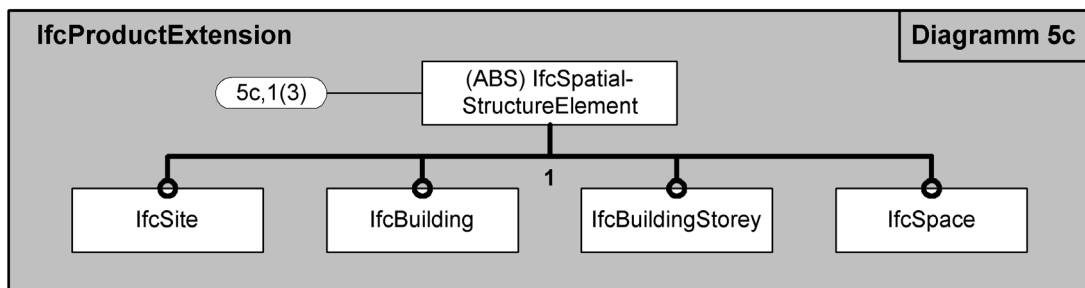


Abbildung 5.11: Nichtgegenständliche Bestandteile des Bauwerks (EXPRESS-G)

Über die Relation *IfcRelAggregates* können diese Bestandteile beliebig aufgelöst werden. Abbildung 5.12 zeigt eine mögliche *Bauwerksstruktur*, die schließlich auch gegenständliche Bauelemente beinhaltet. Sofern diese über geometrische Darstellungen verfügen, müssen sie auf entsprechende Instanzen von *IfcRepresentation*

Context innerhalb der Projektstruktur verweisen. Die dicken Pfeile symbolisieren die Auflösungen, die dünnen Pfeile sind Verweise.

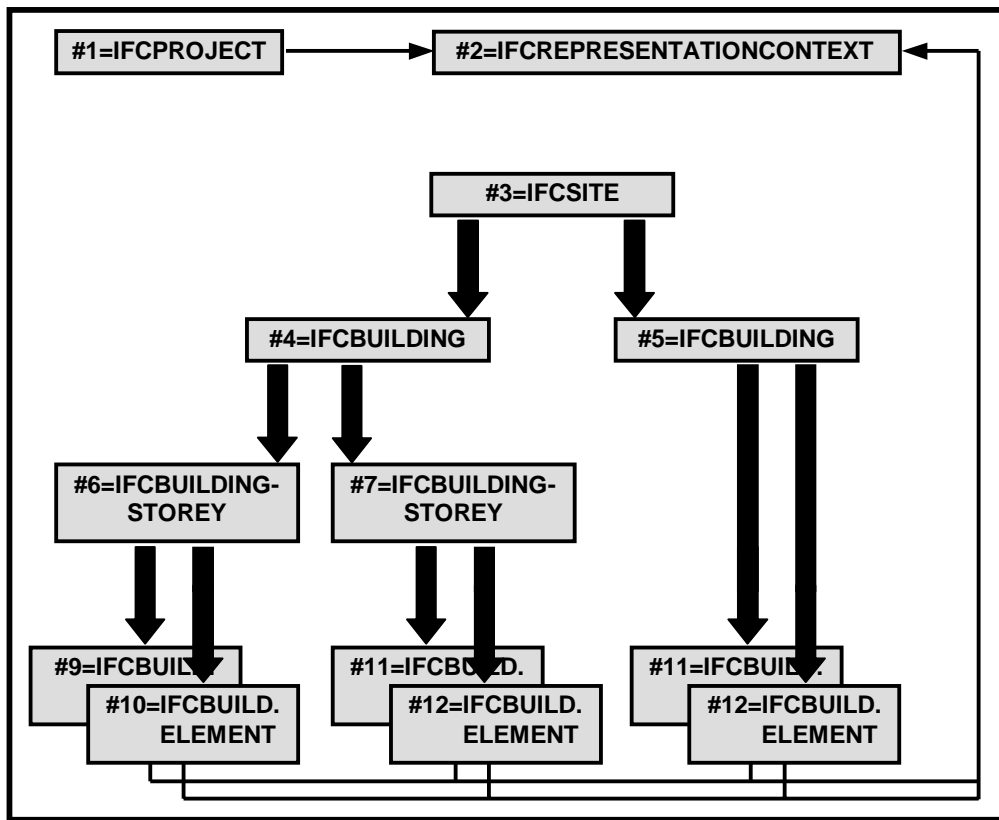


Abbildung 5. 12: Beispiel für eine Bauwerkstruktur in den IFC2x

5.5.4 Bauelemente der IFC2x

Von den gegenständlichen Bestandteilen eines Bauwerkes sind für die konstruktiven Belange des Komplettbaus vor allem die Bauelemente der IFC von Bedeutung. Sie sind als Unterklassen von *IfcBuildingElement* im Schema *IFCSHAREDBLDGELEMENTS* des Interoperability Layers definiert und sind damit domänenübergreifend. Abbildung 5.13 zeigt die Vererbungsstruktur dieser Klassen. Der Übersichtlichkeit wegen sind die Relationen zwischen den Klassen vereinfacht (und formal nicht korrekt) als Pfeile dargestellt.

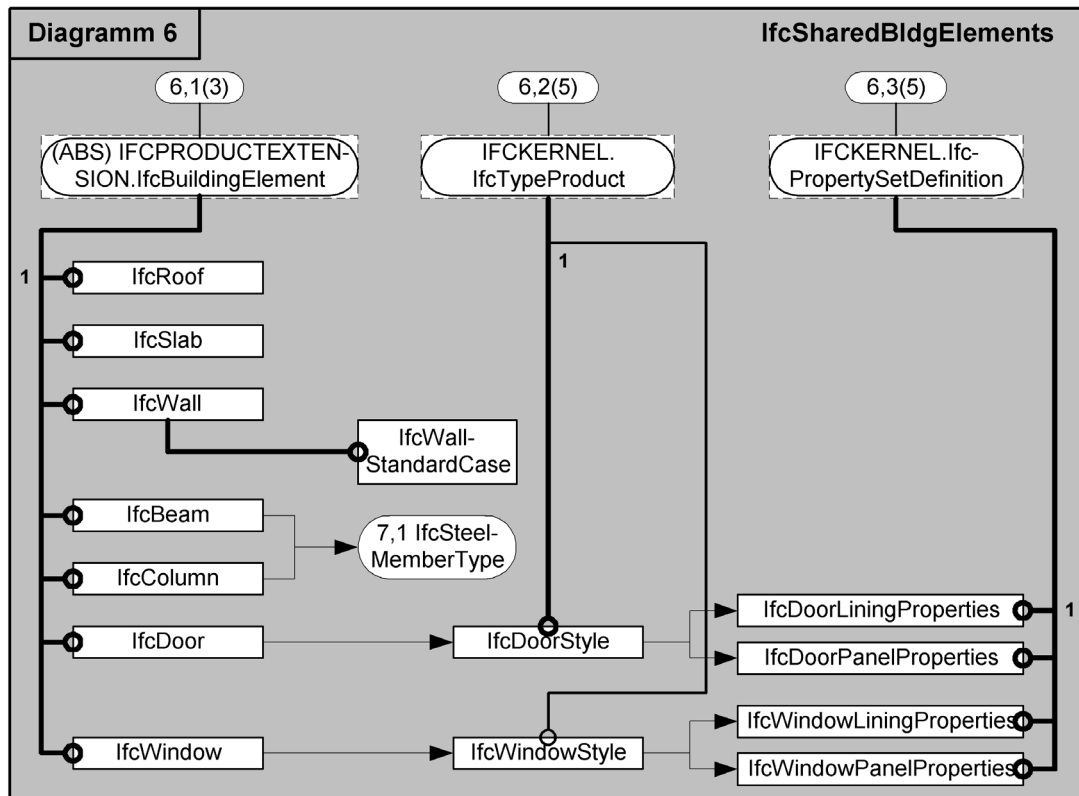


Abbildung 5.13: Klassenstruktur der Bauelemente (EXPRESS-G)

In Tabelle 5.1 sind die Funktionen und Eigenschaften dieser Klassen zusammengestellt. Im Block „Bauelemente“ der Tabelle werden der Klassenname und die Funktion im Bauwerk angegeben. Im Block „Geerbte Datenstrukturen“ wird erläutert, auf welche Art die vom Obertyp *IfcBuildingElement* geerbten generischen Datenstrukturen konkret genutzt werden. Ein Strich bedeutet dabei, daß die Klasse die entsprechende Funktion nicht unterstützt. Die Spalte „Weitere Eigenschaften“ gibt an, ob zusätzlich semantische Eigenschaften definiert werden können, was entweder in expliziten Attributen, in statischen Eigenschaftsklassen oder mit dynamischen Erweiterungen (PSETs) geschieht. Für die Erläuterung der dort angegebenen Typen, Klassen und PSETs wird auf die Online-Dokumentation der IFC2x verwiesen. Im Block „Detaillierung“ wird schließlich der mit diesen Mitteln erreichbare Detaillierungsgrad beschrieben.

Für die Klasse *IfcRoof* bedeuten die Angaben beispielsweise, daß sie entweder über eine geometrische Darstellung vom Typ *Boundary Representation* (BREP) verfügt oder in Komponenten der Klassen *IfcRoof*, *IfcSlab* und *IfcBeam* aufgelöst werden kann. Außerdem kann sie die Möglichkeiten der Material-Ressource uneingeschränkt nutzen und Öffnungen aufweisen. Der Blockmechanismus wird dagegen nicht unterstützt. Zur Definition weiterer Eigenschaften stehen der Aufzählungstyp *IfcRoofTypeEnum* (architektonischer Typ des Daches), die statisch definierte Eigenschaft *TotalRoofArea* (Zahlenwert der Dachfläche) und das dynamisch definierte *Pset_RoofCommon* zur Verfügung, das bereits im Beispiel in Abbildung 5.8 erläutert wurde. Daraus ergibt sich insgesamt ein hoher Detaillierungsgrad für die geometrische Darstellung, jedoch nur ein niedriger für die nichtgeometrische.

Anmerkung 1 in Tabelle 5.1 bedeutet, daß die Geometrie der Bauelemente „Tür“ und „Fenster“ auf zwei unterschiedliche Arten angegeben werden kann: Die einfachere Möglichkeit sieht vor, daß die Abmessungen ihrer Bestandteile parametrisch über statische Eigenschaftsklassen angegeben werden (*IfcDoorLiningProperties*,

IfcDoorPanelProperties, *IfcWindowLiningProperties* und *IfcWindowPanelProperties*). Bei der aufwendigeren Variante wird dagegen für die Blockdefinition eine vollständige, explizite Geometrie definiert. Anmerkung 2 bedeutet, daß für die Bauelemente „Tür“ und „Fenster“ mehrere PSETs existieren, die sich auf die einfügende Klassen (*IfcDoor*, *IfcWindow*) statt auf die Typdefinitionen beziehen. Dies ist keine gute Lösung, da die Eigenschaften ja für alle eingefügten Instanzen gelten und deshalb eigentlich der Typdefinition zugeordnet sein sollten.

5.5.5 Bauelemente des ST-4 Projektes

Auch die Erweiterung der IFC um die Stahlkonstruktion, die im ST-4 Projekt der IAI durchgeführt wird [56], ist für den Komplettbau von Bedeutung. Es handelt sich wiederum um Bauelemente, die dazu notwendigen Eigenschaftsdefinitionen und Erweiterungen der Ressourcen. Die Profil-Ressource der IFC2x wird beispielsweise um alle Profiltypen aus der Produktschnittstelle Stahlbau und aus CIS/2 erweitert werden, so daß das gesamte Spektrum des Stahlbaus abgedeckt ist. Diese Erweiterung ordnet sich problemlos in das bestehende Resource-Schema der IFC ein, so daß hier nicht weiter darauf eingegangen werden muß.

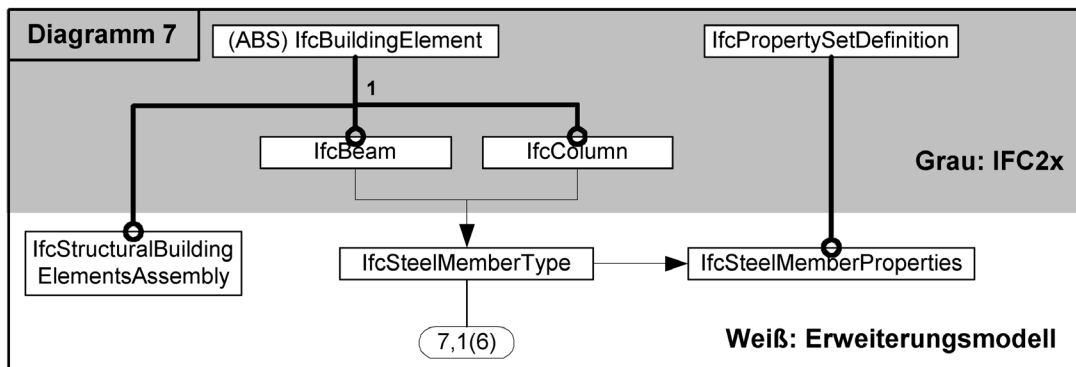


Abbildung 5.14: Bauelemente des ST-4 Projektes (EXPRESS-G)

An neuen Bauelementen wird lediglich die Klasse *IfcStructuralBuildingElementsAssembly* zur Darstellung von Baugruppen hinzukommen. Um außerdem die im Stahlbau übliche Struktur von Teilebeschreibung (Position) und Teileeinfügung zu ermöglichen, wird die Klasse *IfcSteelMemberType* eingeführt, die zur Blockdefinition von Stahlbauteilen dient. Sie kann von *IfcBeam* und *IfcColumn* eingefügt werden. *IfcSteelMemberType* erhält wiederum zusätzliche semantische Informationen in Form von *IfcSteelMemberProperties*.

Abbildung 5.14 zeigt die Klassenstruktur dieser Bauelemente. Die bereits vorhandenen Klassen der IFC2x sind im grau hinterlegten Bereich dargestellt, während der Bereich der Modellerweiterung weiß ist. In Tabelle 5.2 sind die Eigenschaften der neuen Bauelemente angegeben.

Bauelemente		Geerbte Datenstrukturen					Weitere Eigenschaften		Detaillierung	
Klasse	Funktion	Geometrie	Auflösung	lock-mech.	Material	Öffnung	weitere Eigenschaften	geom.	nicht-geom.	
<i>IfcRoof</i>	Dach	BREP	<i>IfcRoof, IfcSlab, IfcBeam</i>	-	Uneingeschränkt	Ja	<i>IfcRoofTypeEnum, TotalRoofArea, Pset_RoofCommon</i>	Hoch	Niedrig	
<i>IfcSlab</i>	Decke, Dach, Wandscheibe	Extrusion mit konst. Dicke oder CSG	-	-	Einfache Materialdefinition	Ja	<i>IfcSlabTypeEnum, SlabArea, SlabVolume, Pset_SlabCommon, Pset_RoofSlab</i>	Hoch	Niedrig	
<i>IfcWall</i>	Wand	-	-	-	Schichtung	Ja	<i>WallGroundArea, WallSideArea, WallVolume, Pset_WallCommon</i>	-	iedrig	
<i>IfcWall-Standard-Case</i>		Extrusion mit konst. Höhe oder CSG						Hoch	Niedrig	
<i>IfcBeam</i>	Träger	Extrusion oder CSG	-	-	Uneingeschränkt	-	<i>CrossSectionArea, BeamVolume, Pset_BeamCommon</i>	Mittel	Niedrig	
<i>IfcColumn</i>	Stütze	Extrusion oder CSG	-	-	Uneingeschränkt	-	<i>CrossSectionArea, BeamVolume, Pset_BeamCommon</i>	Mittel	Niedrig	
<i>IfcDoor</i>	Tür	Parametrisch oder explizit ¹ als Block	-	Ja	<i>IfcDoorStyle-Construction-Enum</i>	-	<i>Pset_DoorCommon², Pset_DoorPanelCommon², IfcDoorStyleOperationEnum</i>	Hoch	Mittel	
<i>IfcDoor-Style</i>								Hoch	Mittel	
<i>IfcWindow</i>	Fenster	Parametrisch oder explizit ¹ als Block	-	Ja	<i>IfcWindowStyle-Construction-Enum</i>	-	<i>Pset_WindowCommon², Pset_WindowPanelCommon², IfcWindowStyleOperationEnum</i>	Hoch	Mittel	
<i>IfcWindow-Style</i>								Hoch	Mittel	

Tabelle 5.1: Bauelemente der IFC2x und deren Eigenschaften

Bauelemente		Geerbte Datenstrukturen					Weitere Eigenschaften	Detailierung	
Klasse	Funktion	Geometrie	Auflösung	Block-mech.	Material	Öffnung		geom.	nicht-geom.
<i>IfcStructuralBldgElementsAssembly</i>	Baugruppe	-	<i>IfcBeam</i> , <i>IfcColumn</i>	-	-	-	<i>IfcStructuralElementsAssemblyTypeEnum</i>	-	Hoch
<i>IfcBeam</i> <i>IfcColumn</i> <i>IfcSteelMemberType</i>	Stütze und Träger	Extrusion	Ja	Ja	Einfache Materialdefinition	-	<i>IfcSteelMemberProperties</i>	Mittel	

Tabelle 5.2: Bauelemente des ST-4 Projektes und deren Eigenschaften

5.6 Bewertung hinsichtlich des Komplettbaus

Das Datenmodell der IFC einschließlich seiner Erweiterung im ST-4 Projekt ist hinsichtlich seiner Eignung als Grundlage für das Komplettbaumodell folgendermaßen zu bewerten.

- 1) Die generischen Datenstrukturen des Datenmodells sind sehr gut für den Komplettbau geeignet und sollten soweit wie möglich genutzt werden. Ihr Nachteil, nämlich die zum Teil große Komplexität, muß dafür in Kauf genommen werden.
- 2) Die hochentwickelten geometrischen Fähigkeiten der IFC erfüllen fast alle Anforderungen des Komplettbaus. Der mögliche geometrische Detaillierungsgrad der meisten Bauelemente ist dementsprechend hoch. Lediglich die Profil-Ressource wird um neue Profilformen ergänzt werden müssen, da die raumbildenden Bauteile für Dach und Wand weder durch die IFC2x noch durch das ST-4 Projekt abgedeckt sind.
- 3) Die Material-Ressource der IFC2x ist sehr leistungsfähig. Besonders die Varianten für Materiallisten und -schichtungen sind für den Leichtbau von Bedeutung. Dagegen fehlt die Möglichkeit zur Erfassung von Bauprodukten, die im Komplettbaumodell eine große Bedeutung haben.
- 4) Die in den IFC vorgeschriebenen Konzepte der Projekt- und Bauwerkstruktur eignen sich für den Komplettbau.
- 5) Die in den IFC vorhandenen Bauelemente sind grundsätzlich für die Zwecke des gewerkeübergreifenden Komplettbaumodells geeignet. Es müssen jedoch einige Änderungen und Ergänzungen vorgenommen werden. So sollen von der Anwendung her gleichwertige Bauelemente bezüglich ihrer grundlegenden Eigenschaften auch gleichwertig modelliert werden. Derzeit ist zum Beispiel für *IfcRoof* eine Dekomposition vorgesehen, für *IfcWall* jedoch nicht.

- 6) Folgende, im Komplettbau wichtige Bauelemente können in der aktuellen Version der IFC nicht dargestellt werden: Glasfassaden und verglaste Flächen in Dächern und Wänden, Fensteranordnungen, Fundamente und Bodenplatten, Industrietore, Dachbelichtungselemente sowie Rauch- und Wärmeabzugseinrichtungen. Entsprechende Klassen müssen im Komplettbaumodell neu geschaffen werden.
- 7) Die in den IFC2x existierenden Bauelemente verwenden unterschiedliche Konzepte zur Beschreibung nichtgeometrischer Eigenschaften, nämlich direkt in den Bauelementklassen definierte Attribute, statische Eigenschaftsklassen und dynamische Modellerweiterungen. Die in Kapitel 4 dieser Arbeit genannten besonderen Anforderungen für Bauelemente sind jedoch zu einem großen Teil nicht abgedeckt. Die fehlenden Anforderungen müssen im Komplettbaumodell ergänzt werden, wobei ein einheitliches Konzept einzuhalten ist. Dasselbe gilt für die Anforderungen, die an die für den Komplettbau neu zu schaffenden Bauelemente gestellt werden.
- 8) Die für Ingenieurzwecke benötigten hohen Detaillierungsgrade sind durch die ausschließliche Darstellung eines Gebäudes mit Bauelementen, also relativ großen Teilen mit globaler Funktion im Gebäude, nicht zu erreichen. Das Konzept des Bauelementes muß deshalb um kleinformatigere Bauteile ergänzt werden. Die Darstellung dieser Bauteile muß nicht nur kompatibel zur Geometrie der übergeordneten Bauelemente sein, sondern auch zu deren Material- oder Bauproduktzuordnung sowie deren semantischen Eigenschaften.

Die Definition des Komplettbaumodells als Erweiterung der IFC2x ist also möglich und unbedingt sinnvoll. Sie wird im folgenden Kapitel durchgeführt.

6 Datenmodell Komplettbau

6.1 Allgemeines

In diesem Kapitel wird das Datenmodell für den Komplettbau als Erweiterung der IFC2x definiert. Es erfüllt die in Kapitel 4 aufgestellten Vorgaben bezüglich der Modellstruktur und der zu behandelnden Dateninhalte. Der Umfang dieses Kapitels soll sich in Grenzen halten. Anstelle von vollständigen und ausführlichen Beschreibungen aller Klassen wird deshalb versucht, die Zusammenhänge, die Datenstrukturen und auch die Hintergründe des Modells zu vermitteln. Die Inhalte der Darstellung werden dagegen nur bis zu einem gewissen Grad beschrieben. Die vollständigen und formalen Spezifikationen aller Klassen und Typen sind im Anhang zu dieser Arbeit zu finden.

6.2 Vorüberlegungen

6.2.1 Leitgedanken für die Modellierung

Bereits bei den ersten Modellierungsversuchen stellte sich heraus, daß die unbedachte Anwendung der in den IFC2x zur Verfügung stehenden Datenstrukturen auf hohe Detaillierungsgrade sowohl zu einer ausufernden Komplexität des Modells als auch zu einem sehr großen Datenvolumen in den Austauschdateien führen würde. Um dies zu vermeiden, sind die folgenden Leitgedanken zu berücksichtigen.

- 1) Das Modell soll so einfach wie möglich sein. So wird zum Beispiel darauf verzichtet, so komplexe Strukturen wie Blockdefinitionen durch Dekomposition in Bestandteile aufzulösen.
- 2) Auch in den hohen Detaillierungsgraden müssen nicht immer alle Sachverhalte ausführlich dargestellt werden. Für diese Fälle soll das Modell die Möglichkeit zur vereinfachten Darstellung zur Verfügung stellen. Dies geschieht bevorzugt durch Anwendung der Klasse *IfcProxy*, mit der Objekte ohne explizite Geometrie dargestellt werden.
- 3) Die Detaillierungsgrade im Modell müssen variabel sein, um nachträglich Sichten definieren zu können. Dies ist notwendig, um später flexibel bei der Implementierung und Anwendung des Modells zu sein.
- 4) In der einschlägigen Normung vorhandene Typisierungen von Bauarten und Bauteilen sollen soweit wie möglich berücksichtigt werden. Dagegen ist es nicht das Ziel der Modellierung, Inhalte von Bemessungsnormen oder spezielle Bemessungsparameter abzubilden.
- 5) Das Modell kann den vielfältigen Ansprüchen nur dann genügen, wenn das Ingenieurdenken als Vorbild genommen wird. Durch die Anwendungsprozesse aus Kapitel 4 ist dies auf konzeptioneller Ebene gewährleistet. Das Vorbild soll jedoch auch im Detail gelten.

6.2.2 Zentrale Begriffe

Abbildung 6.1 zeigt die zentralen Begriffe des Komplettbaumodells am Beispiel einer Wand: Der obere Teil der Abbildung beinhaltet den Detaillierungsgrad der Übersichtsplanung. Die darin vorkommenden Begriffe der Wand als ein ganzheitliches Bauelement mit globaler Funktion im Gebäude und die daran geknüpfte Anforderung eines Mindestwärmedurchlaßwiderstandes wurden bereits erläutert. Für die in der Fachplanung notwendige Detaillierung, die im unteren Teil abgebildet ist, wird die Wand durch mehrere Bauteile dargestellt, die nur für den Fachplaner von Bedeutung sind. Dieser Übergang wird als Auflösung oder Dekomposition des Bauelementes bezeichnet.

Ein weiterer zentraler Begriff ist der des Bauproduktes: Im Komplettbau werden Bauteile anstatt durch eine individuelle Aufschlüsselung von Form, Werkstoff und weiteren Eigenschaften oft einfach durch Angabe des Herstellers und einer quasi-standardisierten Produktbezeichnung, zum Beispiel durch eine Bestellnummer, beschrieben.

Die Zuordnung eines Produktes zu einem Bauteil impliziert in der Regel auch Eigenschaften. Die tatsächlichen Eigenschaften der im Beispiel gezeigten Wand werden jedoch erst durch das Zusammenwirken aller Bauteile bestimmt. Angesichts der Vielfalt der möglichen Bauarten und der großen Zahl an Bauteilen innerhalb eines Bauelementes wird deutlich, daß die Erfassung sämtlicher Eigenschaften im Produktmodell mit vertretbarem Aufwand nicht möglich ist und hier deshalb nicht versucht wird. Es bleibt also die Verantwortung des Planers, daß die in der Übersichtsplanung gegebenenfalls gestellten Anforderungen an die Bauelemente durch die richtige Verwendung der Bauprodukte erfüllt werden.

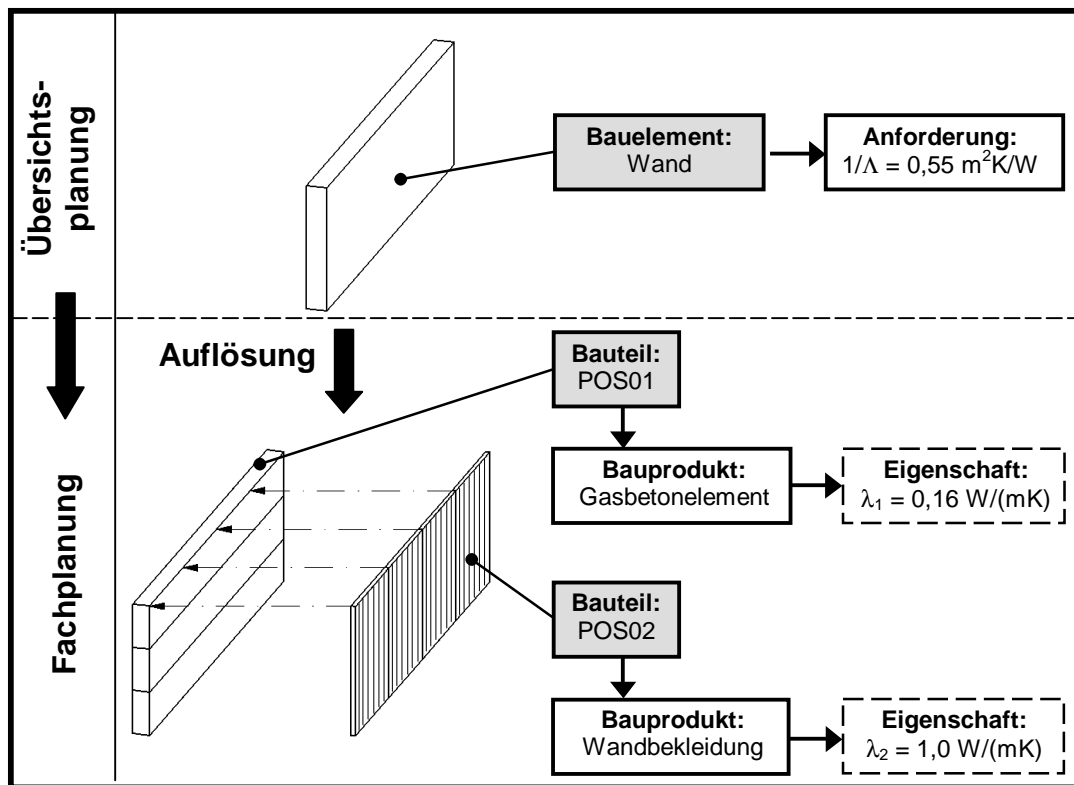


Abbildung 6.1: Zentrale Begriffe des Datenmodells

6.2.3 Notation

Zur Verdeutlichung der Zusammenhänge innerhalb des Modells werden die neu definierten Klassen in der EXPRESS-G-Notation dargestellt, die bereits im vorangegangenen Kapitel verwendet wurde. Die Diagramme sind zum Teil stark vereinfacht. Um die Zusammenhänge des neuen Modells zu den IFC zu zeigen, werden beispielsweise auch die betroffenen Oberklassen der IFC2x dargestellt. Diese befinden sich zur besseren Erkennung in grau hinterlegten Bereichen der Diagramme.

Darüber hinaus werden zur Anschauung konkrete Beispiele von Austauschstrukturen aufgeführt. Da vollständige Austauschdateien, die in den IFC in der Regel sehr umfangreich sind, aus Platzgründen nicht in Frage kommen, wurde eine spezielle Darstellung entwickelt. Abbildung 6.2 skizziert in nicht formaler, aber anschaulicher und sehr kompakter Weise eine Austauschdatei mit zwei Bauelementen und einer Anforderung. Im Gegensatz zu EXPRESS-G symbolisieren die Kästchen keine Klassen, sondern Instanzen. In Anlehnung an die Syntax gemäß ISO 10303 hat jede Instanz eine sogenannte Gitternummer, anhand derer sie innerhalb der Datei referenziert werden kann. Bauelemente (#1 und #2) und Bauteile sind grau hinterlegt. Eigenschaften und Anforderungen (#4) sind weiß und schattiert, Instanzen von Relationsklassen (#3) dagegen weiß und nicht schattiert. Aus Platzgründen werden Eigenschaften zum Teil einfach als Text in die Kästchen der Instanzen geschrieben. Verweise zwischen den Instanzen werden durch Pfeile symbolisiert.

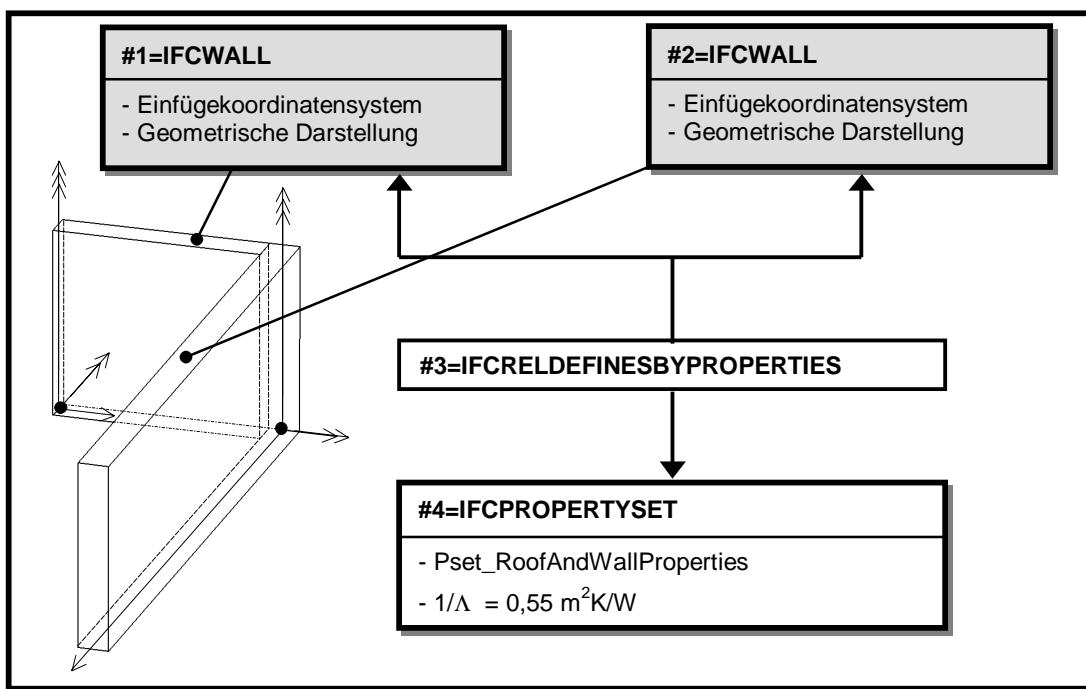


Abbildung 6.2: Anschauliche Darstellung einer Austauschstruktur

6.2.4 Übersicht und Einordnung in die IFC2x

In Kapitel 4 wurde festgelegt, daß das Produktmodell Komplettbau aus den drei gewerkespezifischen Modellen für Dach und Wand, Dach- und Wandöffnungen und Glasbau bestehen soll, die in einem gewerkeübergreifenden Modell verankert sind. Dazu wird jedes Modell in einem separaten EXPRESS-Schema definiert und gemäß Abbildung 6.3 in die Hierarchie der funktionalen Schichten der IFC eingeordnet.

Die drei gewerkespezifischen Modelle werden als neue Domänen im Domain Layer angesiedelt. Alle Klassen, die gewerkeübergreifende Bedeutung haben, müssen dagegen in den tieferen Schichten der IFC angeordnet werden. Inhaltlich gesehen ist es sinnvoll, diese Klassen in bestehende Schemata der IFC2x zu integrieren, die in der Abbildung grau hinterlegt sind. Da dies teilweise jedoch den unveränderlichen Teil der IFC-Plattform betrifft, werden die Klassen zunächst in separaten EXPRESS-Schemata definiert, deren Namen auf „EXTENSION“ endet. Die für später vorgeschlagenen Integrationen sind in der Abbildung durch schwarze Pfeile gekennzeichnet.

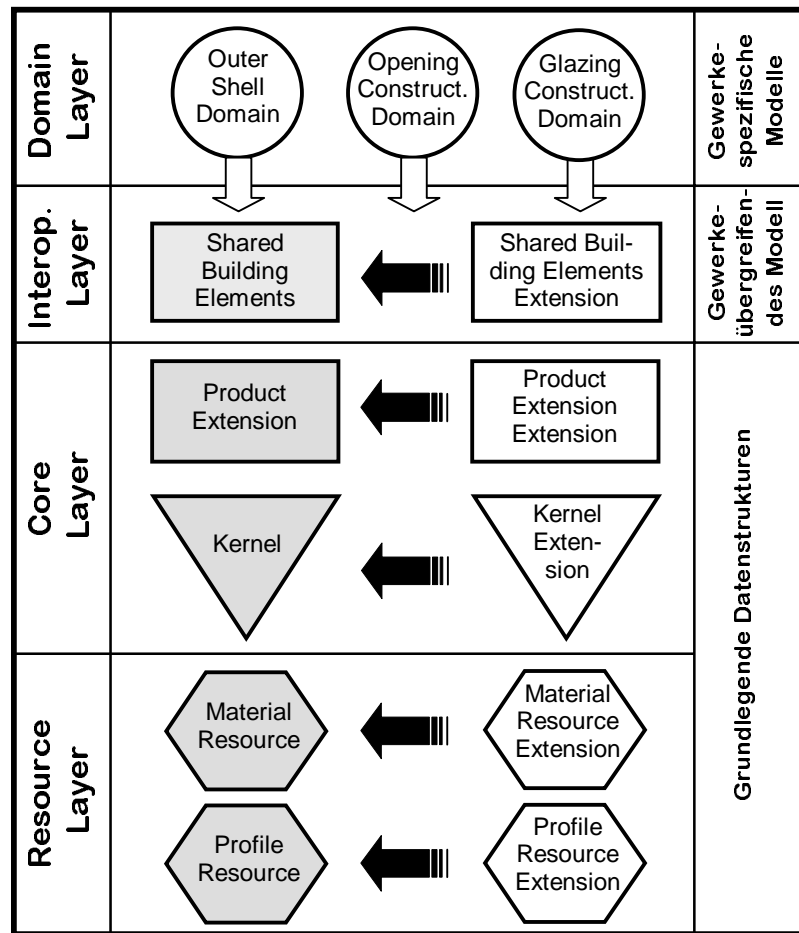


Abbildung 6.3: Einordnung in die IFC2x

Die neu zu definierenden Klassen werden gemäß ihrer modelltechnischen Funktion und der darzustellenden Inhalte auf die insgesamt acht neuen EXPRESS-Schemata verteilt, und zwar nach folgendem Prinzip: In den Resource- und Core Layer gehören grundlegende Datenstrukturen. Im Interoperability Layer wird das gewerkeübergreifende Modell mit allen Bauelementen und Anforderungen definiert. Die gewerkespezifischen Modelle werden, wie gesagt, im Domain Layer angeordnet. Sie enthalten die Bauteile, in die die Bauelemente aufgelöst werden, und die dazu verwendeten Bauprodukte.

6.3 Grundlegende Datenstrukturen

6.3.1 Profilformen

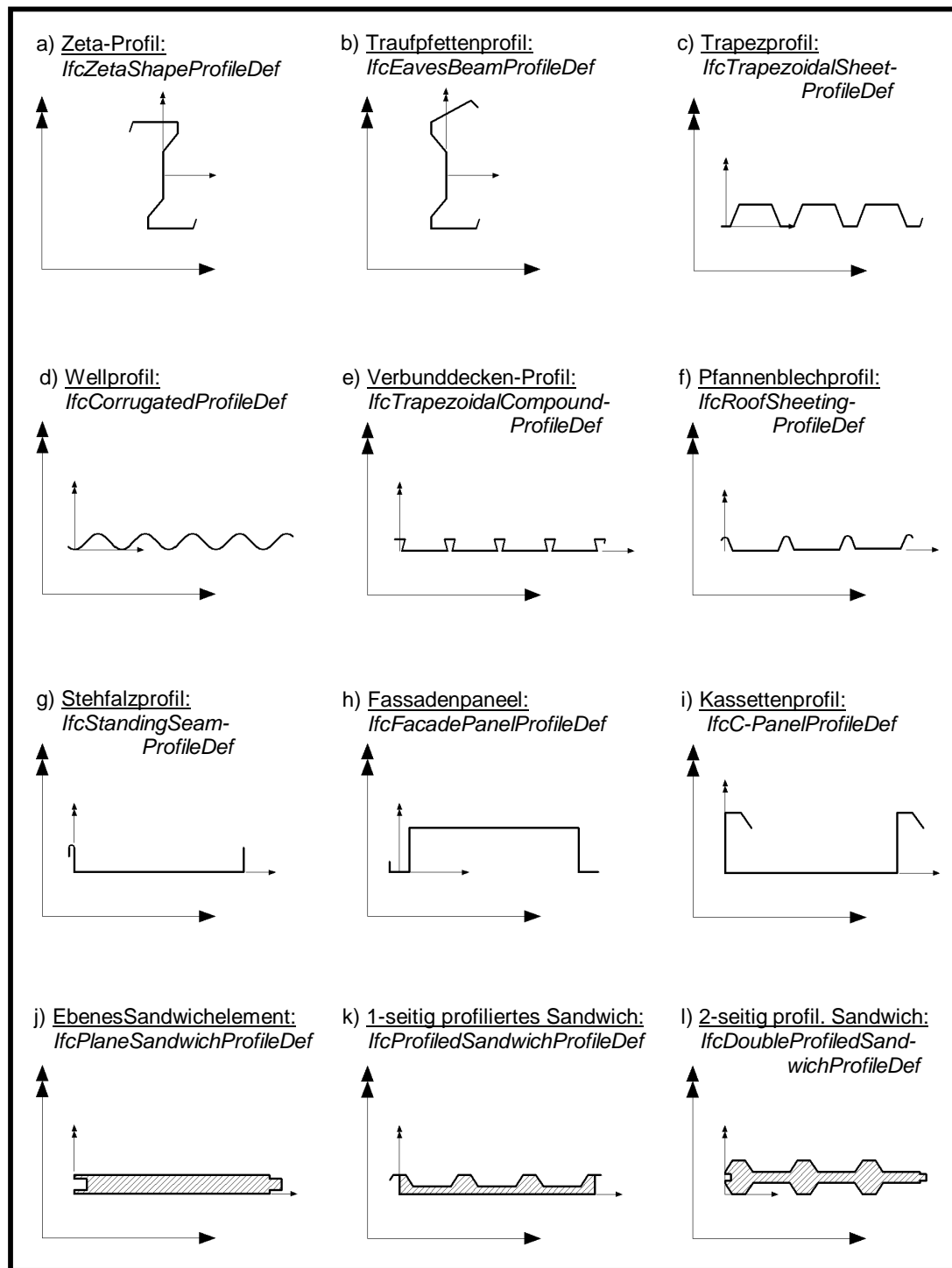


Abbildung 6.4: Neue Profilformen

Profildefinitionen werden in den IFC ausschließlich als Teil expliziter Geometriedarstellung verwendet, entweder zur Oberflächen- oder zur Festkörperextrusion. Sie dürfen per definitionem keine semantischen Informationen enthalten, wie etwa Tragfähigkeitsangaben. Bereits mit den Mitteln der IFC2x ist jede Profilform durch freie Polygone darstellbar. Es ist jedoch sinnvoll, anstelle dieser datentechnisch aufwen-

digen Möglichkeit das Spektrum an Standardprofilformen der IFC gemäß Abbildung 6.4 zu erweitern.

Diese Notwendigkeit besteht für die längsprofilierten Bauteile des Leichtbaus. Alle Profile sind, entgegen der Darstellung in Abbildung 6.4, als geschlossene Flächen definiert, so daß sie zur Festkörperextrusion herangezogen werden können. Dies ist notwendig, da im Stadium der Verlegeplanung die Form der Bauteile unter Umständen mit *boole'schen Operationen* bearbeitet werden muß. Die Detaillierung der Profile ist für die gängigen Bauarten bis zur Makroprofilierung sinnvoll, das heißt ohne Sicken, Ausrundungen oder Linierungen, so wie es den konstruktionsrelevanten Herstellerinformationen entspricht. Stark vereinfacht ist die Ausbildung der Querstöße bei Sandwichprofilen (Typen j bis l), was jedoch für die Erstellung von Verlegeplänen ausreichend ist.

Die Darstellung von Bauteilen, die auch in Querrichtung profiliert sind, also beispielsweise kofferrörmige Fassadenelemente, ist durch Extrusion von Profilen nicht möglich. Sie müssen statt dessen entweder vereinfacht als Quader ohne die abgekannten Ränder oder über eine BREP-Geometrie dargestellt werden.

Für den Glas- und Fensterbau werden keine speziellen Profiltypen eingeführt, da die Formenvielfalt in diesen Bereichen so groß ist, daß eine auch nur annähernd vollständige Typisierung unrealistisch ist.

Die Klassen werden im Schema *IFCPROFILERESOURCEEXTENSION* definiert. Bei allen Typen werden vollständige Bauteilbreiten dargestellt. Dazu wird bei trapez- und wellprofilierten Formen (Typen c bis f sowie k und l) die Anzahl der Rippen angegeben, aus der sich die Verlege- und die Bauteilbreite ergeben. Der Übersichtlichkeit halber sind in Abbildung 6.4 keine Maße eingetragen.

6.3.2 Bauprodukte

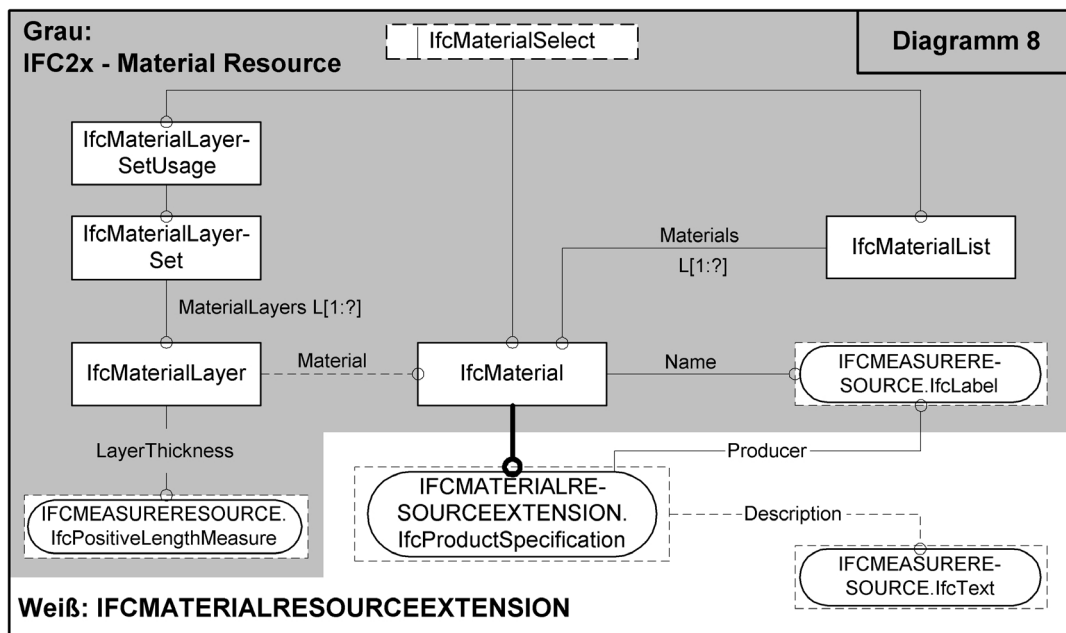


Abbildung 6.5: Erweiterte Material-Ressource (EXPRESS-G)

Die Spezifikation und Zuordnung von Bauprodukten sollte entsprechend ihrer Bedeutung im Komplettbau möglichst hoch im Vererbungsbaum von Bauelementen oder Bauteilen angesiedelt sein. Dazu bietet sich die Erweiterung der Material-Ressource der IFC2x an, so daß es möglich ist, anstelle eines Materials auch ein

Bauprodukt anzugeben. Das hat den Vorteil, daß die erweiterte Datenstruktur allen in den IFC2x vorhandenen Bauelementen zur Verfügung steht. Außerdem können die bereits vorhandenen Möglichkeiten zur Definition von Materiallisten und Materialschichtungen genutzt werden.

Die Erweiterung wird im Schema *IFCMATERIALRESOURCEEXTENSION* definiert, siehe Abbildung 6.5. Um die Zusammenhänge zu den IFC zu verdeutlichen, ist die ursprüngliche Material-Ressource noch einmal abgebildet (grau hinterlegter Bereich). Die Erweiterung besteht lediglich aus der neuen Klasse *IfcProductSpecification*, die als Unterklasse von *IfcMaterial* definiert wird. Die Produktbezeichnung wird im geerbten Attribut *Name* angegeben, der Hersteller im neuen Attribut *Producer*. *Description* bietet die optionale Möglichkeit einer zusätzlichen Beschreibung des Produktes in Textform. Die Zuordnung eines Bauproduktes zu einem Bauteil impliziert automatisch alle Eigenschaften, die der Hersteller für sein Produkt angibt. Eine Möglichkeit zur expliziten Angabe von Produkteigenschaften ist im Komplettbaumodell nicht vorgesehen.

Abbildung 6.6 zeigt eine Austauschdatei, in der der Schichtaufbau einer Kassettenwand dargestellt ist. Dabei wird dem Bauelement #1 eine vierschichtige Produktdefinition zugeordnet. In den Instanzen der Klasse *IfcMaterialLayer* ist jeweils die Dicke der Schicht angegeben, in den Instanzen *IfcProductSpecification* das zugehörige Bauprodukt. Aus Platzgründen sind die Hersteller der Produkte in der Abbildung nicht aufgeführt. Die Dämmschicht #7 besitzt keine Dicke, die für das Bauelement angerechnet werden kann, da sie zwischen den Stegen der Kassettenprofile liegt.

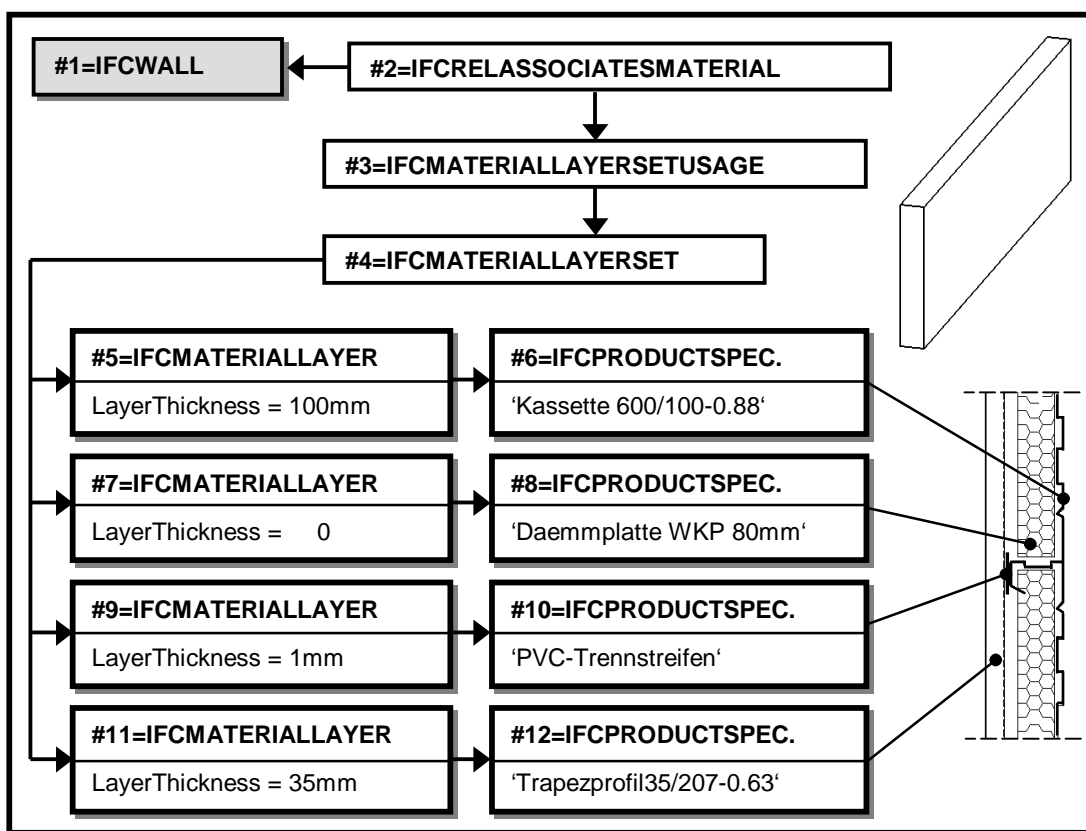


Abbildung 6.6: Beispiel für einen geschichteten Wandaufbau

6.3.3 Befestigungen

Diejenigen Kleinstteile eines Bauwerks, die wie zum Beispiel Befestigungsmittel in sehr großen Stückzahlen eingesetzt werden, müssen stark vereinfacht dargestellt werden. Dies gilt sowohl für die Teile selbst als auch für deren Anordnung. Dazu wird im Schema *IFCKERNELEXTENSION* die abstrakte Klasse *IfcRelFixes* definiert, siehe Abbildung 6.7. Sie stellt keine Verbindung zwischen mehreren Elementen her, sondern besagt lediglich, daß diese Elemente in irgendeiner Form befestigt sind (Verweis im Attribut *RelatingElements*). Worauf diese Befestigung erfolgt, ergibt sich aus dem Zusammenhang. Im Attribut *ConnectionElement* wird das verwendete Befestigungsmittel referenziert, das im einfachsten Fall ein *IfcProxy* ist, also keine Geometrie besitzt.

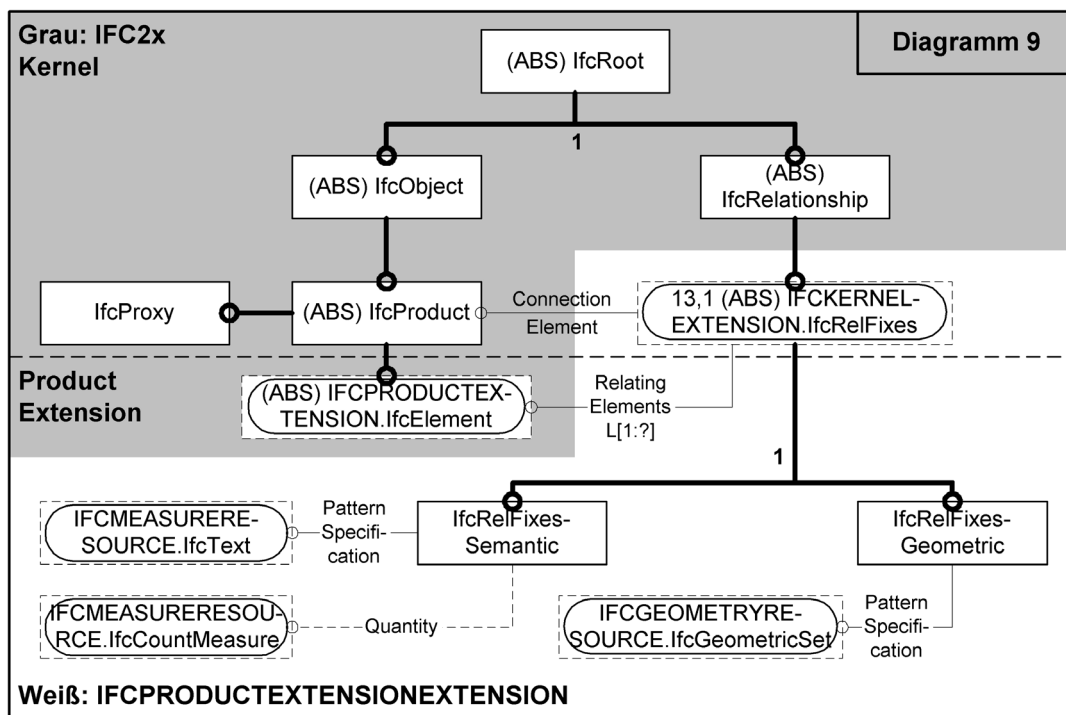


Abbildung 6.7: Erweiterung der IFC um Befestigungen (EXPRESS-G)

Das Muster, in dem die Befestigungsmittel angeordnet werden, wird in den im Schema *IFCPRODUCTEXTENSIONEXTENSION* definierten Unterklassen von *IfcRelFixes* festgelegt. *IfcRelFixesSemantic* beschreibt das Anordnungsmuster in Textform (Attribut *PatternSpecification*). Dadurch ist die anfallende Datenmenge sehr gering. Im Attribut *Quantity* kann die dazu benötigte Stückzahl der Verbindungsmittel angegeben werden. Die Klasse *IfcRelFixesGeometric* legt dagegen die Anordnung explizit fest. Dies geschieht durch Verweis über *IfcGeometricSet* auf Scharen von Punkten, Linien oder Flächen.

6.3.4 Allgemeines Bauteil: *IfcConstructionalElement*

Zur Darstellung von Bauteilen wird im Schema *IFCPRODUCTEXTENSIONEXTENSION* die allgemeingültige Oberklasse *IfcConstructionalElement* definiert, siehe Abbildung 6.8. Sie ist modelltechnisch der Klasse *IfcBuildingElement* der IFC2x gleichgestellt und verfügt deshalb über dieselben, geerbten Attribute, die vereinfacht mit Pfeilen im Vererbungsbaum angedeutet sind. Beide Klassen können zum Bei-

spiel dieselben Typdefinitionen benutzen. Das bedeutet, daß etwa ein Stahlträger, dargestellt durch *IfcSteelMemberType*, sowohl von einem Bauelement (*IfcBeam*) als auch von einem Bauteil (*IfcConstructionalElement*) eingefügt werden kann. Der Unterschied besteht dann lediglich darin, daß der Träger im ersten Fall eine globale Funktion für das Bauwerk hätte, beispielsweise als tragender Unterzug, während er im zweiten Fall nur für den Stahlbaukonstrukteur von Interesse wäre, beispielsweise als untergeordnete Konsole. Obwohl *IfcConstructionalElement* selbst instanziiert werden kann, werden in den gewerkespezifischen Modellen spezielle Unterklassen definiert, wie es die drei Anknüpfungspunkte im unteren Bereich von Abbildung 6.8 anzeigen.

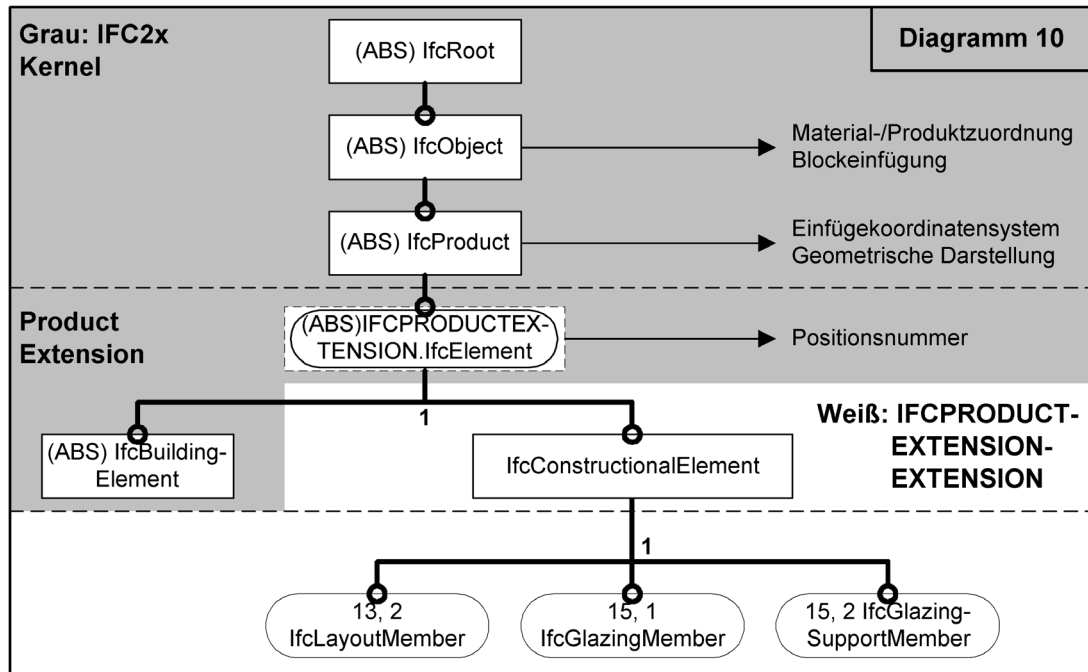


Abbildung 6.8: Erweiterung der IFC um Bauteile (EXPRESS-G)

6.3.5 Auflösung von Bauelementen in Bauteile

Die IFC2x bieten mit der Relation *IfcRelDecomposes* eine universelle Möglichkeit, Bauelemente in Komponenten aufzulösen. Für komplexe Bauelemente, die wie geschichtete Dächer und Wände oder wie Glasfassaden in Bauteile aufgelöst werden sollen, müssen die Rahmenbedingungen der Auflösung jedoch sehr genau definiert werden. Dies soll jeweils für den konkreten Anwendungsfall geschehen, also innerhalb der gewerkespezifischen Modelle. An dieser Stelle wird das grundsätzliche Konzept hierzu vorgestellt.

Im allgemeinen Fall ist ein raumabschließendes Bauelement aus dem Blickwinkel des Fachplaners sowohl in Dickenrichtung als auch in seiner Ebene aufgelöst. Auflösung in Dickenrichtung bedeutet einen geschichteten Aufbau auf dem Detaillierungsniveau der Detailplanung. Auflösung in der Ebene bedeutet die Zerlegung der Schichten in Verlegebauteile auf dem Detaillierungsniveau der Verlegeplanung. Vor diesem Hintergrund erscheint eine zweistufig hierarchische Auflösung gemäß Abbildung 6.9a) sinnvoll, wobei die Stufen zweckmäßigerweise den Detaillierungsgraden entsprechen. Es wären jedoch relative Einfügekoordinatensysteme und vollständige geometrische Darstellungen für die Komponenten auf jeder Stufe notwendig, also auch für die Schichten selbst.

Anstelle dieser aufwendigen Modellierung wird im folgenden eine einfachere und flexiblere Variante gewählt, bei der das Bauelement unmittelbar in gleichwertige Bauteile aufgelöst wird, siehe Abbildung 6.9b). Bei Bedarf können diese sozusagen „nachträglich“ zu Schichten gruppiert werden, ohne daß dies jedoch eine geometrische Bedeutung hätte. Die im Gegensatz zur ersten Variante eigentlich fehlende Detaillierungsstufe der Detailplanung wird durch Verweis auf eine geschichtete Bauproduktdefinition ersetzt.

Diese Vorgabe betrifft lediglich die Auflösung in Bauteile. Es ist dagegen uneingeschränkt möglich, daß gemäß Abbildung 6.9b) aufgelöste Bauelemente als Komponenten in der Auflösung von anderen Bauelementen verwendet werden.

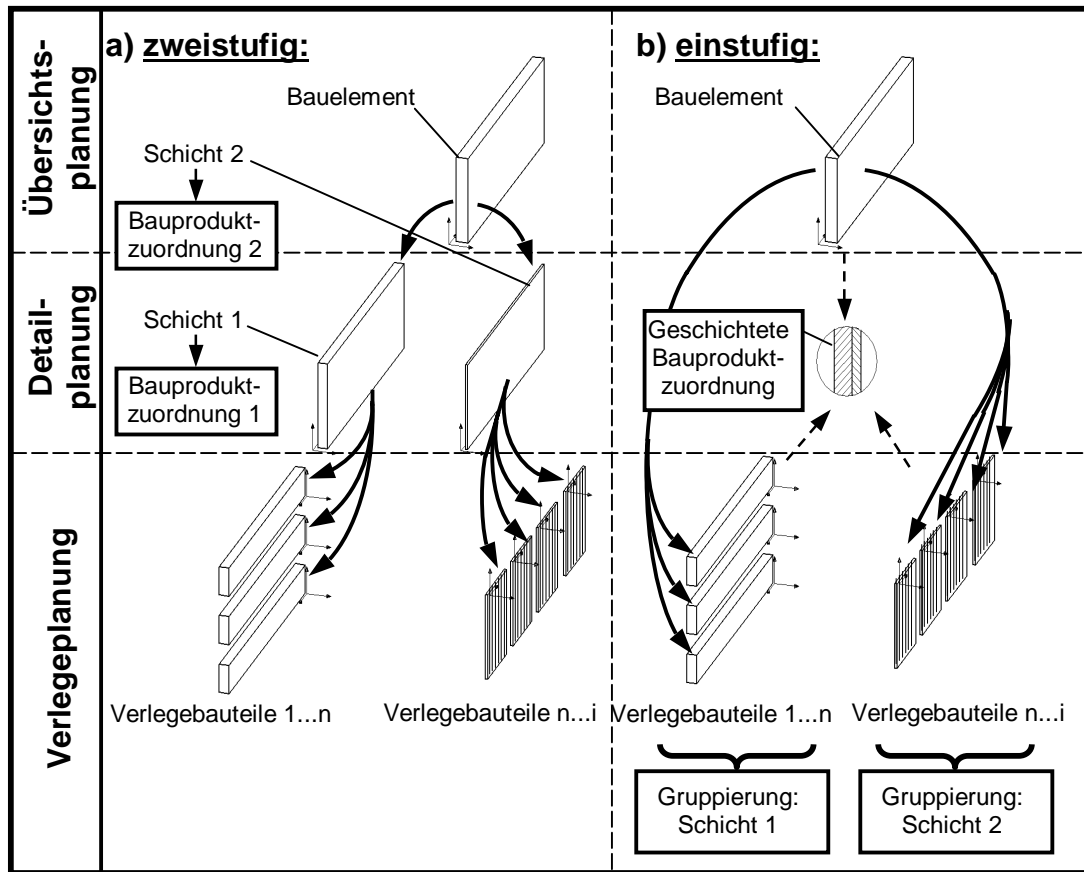


Abbildung 6.9: a) Zweistufige und b) einstufige Auflösung von Bauelementen

6.4 Gewerkeübergreifendes Modell

6.4.1 Grundzüge der Darstellung

In diesem Abschnitt werden die neuen Klassen des Interoperability Layers definiert, die zusammen mit den in den IFC2x bereits vorhandenen Bauelementen das gewerkeübergreifende Modell bilden. Gemäß Abschnitt 5.6 sind dazu drei Arten von Erweiterungen der IFC2x notwendig:

- 1) Alle raumabschließenden Bauelemente müssen im Bezug auf Dekomposition gleichwertig behandelt werden. Dächer (*IfcRoof*) können in den IFC2x beispielsweise in Einzelflächen aufgelöst werden (*IfcSlab*). Deshalb erhalten die Bauelemente der IFC2x, für die keine solche Dekomposition vorgesehen ist, entsprechende neue Unterklassen.

- 2) Für eine vollständige Darstellung des Komplettbaus müssen die IFC2x um Klassen für die folgenden Bauelemente erweitert werden: Glasfassaden und sogenannte Verglasungsflächen (verglaste Komponenten in Fassaden, Wänden oder Dächern), Fensteranordnungen, Fundamente und Bodenplatten, Industrietore, Dachbelichtungselemente sowie Rauch- und Wärmeabzugseinrichtungen.
- 3) An alle Bauelemente, also sowohl an die in den IFC2x bereits vorhandenen als auch an die in dieser Arbeit neu zu definierenden, müssen Anforderungen gestellt werden können. Um eine einheitliche, aber mit den Eigenschaften der vorhandenen Bauelemente verträgliche Lösung zu erzielen, werden alle Anforderungen als dynamische Eigenschaften (PSETs) definiert. Falls das Konzept der Anforderungen in der IAI Anklang findet, können diese Definitionen später in statische Eigenschaftsklassen überführt werden.

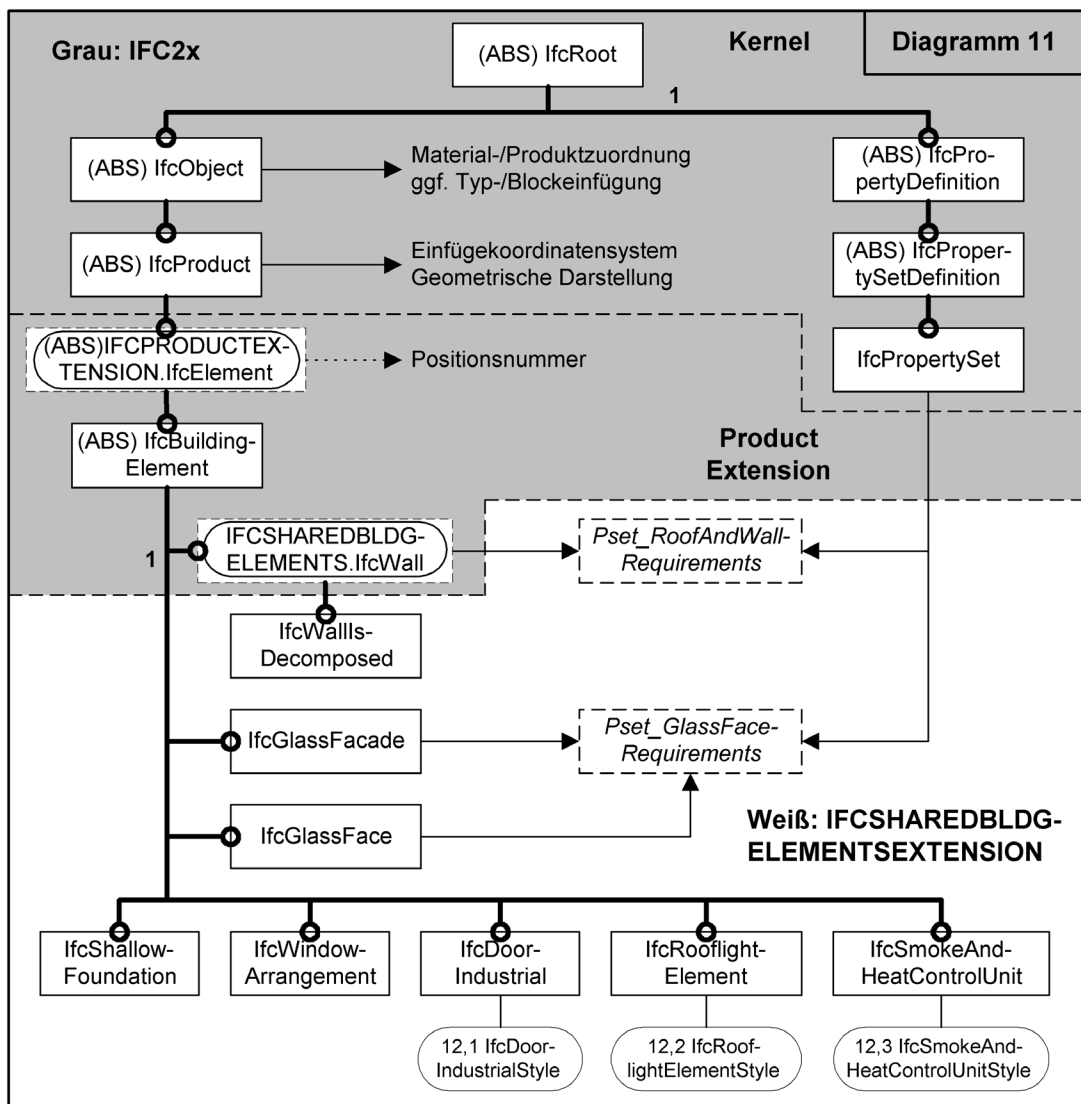


Abbildung 6.10: Bauelemente (EXPRESS-G)

Abbildung 6.10 zeigt die neuen Bauelemente des Komplettbaumodells, die als Unterklassen von *IfcBuildingElement* definiert werden (ausgenommen die Klasse *IfcWallIsDecomposed*, die eine direkte Unterklasse von *IfcWall* ist). Denjenigen Bauelementklassen, die keinen Typmechanismus unterstützen (*IfcWall*, *IfcGlassFacade* und *IfcGlassFace*), können die unter Punkt 3) angesprochenen Anforderungen direkt

zugeordnet werden. Für diejenigen Bauelemente, die sich des Typmechanismus bedienen (*IfcDoorIndustrial*, *IfcRooflightElement* und *IfcSmokeAndHeatControlUnit*), müssen die Anforderungen dagegen an die Typdefinition und nicht an die einfügende Klasse geknüpft werden, da dieselben Anforderungen ja für alle eingefügten Bauelemente gelten sollen. Die Klassen für die Typdefinitionen und deren Anforderungen sind in Abbildung 6.11 gezeigt.

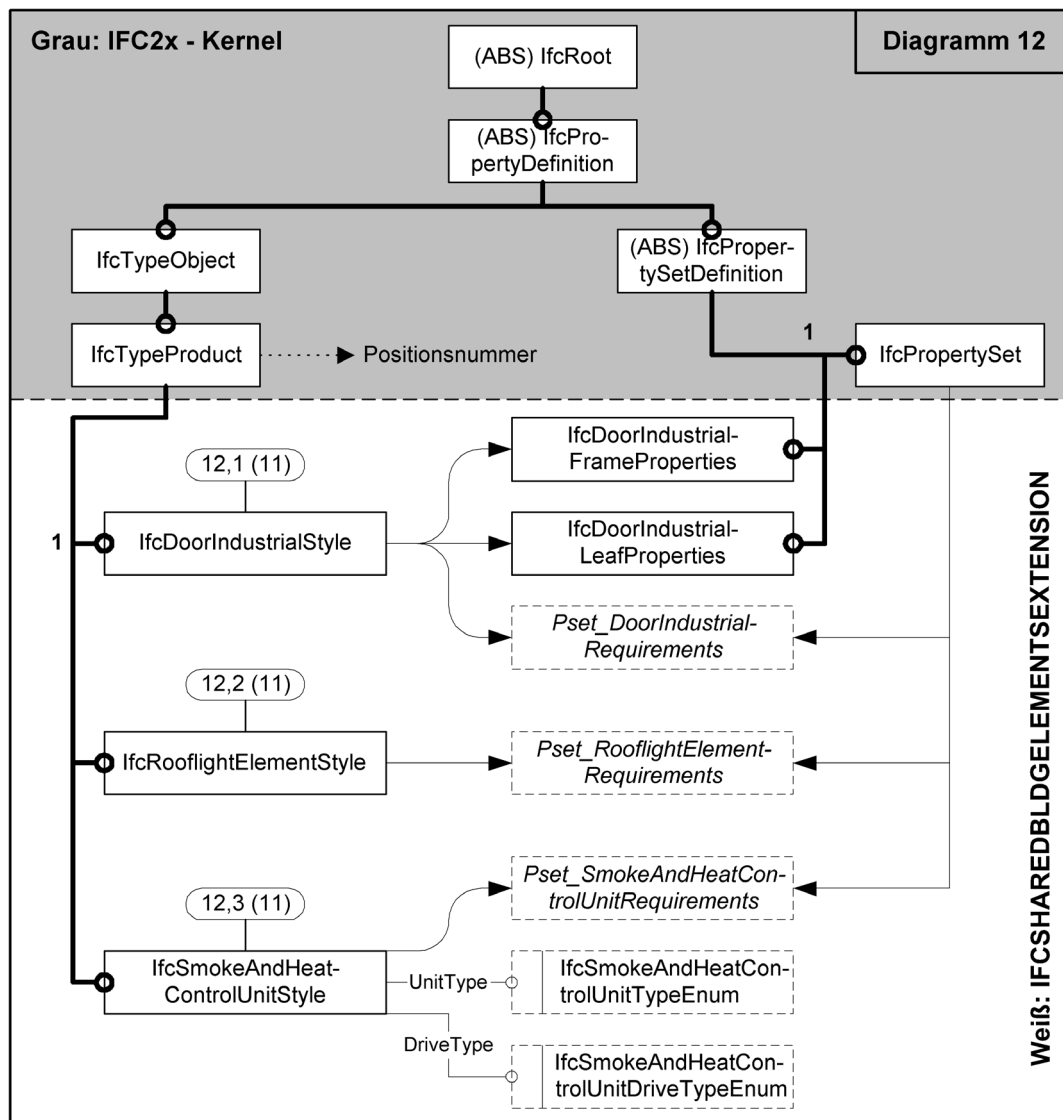


Abbildung 6.11: Typdefinitionen (EXPRESS-G)

Alle Verknüpfungen, also sowohl die von Anforderungen an Bauelemente (*IfcRelDefinesByProperties*) als auch die von Typdefinitionen an die einfügenden Klassen (*IfcRelDefinesByType*), sind in den Abbildungen vereinfacht durch Pfeile angedeutet. Die PSETs werden durch gestrichelte Kästchen mit kursiver Schrift symbolisiert. Alle neuen Klassen sind im Schema *IFCSHARED BLDGELEMENTSEXTENSION* definiert. Die Anforderungen an die in den IFC2x bereits bestehenden Bauelemente sind aus Platzgründen in diesem Kapitel nicht aufgeführt, die Definitionen der entsprechenden PSETs sind jedoch in **Anhang B** gegeben.

6.4.2 Zusammengesetzte Wände

6.4.2.1 Bauelement: *IfcWallIsDecomposed*

Da die Klasse *IfcWall* der IFC2x nicht in Komponenten aufgelöst werden kann, wird *IfcWallIsDecomposed* als neue Unterklasse eingeführt. Anstelle einer eigenen geometrischen Darstellung wird sie in andere Bauelemente aufgelöst. Diese Komponenten können flächige Bauelemente (*IfcSlab*, *IfcGlassFace*) und, zum Beispiel für Pfosten-Riegel-Konstruktionen, Bauelemente der Unterkonstruktion (*IfcBeam*, *IfcColumn*, *IfcStructuralBuildingElementsAssembly*) sein. *IfcWallIsDecomposed* wird jedoch nicht in Bauteile aufgelöst. Diese Stufe der Dekomposition findet erst in der Domäne für Dach und Wand statt.

6.4.2.2 Anforderungen: *Pset_RoofAndWallRequirements*

Mit *Pset_RoofAndWallRequirements* können die Anforderungen gemäß Tabelle 6.1 an Dächer (*IfcRoof*), Wände (*IfcWall*, *IfcWallIsDecomposed*) und deren Komponenten (*IfcSlab*) gestellt werden. Für die vollständigen Definitionen siehe **Anhang B.1**.

<i>Pset_RoofAndWallRequirements</i>	
<u>ConstructionType</u> :	Geforderte Bauart (Aufzählungstyp)
<u>Cladding</u> :	Vorgehängte, hinterlüftete Fassade, sofern erforderlich (Aufzählungstyp)
<u>FireRating</u> :	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm (Zeichenkette)
<u>AcousticRating</u> :	Erforderliches Schalldämm-Maß in dB (reelle Zahl)
<u>LoadBearing</u> :	Tragendes oder nichttragendes Bauelement (boole'scher Typ)
<u>ThermalTransmittanceCoefficient</u> :	Maximal erlaubter Wärmedurchgangskoeffizient (reelle Zahl)
<u>ThermalResistance</u> :	Erforderlicher Wärmedurchlaßwiderstand (reelle Zahl)
<u>SpecialRequirements</u> :	Weitere Anforderungen in Textform (Zeichenkette)

Tabelle 6.1: *Pset_RoofAndWallRequirements*

6.4.2.3 Beispiel

Abbildung 6.12 zeigt die Anwendung der Klasse *IfcWallIsDecomposed* (#1) samt Anforderungen. Die Wand besitzt keine geometrische Darstellung, sondern ist in drei Wandscheiben *IfcSlab* aufgelöst (#3, #4 und #6). Die skizzierte Stahlkonstruktion ist, ebenso wie der Betonsockel, in diesem Fall nicht als Teil der Wand definiert und deshalb in der Abbildung nicht aufgeführt. An alle Wandscheiben sind jeweils dieselben Anforderungen in Form der Instanz #8 gestellt: Sie sollen in Kassettenbauart ausgeführt werden, die Feuerwiderstandsklasse F30 nach DIN 4102 erfüllen und einen Wärmedurchlaßwiderstand von mindestens 0,55 m²K/W aufweisen. Die übrigen Anforderungen von *Pset_RoofAndWallRequirements* sind aus Platzgründen nicht angegeben.

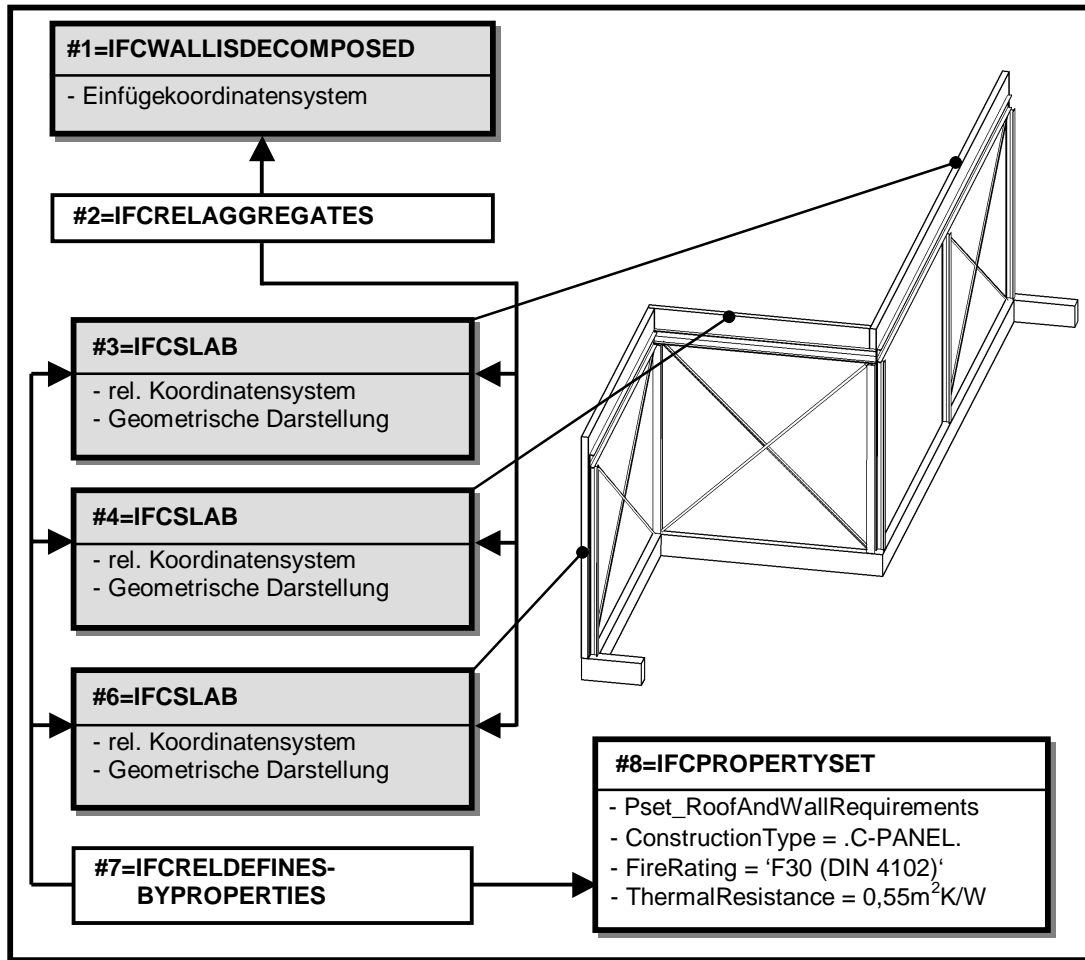


Abbildung 6.12: Zusammengesetzte Wand mit Anforderungen

6.4.3 Glasfassaden

6.4.3.1 Bauelement: *IfcGlassFacade*

Die Klasse *IfcGlassFacade* bietet zur Darstellung von Glasfassaden zwei Möglichkeiten:

- 1) Die Glasfassade ist nicht in Komponenten aufgelöst und verfügt über eine vollständige geometrische Darstellung des Fassadenkörpers. Diese Darstellung soll ein extrudierter Volumenkörper (*IfcSweptAreaSolid*) sein, der bei Bedarf zusätzlich mit CSG-Operationen (*IfcCsgSolid*) bearbeitet werden kann. Mit einer zweiten Geometrie, beispielsweise in Form eines *IfcGeometricSet*, kann für die spätere Ausführungsplanung das Raster der Verglasungsbauweise angegeben werden.
- 2) Die Glasfassade verfügt über keine geometrische Darstellung, sondern wird in Komponenten aufgelöst. Diese Komponenten können Verglasungsflächen (*IfcGlassFace*) und Bauelemente der Unterkonstruktion (*IfcBeam*, *IfcColumn*, *IfcStructuralBuildingElementsAssembly*), zum Beispiel für Fassadenträger, sein. *IfcGlassFacade* wird jedoch nicht in Bauteile aufgelöst. Diese Stufe der Dekomposition findet erst in der Domäne für den Glasbau statt.

Mit der zweiten Variante können komplexere Fassadenformen dargestellt werden. Der Detaillierungsgrad ist jedoch bei beiden Möglichkeiten derselbe, nämlich der zur Übersichtsplanung notwendige.

6.4.3.2 Bauelement: *IfcGlassFace*

Verglasungsflächen sind verglaste Komponenten innerhalb von Glasfassaden (*IfcGlassFacade*), Dächern (*IfcRoof*) oder Wänden (*IfcWallIsDecomposed*). Sie werden durch die Klasse *IfcGlassFace* dargestellt, die über dieselben Darstellungsmöglichkeiten verfügt, die unter Punkt 1) der Klasse *IfcGlassFacade* beschrieben sind. Die Darstellung soll neben dem Verglasungsraster eine vollständige Volumenkörpergeometrie für das Bauelement besitzen, und nicht, wie etwa der Name suggerieren könnte, lediglich eine Oberflächendarstellung. Verglasungsflächen dürfen nicht mit Verglasungsbauteilen verwechselt werden, die erst in der Glasbaudomäne definiert werden.

6.4.3.3 Anforderungen: *Pset_GlassFaceRequirements*

Pset_GlassFaceRequirements definiert Anforderungen gemäß Tabelle 6.2 für Verglasungsflächen (*IfcGlassFace*) und Glasfassaden (*IfcGlassFacade*). Für die vollständigen Definitionen siehe **Anhang B.2**.

<i>Pset_GlassFaceRequirements</i>	
<u>GlazingSupportType:</u>	Geforderte Bauart. Unterscheidungskriterium ist die Lagerung der Verglasung (Aufzählungstyp).
<u>GlazingType:</u>	Geforderte Art der Verglasung (Aufzählungstyp)
<u>SafetyRequirement:</u>	Besondere Sicherheitsanforderung (Aufzählungstyp)
<u>GlazingThickness</u>	Statisch erforderliche Glasdicke (reelle Zahl)
<u>FireRating:</u>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm (Zeichenkette)
<u>HeatRadiationResistance:</u>	Erforderlicher Widerstand gegen Hitzestrahlung mit Angabe der gültigen Norm (Zeichenkette)
<u>AcousticInsulation:</u>	Erforderliches Schalldämm-Maß in dB (reelle Zahl)
<u>ThermalTransmittanceCoefficient:</u>	Maximal erlaubter Wärmedurchgangskoeffizient (reelle Zahl)
<u>ThermalResistance:</u>	Erforderlicher Wärmedurchlaßwiderstand (reelle Zahl)
<u>RainImpermeability:</u>	Schlagregendichtigkeit mit Angabe der gültigen Norm (Zeichenkette)
<u>Colour:</u>	Geforderte Färbung oder Tönung (Zeichenkette)
<u>SpecialRequirements:</u>	Weitere Anforderungen in Textform (Zeichenkette)

Tabelle 6.2: *Pset_GlassFaceRequirements*

6.4.3.4 Beispiel

Das Beispiel in Abbildung 6.13 zeigt die Anwendung von *IfcGlassFace* und *Pset_GlassFaceRequirements*: Eine Wand wird in drei Komponenten aufgelöst, davon eine Verglasungsfläche (Instanz #5). Diese verfügt über zwei geometrische Darstellungen, nämlich die Volumenkörpergeometrie des eigentlichen Bauelementes

(#7) und einen Satz aus Linien, der das Raster der Verglasung darstellt (#8). In der Instanz #10 werden die Entwurfsanforderungen an die Verglasung gestellt: Die Verglasung soll mit Prebleistenhalterungen und Zwei-Scheiben-Isolierglas der Dicke 26 mm ausgeführt werden. Die übrigen Anforderungen von *Pset_GlassFaceRequirements* sind aus Platzgründen nicht im Beispiel aufgeführt. Die Verwendung der Klasse *IfcGlassFacade* ist analog zu der von *IfcGlassFace*, so daß kein eigenes Beispiel hierfür gezeigt wird.

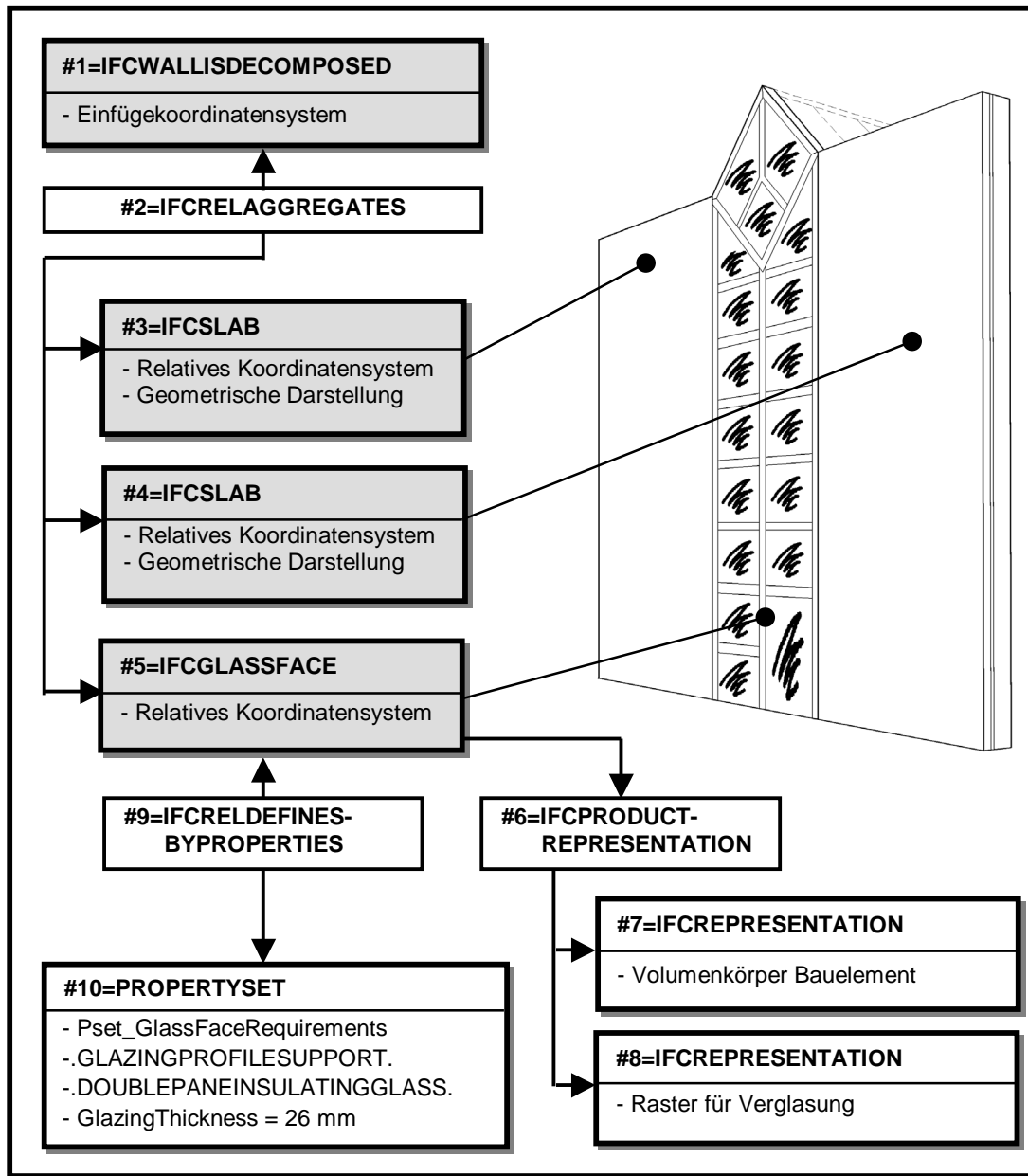


Abbildung 6.13: Verglasungsfläche innerhalb einer Wand

6.4.4 Fundamente: *IfcShallowFoundation*

Die Bauelemente des Stahlbetonbaus sollen im Rahmen dieser Arbeit lediglich vereinfacht dargestellt werden, das heißt ohne Darstellung der Bewehrung und ohne Definition von Anforderungen. Deshalb wird für Fundamente in Anlehnung an die Arbeit von *Weise* lediglich die neue Klasse *IfcShallowFoundation* eingeführt, die

alle Arten von Flachgründungen umfaßt [119]. Sie besitzt neben einem Einfügekoordinatensystem eine geometrische Darstellung vom Typ BREP (*IfcManifoldSolidBrep*). Außerdem wird dem Fundament eine Materialliste (*IfcMaterialList*) zugeordnet, in der der zu verwendende Beton und die Betonstahlsorte angegeben werden. Die Art der Gründung wird im Attribut *PredefinedType* festgelegt (Einzelfundament, Streifenfundament oder Bodenplatte).

6.4.5 Fensteranordnungen

6.4.5.1 Bauelement: *IfcWindowArrangement*

Mit der Klasse *IfcWindowArrangement* werden Fensteranordnungen dargestellt, also mehrere Einzelfenster, die zu einem Bauelement zusammengefaßt sind und zusammen eine einzige Wandöffnung ausfüllen. Die Klasse besitzt ein Einfügekoordinatensystem, jedoch weder eine Materialzuordnung noch eine eigene geometrische Darstellung. Statt dessen wird sie in einzelne Fenster (*IfcWindow*) und eventuell in Sonderteile (*IfcConstructionalElement*) aufgelöst. Letztere können zum Beispiel Pfosten und Stürze zwischen den einzelnen Fenstern sein. Über *IfcRelFillsElement* wird die Fensteranordnung in die Öffnung der betroffenen Wand eingesetzt.

Für Fensteranordnungen werden keine Anforderungen definiert. Diese können statt dessen an die Instanzen der Einzelfenster geknüpft werden.

6.4.5.2 Beispiel

Das Beispiel in Abbildung 6.14 verdeutlicht die Anwendung der Klasse *IfcWindowArrangement*. Es zeigt eine Wand (#1) mit einer Öffnung (#3), in die eine Fensteranordnung (#5) eingefügt ist. Die X-Z-Ebene des Einfügekoordinatensystems der Fensteranordnung ist als schraffierte Fläche skizziert. In dieses Koordinatensystem werden die Instanzen der Klasse *IfcWindow* eingefügt, von denen hier aus Platzgründen lediglich die zwei Instanzen #7 und #8 aufgeführt sind. Die Auflösung der Fensteranordnung erfolgt durch *IfcRelAggregates* (#6).

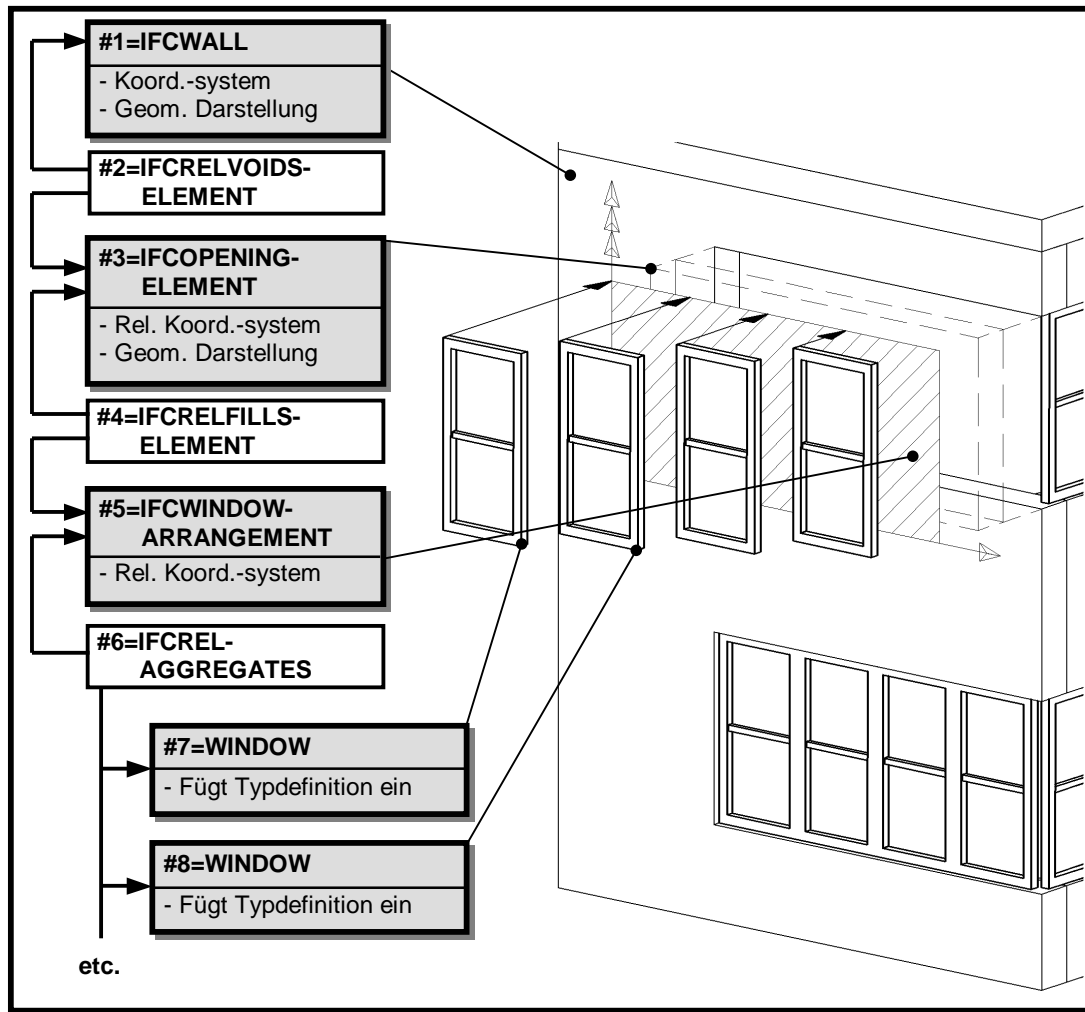


Abbildung 6.14: Fensteranordnung

6.4.6 Industrietore

6.4.6.1 Bauelement: *IfcDoorIndustrial*

Analog zur Modellierung von Fenstern und Türen in den IF2x wird die Darstellung von Industrietoren im Komplettbaumodell eingeführt: Die Klasse *IfcDoorIndustrial* fügt eine Typdefinition der Klasse *IfcDoorIndustrialStyle* in Wandöffnungen ein. *IfcDoorIndustrial* besitzt dazu ein Einfügekoordinatensystem, jedoch keine eigene geometrische Darstellung. Die Zuordnung der Typdefinition erfolgt durch *IfcRelDefinesByType*; über *IfcRelFillsElement* wird das Tor in eine Wandöffnung eingesetzt.

6.4.6.2 Typdefinition: *IfcDoorIndustrialStyle*

Die Klasse *IfcDoorIndustrialStyle* dient zur Typdefinition von Industrietoren. Sie verfügt über die Attribute gemäß Tabelle 6.3, wobei für die ausführlichen Typisierungen aus Platzgründen auf **Anhang A** verwiesen wird.

IfcDoorIndustrialStyle verfügt über keine geometrische Darstellung. Die Geometrien von Rahmen und Torflügeln werden stattdessen parametrisch durch die in den beiden folgenden Abschnitten definierten statischen Eigenschaftsklassen angegeben. Sie werden vom Attribut *HasProperties* der Oberklasse *IfcTypeObject* von *IfcDoor-*

IndustrialStyle aus referenziert. Es können lediglich rechteckige Torformen dargestellt werden, was aber im allgemeinen ausreichend ist.

<i>IfcDoorIndustrialStyle</i>	
OperationType:	Bewegungsart nach DIN12433 (Aufzählungstyp)
FrameConstructionType:	Rahmenbauweise (Aufzählungstyp)
LeafConstructionType:	Torblatt- oder Torsegmentbauweise (Aufzählungstyp)
OperationMethod:	Betätigungsart nach DIN12433 (Aufzählungstyp)
SurfaceArea:	Torfläche (reelle Zahl)

Tabelle 6.3: *IfcDoorIndustrialStyle*

6.4.6.3 Rahmengeometrie: *IfcDoorIndustrialFrameProperties*

Die Klasse *IfcDoorIndustrialFrameProperties* beschreibt die Form rechteckiger Torrahmen. Sie verfügt über keine vollständige geometrische Darstellung, sondern definiert lediglich die Parameter, aus denen Software-Applikationen eine Darstellung erzeugen können, siehe Abbildung 6.15. Alle Parameter beziehen sich auf das Koordinatensystem der einfügenden Instanz *IfcDoorIndustrial*.

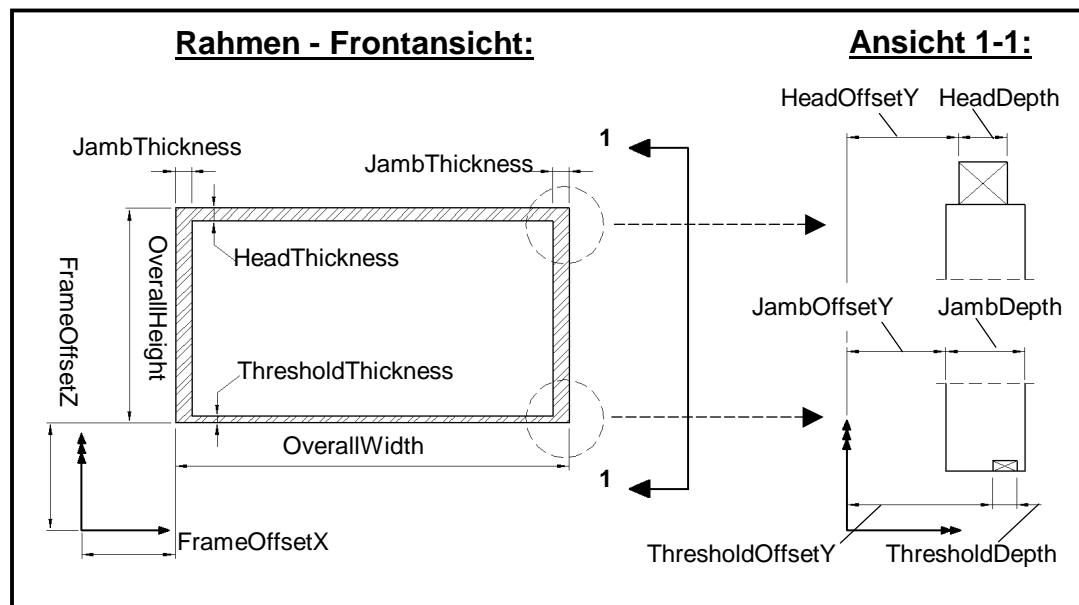


Abbildung 6.15: Parameter des Torrahmens

6.4.6.4 Torflügelgeometrie: *IfcDoorIndustrialLeafProperties*

Die Klasse *IfcDoorIndustrialLeafProperties* beschreibt keine einzelnen Torflügel oder -lamellen, sondern die Geometrie der Gesamtheit aller beweglichen Teile eines Tores im geschlossenen Zustand. Wie auch *IfcDoorIndustrialFrameProperties* verfügt sie dazu über keine vollständige geometrische Darstellung, sondern lediglich über die Parameter, aus denen Software-Applikationen eine Darstellung erzeugen können, siehe Abbildung 6.16. Alle Parameter beziehen sich auf das Koordinatensystem-

stem der einfügenden Instanz *IfcDoorIndustrial*. Durch ein zusätzliches Attribut *SegmentNumber* kann bei den Torarten „Falttor“ und „Schiebefalttor“ die Anzahl der Torflügel, und bei den Typen „Rundlauf-Schiebetor“ und „Sektionaltor“ die Anzahl der Segmente angegeben werden. Bei den übrigen Torarten hat das Attribut keine Bedeutung.

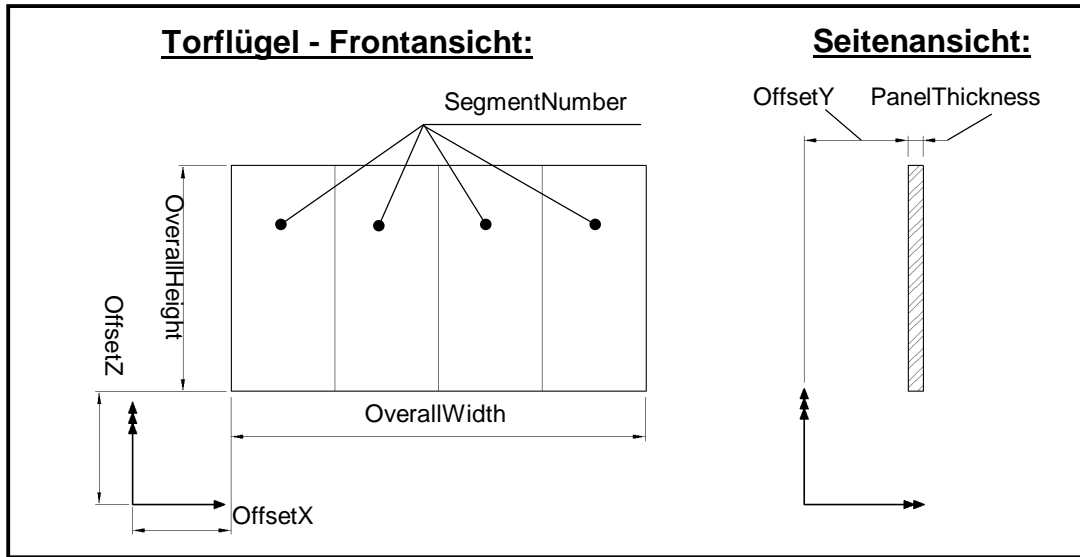


Abbildung 6.16: Parameter der Torflügel

6.4.6.5 Anforderungen: *Pset_DoorIndustrialRequirements*

In *Pset_DoorIndustrialRequirements* können Anforderungen für Typdefinitionen von Industrietoren gemäß Tabelle 6.4 festgelegt werden. Sie sind aus dem Grunde als Anforderungen und nicht als direkte Attribute in *IfcDoorIndustrialStyle* definiert, weil in dem hier erreichten Detaillierungsgrad nicht von tatsächlichen Eigenschaft gesprochen werden kann. Für die vollständigen Definitionen siehe **Anhang B.3**.

<i>Pset_DoorIndustrialRequirements</i>	
<u>SpecialPurpose:</u>	Besonderes Einsatzgebiet, zum Beispiel als Brandschutztor (Aufzählungstyp)
<u>SelfClosing:</u>	Selbstschließendes Tor (boole'scher Typ)
<u>RapidOperation:</u>	Schnellauftor (boole'scher Typ)
<u>FireRating:</u>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm (Zeichenkette)
<u>AcousticRating:</u>	Erforderliches Schalldämm-Maß in dB (reelle Zahl)
<u>ThermalTransmittanceCoefficient:</u>	Maximal erlaubter Wärmedurchgangskoeffizient (reelle Zahl)
<u>SpecialRequirements:</u>	Weitere Anforderungen in Textform (Zeichenkette)

Tabelle 6.4: *Pset_DoorIndustrialRequirements*

6.4.6.6 Beispiel

Das Beispiel in Abbildung 6.17 zeigt eine Wand (#1) mit einer Öffnung (#3), in die ein Industrietor eingefügt ist. Von der einfügenden Instanz des Tores (#5) ist nur die X-Z-Ebene des Koordinatensystems als schraffierte Fläche dargestellt, auf das sich die geometrischen Parameter der Eigenschaftsklassen (#8 und #9) beziehen. Das Attribut *SegmentNumber* in Instanz #8 definiert die Anzahl der Torflügel. Die semantischen Attribute in der Typdefinition (#7) beschreiben die grundlegenden Eigenschaften des Tores: Es handelt sich um ein Schiebefalttor mit Rahmen und Torblättern aus Stahl, das im Handbetrieb bewegt wird.

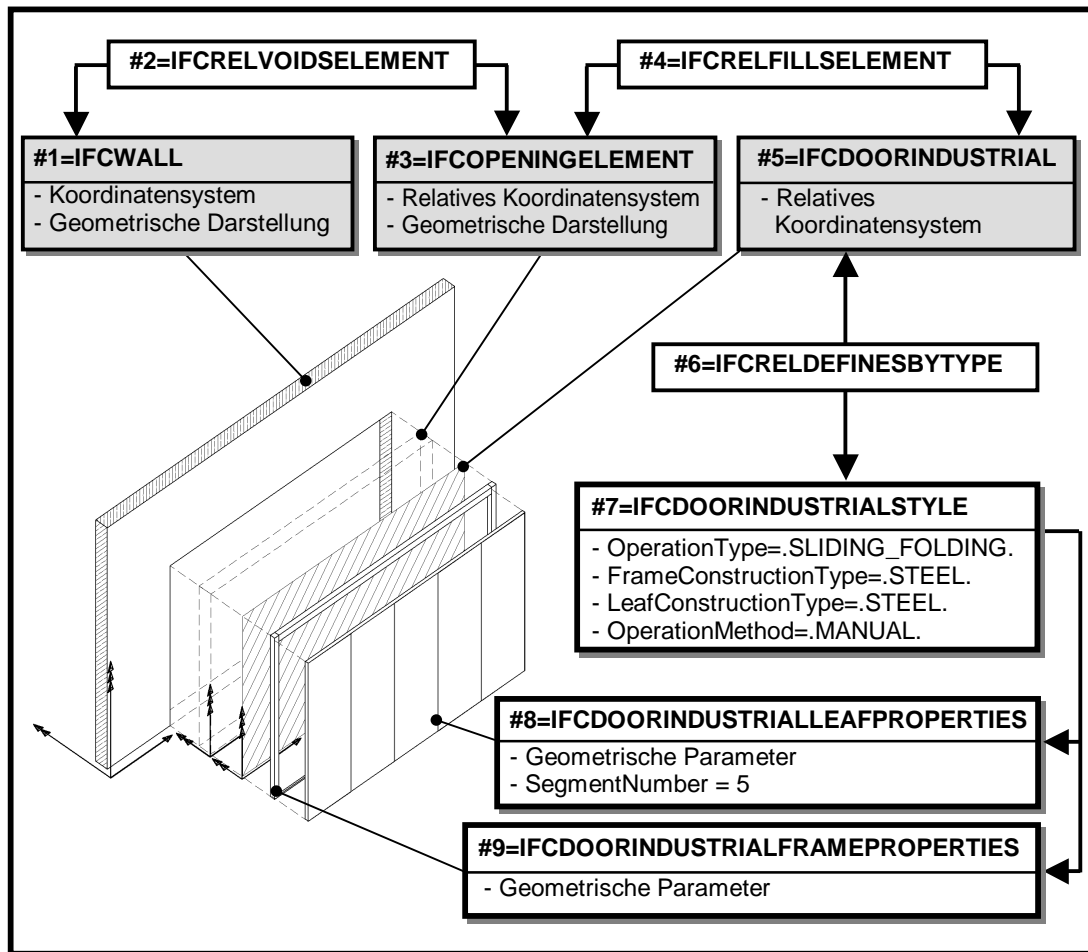


Abbildung 6.17: Beispiel Schiebefalttor

6.4.7 Dachbelichtungselemente

Die Modellierung von Dachbelichtungselementen erfolgt ähnlich wie die von Industrietoren mit Typdefinition und Typeinfügung. Die Geometrien von Dachbelichtungselementen sind jedoch vielfältiger, so daß an dieser Stelle eine explizite geometrische Darstellung für die Typdefinition sinnvoller ist als die parametrische Beschreibung über Eigenschaftsklassen.

6.4.7.1 Bauelement: *IfcRooflightElement*

Die Klasse *IfcRooflightElement* fügt Instanzen der Klasse *IfcRooflightElementStyle* (Typdefinition des Elementes) in Öffnungen von Dächern ein. Sie verfügt über ein

Einfügekoordinatensystem, jedoch keine eigene geometrische Darstellung. Die Einfügung eines Elementes in eine Öffnungen erfolgt durch *IfcRelFillsElement*.

6.4.7.2 Typdefinition: *IfcRooflightElementStyle*

Die Klasse *IfcRooflightElementStyle* dient zur Typdefinition von Dachbelichtungselementen. Sie verfügt über eine vollständige geometrische Darstellung, die als Block definiert ist. Die Detaillierung dieser Darstellung wird nicht vorgeschrieben, sie sollte jedoch möglichst vereinfacht sein.

Außerdem sind die semantischen Attribute gemäß Tabelle 6.5 definiert, wobei die ausführlichen Typisierungen aus Platzgründen nur in **Anhang A** aufgeführt sind.

<i>IfcRooflightElementStyle</i>	
<u>ElementType:</u>	Grundform des Belichtungselementes (Aufzählungstyp)
<u>CoverConstructionType:</u>	Bauweise des eigentlichen Lichtelementes (Aufzählungstyp)
<u>FrameConstructionType:</u>	Bauweise von Zarge oder Aufsetzkranz (Aufzählungstyp)
<u>OpeningMode:</u>	Starres oder zu öffnendes Element (Aufzählungstyp)
<u>OperationMethod:</u>	Betätigungsart (Aufzählungstyp)
<u>OpeningArea:</u>	Fläche der lichten Öffnung (reelle Zahl)

Tabelle 6.5: *IfcRooflightElementStyle*

6.4.7.3 Anforderungen: *Pset_RooflightElementRequirements*

In *Pset_RooflightElementRequirements* werden die Anforderungen gemäß Tabelle 6.6 für Dachbelichtungselemente spezifiziert. Sie sind aus dem Grunde als Anforderungen und nicht als direkte Attribute in *IfcRooflightElementStyle* definiert, weil in dem hier erreichten Detaillierungsgrad nicht von tatsächlichen Eigenschaft gesprochen werden kann. Die vollständigen Definitionen sind in **Anhang B.4** zu dieser Arbeit zu finden.

<i>Pset_RooflightElementRequirements</i>	
<u>SmokeAndHeatVentilation:</u>	Rauch- und Wärmeabzugsfunktion (boole'scher Typ)
<u>EffectiveOpeningArea:</u>	Erforderliche aerodynamisch wirksame Öffnungsfläche zum Rauch- und Wärmeabzug (reelle Zahl)
<u>FireRating:</u>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm (Zeichenkette)
<u>ThermalTransmittanceCoefficient:</u>	Maximal erlaubter Wärmedurchgangskoeffizient (reelle Zahl)
<u>SpecialRequirements:</u>	Weitere Anforderungen in Textform (Zeichenkette)

Tabelle 6.6: *Pset_RooflightElementRequirements*

6.4.7.4 Beispiel

Das in Abbildung 6.18 skizzierte Beispiel soll die Anwendung der Klassen *IfcRooflightElement* und *IfcRooflightElementStyle* verdeutlichen. In einer Dachfläche (#1) mit zwei Öffnungen (#3 und #6) sind zwei identische Lichtkuppeln eingesetzt. Dazu wird dieselbe Instanz zur Typdefinition (#10) durch die zwei Instanzen der Klasse *IfcRooflightElement* (#5 und #8) eingefügt. Die Typdefinition beinhaltet eine vereinfachte geometrische Darstellung als Blockdefinition sowie semantische Attribute, die besagen, daß es sich um eine Lichtkuppel aus Acrylglas mit PVC-Aufsetzkranz handelt.

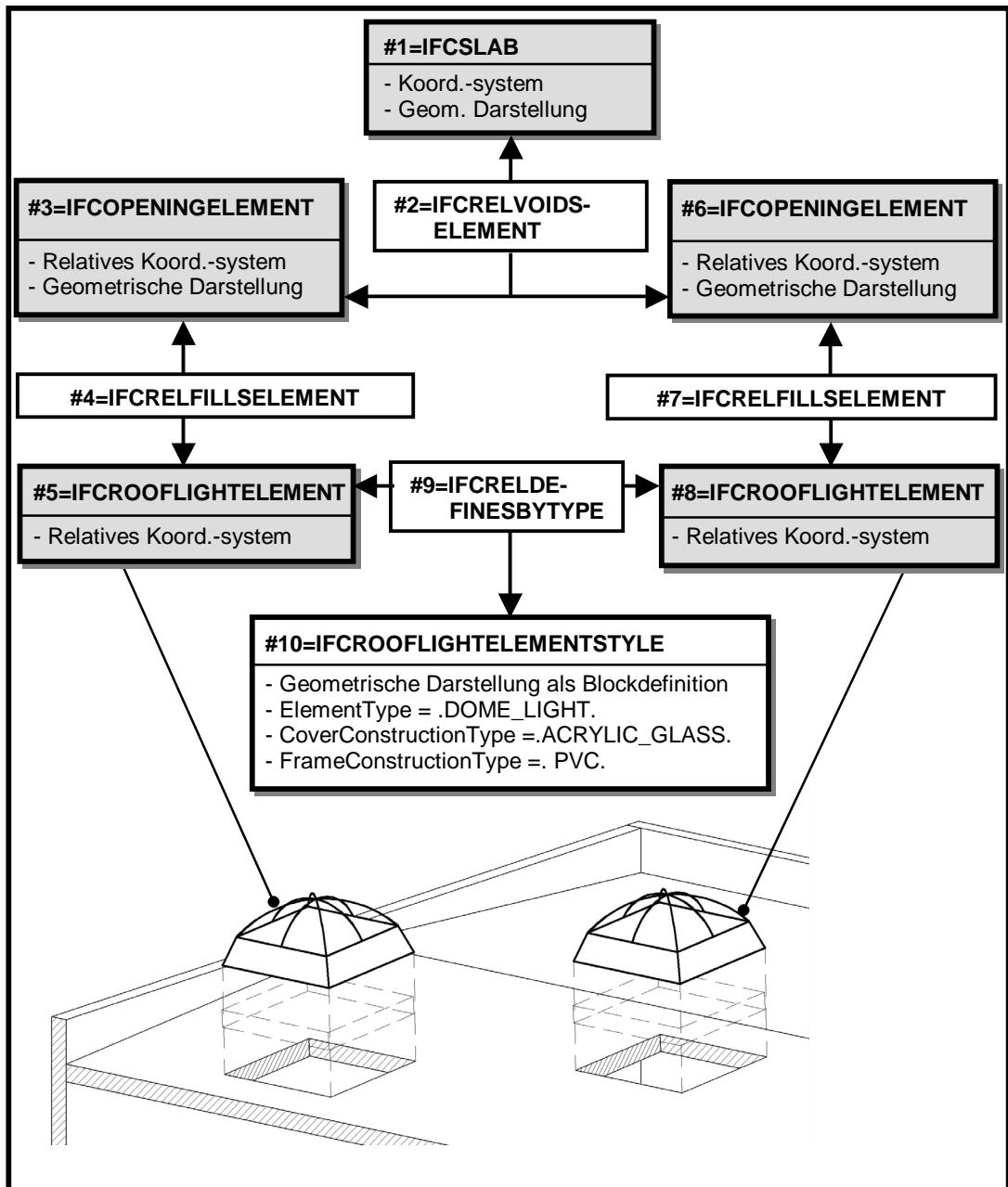


Abbildung 6.18: Beispiel Lichtkuppeln

6.4.8 RWA-Einrichtungen

Obwohl RWA-Geräte, wie zum Beispiel Brandlüfter, eher Ausrüstungsgegenstände als Bauelemente sind, werden sie in der Planung von Gebäuden genauso behandelt wie Dachbelichtungselemente. Deshalb werden sie diesen modelltechnisch gleichgestellt als Unterklassen von *IfcBuildingElement* definiert. Die Klassen *IfcSmokeAndHeatControlUnit* und *IfcSmokeAndHeatControlUnitStyle* sowie die Anforderung *Pset_SmokeAndHeatControlUnitRequirements* sind denen der Dachbelichtungselemente somit sehr ähnlich und werden aus Platzgründen hier nicht weiter erläutert. Die vollständigen Definitionen sind jedoch in den **Anhängen A und B** dieser Arbeit zu finden.

6.4.9 Anforderungen an Bauelemente der IFC2x

Die bislang definierten Anforderungen betreffen ausschließlich die im Komplettbaumodell neu definierten Bauelemente. Analog dazu werden Anforderungen auch für diejenigen Bauelemente definiert, die in den IFC2x bereits existieren. Dies geschieht durch *Pset_SlabRequirements* für Dach- und Wandscheiben und durch *Pset_BeamAndColumnRequirements* für Träger und Stützen des Tragwerkes. Aus Platzgründen sind diese lediglich in **Anhang B.6** und **B.7** aufgeführt. Die für Türen und Fenster dynamisch definierten Eigenschaften *Pset_DoorCommon* und *Pset_WindowCommon* der IFC2x decken sich inhaltlich weitgehend mit den Anforderungen des Komplettbaus, so daß auf eine Formulierung expliziter Anforderungen verzichtet werden kann.

6.5 Dach und Wand

6.5.1 Grundzüge der Darstellung

Dieser Abschnitt enthält die Klassendefinitionen der Domäne für Dach und Wand. In Kapitel 4 wurde festgelegt, daß dazu die Bauarten mit Trapez- und Kassettenprofilen sowie Sandwich- und Gasbetonelementen darzustellen sind. Die Darstellung muß die Detail- und Verlegeplanung abdecken – die Übersichtsplanung wurde ja bereits im gewerkeübergreifenden Modell behandelt. Die Detailplanung beinhaltet eine vollständige Darstellung des Schichtaufbaus der Bauelemente, während es für die Verlegeplanung der Auflösung der Bauelemente in Bauteile bedarf. Bauteile sind entweder raumbildende Verlegebauteile oder Zubehörteile. Zur Begrenzung der anfallenden Datenmenge werden Befestigungsmittel nicht explizit dargestellt.

Die Domäne ist auf die unmittelbaren Gewerke der Gebäudehülle beschränkt, so daß zum Beispiel Innenverkleidungen von Dächern und Wänden oder Kiesschüttungen nicht eingeschlossen sind. Es sei noch einmal darauf hingewiesen, daß es sich um ein räumliches Objektmodell des Gebäudes handelt. Die eigentlichen Verlegepläne im Sinne von Zeichnungen sind also nicht Teil dieser Darstellung, sondern müssen von einer entsprechenden Software-Applikation daraus abgeleitet werden.

Alle neuen Klassen werden im Schema *IFCOUTERSHELLDOMAIN* definiert, siehe Abbildung 6.19. Die Abbildung ist so aufgebaut, daß von unten nach oben die Detaillierung der Darstellung zunimmt, während von links nach rechts die Bauelemente immer weiter aufgelöst werden. Links unten sind also die einfachen Bauelemente zu finden, ganz rechts oben die Darstellung der Befestigungsmittel.

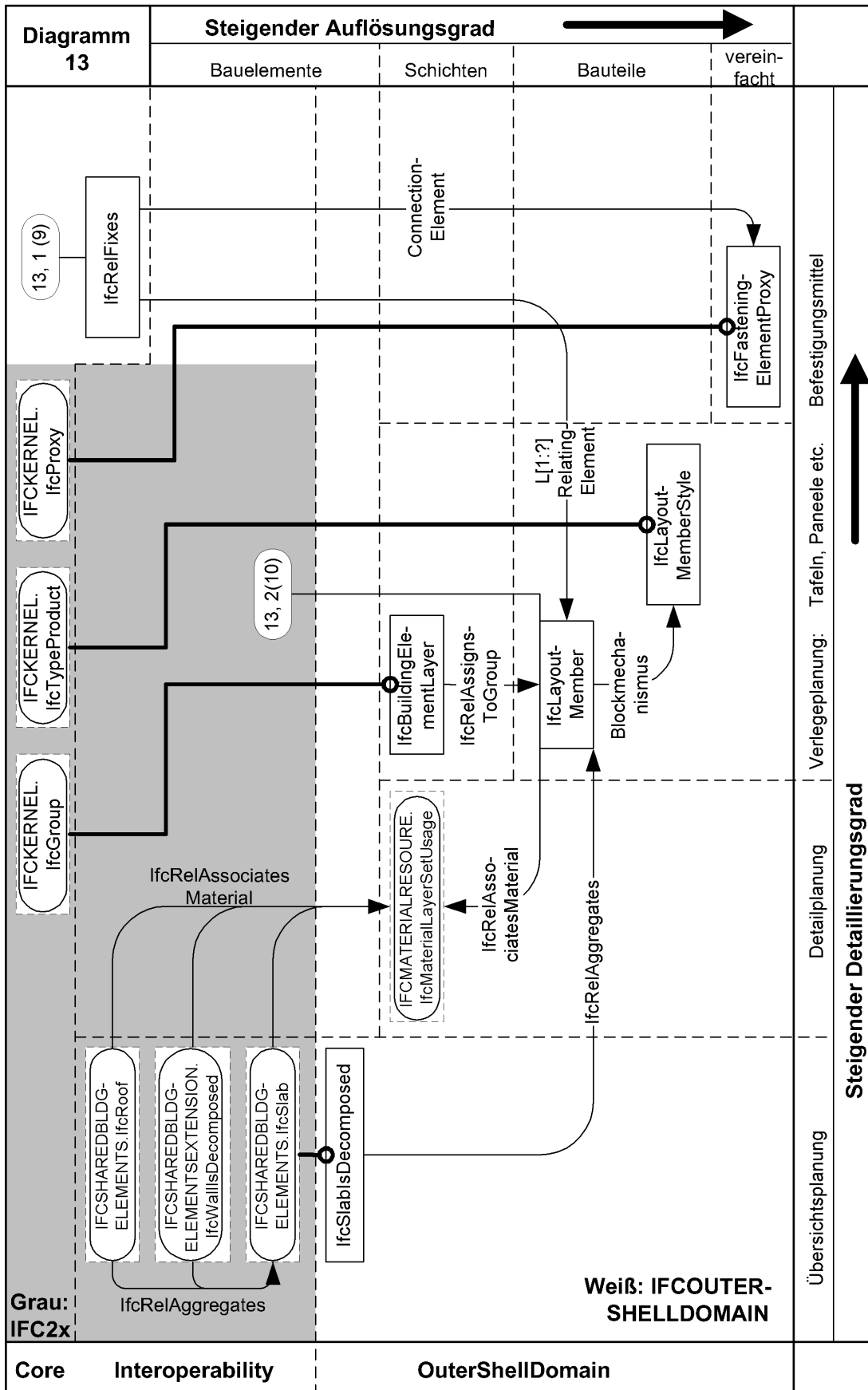


Abbildung 6.19: Domäne IFCOUTER-SHELLDOMAIN (EXPRESS-G)

6.5.2 Detailplanung: Schichtaufbau und Bauprodukte

Zur Darstellung von Dach und Wand im Detaillierungsgrad der Detailplanung werden weiterhin die Bauelementklassen des gewerkeübergreifenden Modells verwendet: *IfcRoof*, *IfcWallIsDecomposed* und *IfcSlab*. Es wird jedoch vorgeschrieben, daß ihr Aufbau vollständig in Form einer geschichteten Bauproduktdefinition (*IfcMaterialLayerSetUsage*) dargestellt sein muß, der sogenannten *Leitschichtung*. Für einschichtige Bauarten, zum Beispiel mit Sandwichelementen, wird nur eine einzige Schicht definiert.

Der Gebrauch der entsprechenden Klassen aus der erweiterten Material-Ressource wurde bereits in Abschnitt 6.3.2 erläutert und wird hier nicht wiederholt. Es ist die Aufgabe des Konstrukteurs, den Schichtaufbau mit Unterstützung seiner Software in Einklang mit Anforderungen zu bringen, die im Stadium der Übersichtsplanung gegebenenfalls in Form von *Pset_RoofAndWallRequirements* oder *Pset_SlabRequirements* an das Bauelement gestellt wurden.

6.5.3 Verlegeplanung

6.5.3.1 Auflösung der Bauelemente in Bauteile: *IfcSlabIsDecomposed*

Die Auflösung der globalen Bauelemente in Bauteile erfolgt nach dem in Abschnitt 6.3.5 erläuterten Prinzip. Dazu wird die neue Klasse *IfcSlabIsDecomposed* definiert. Die Auflösung darf mit keiner anderen Bauelementklasse geschehen. Das bedeutet, daß Dächer oder Wände (*IfcRoof*, *IfcWallIsDecomposed*) ihrerseits zuvor in Bauelemente der Klasse *IfcSlabIsDecomposed* aufgelöst werden müssen. Abbildung 6.20 zeigt das vollständige Dekompositionsschema von Bauelementen für Dach und Wand. Da die Befestigungsmittel nicht explizit dargestellt werden, sind sie in der Abbildung nicht aufgeführt.

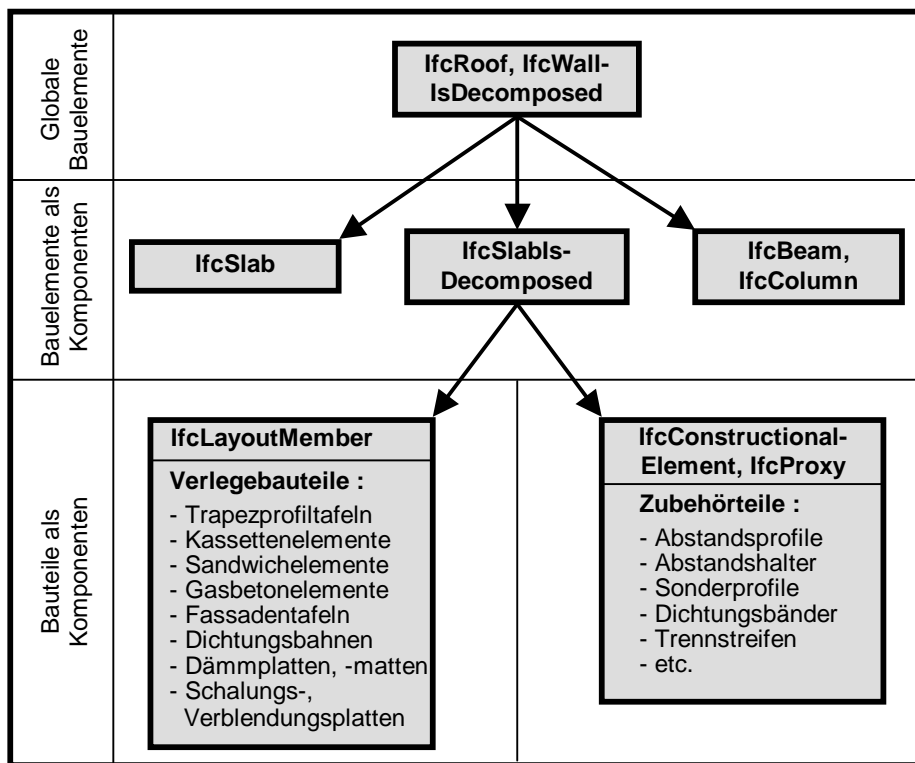


Abbildung 6.20: Dekompositionsschema für Dächer und Wände

Für die Klasse *IfcSlabsDecomposed* sind folgende Eigenschaften definiert.

- 1) Eine Instanz von *IfcSlabsDecomposed* verfügt entweder über eine eigene Leitschichtung, oder es gilt implizit die Leitschichtung der übergeordneten Instanz *IfcRoof* oder *IfcWallIsDecomposed*.
- 2) Eine Positionsnummer kann im Attribut *Tag* der Oberklasse *IfcElement* angegeben werden.
- 3) *IfcSlabsDecomposed* verfügt über ein Einfügekoordinatensystem.
- 4) *IfcSlabsDecomposed* verfügt über keine geometrische Darstellung. Statt dessen wird sie in Verlege- oder Zubehörbauteile aufgelöst. Es ist wiederum die Aufgabe des Konstrukteurs, die Art und Anordnung der Bauteile in Einklang mit der Leitschichtung zu bringen. Um die Darstellung zu vereinfachen, müssen nicht alle Schichten explizit mit Bauteilen dargestellt werden. Für Dämm- oder Dichtungsschichten beispielsweise ist es in der Regel ausreichend, wenn diese lediglich in der Leitschichtung aufgeführt werden.

6.5.3.2 Verlegebauteile: *IfcLayoutMember*

Um das anfallende Datenvolumen zu begrenzen, wird der Blockmechanismus zur Darstellung von Verlegebauteilen angewandt. Außerdem entspricht er der Ingenieurvorstellung von Teilebeschreibung (Position) und Einfügung dieser Teile. Dabei müssen identische Bauteile mit derselben Instanz zur Typdefinition dargestellt werden.

Die Klasse *IfcLayoutMember* fügt Typdefinitionen (*IfcLayoutMemberStyle*) innerhalb des Koordinatensystem der übergeordneten Instanz *IfcSlabsDecomposed* ein. *IfcLayoutMember* verfügt dazu über ein relatives Koordinatensystem, jedoch keine geometrische Darstellung. Die komplizierte Beziehung zwischen einfügender Instanz und Blockdefinition wird in den Abbildungen dieses Abschnittes der Übersichtlichkeit halber vereinfacht als „Blockmechanismus“ bezeichnet.

An dieser Stelle wäre es eigentlich sinnvoll, Verlegebauteile aus Trapezprofilen in ihrer Breite stauchen und dehnen zu können. Dies ist mit den Mitteln der IFC2x jedoch nicht möglich, da der Transformationsoperator innerhalb der Blockeinfügung nur einheitliche Skalierungen für alle drei Koordinatenrichtungen zuläßt.

6.5.3.3 Typdefinition für Verlegebauteile: *IfcLayoutMemberStyle*

Die Klasse *IfcLayoutMemberStyle* dient zur Typdefinition von Verlegebauteilen, wozu sie über folgende, von ihrer Oberklasse geerbte Eigenschaften verfügt:

- 1) Über *IfcRelAssociatesMaterial* wird ein Bauprodukt zugeordnet. Sofern dieses Produkt Bestandteil der Leitschichtung ist, soll die dort verwendete Instanz referenziert werden. Dieser Fall wird im Beispiel in Abschnitt 6.5.4b) gezeigt.
- 2) *IfcLayoutMemberStyle* verfügt immer über eine vollständige geometrische Darstellung, die als Blockdefinition „gekapselt“ ist. Als Standardgeometrien für unbearbeitete Bauteile kommen *IfcSweptAreaSolid* (in Längsrichtung gerade Bauteile) und *IfcSurfaceCurveSweptAreaSolid* (in Längsrichtung gekrümmte Bauteile) zur Anwendung. Es können alle in Abschnitt 6.3.1 definierten Profilformen dazu verwendet werden. Darauf basierend können CSG-Geometrien (*IfcCsgSolid*) für bearbeitete Bauteile erzeugt werden, also beispielsweise für Verlegetafeln, die nicht rechteckig zugeschnitten sind, oder die über Innenausschnitte verfügen. Vereinfacht darf das Bauteil auch ohne Profilierung lediglich über seinen Hüllkörper dargestellt werden.

- 3) Eine Positionsnummer wird im Attribut *Tag* der Oberklasse *IfcTypeProduct* angegeben.

Außerdem ist das folgende Attribut definiert.

<i>IfcLayoutMemberStyle</i>	
<u>MemberType:</u>	Art des Bauteils (Aufzählungstyp), siehe Anhang A .

Tabelle 6.7: *IfcLayoutMemberStyle*

6.5.3.4 Gruppierung der Bauteile zu Schichten: *IfcBuildingElementLayer*

Mit der Klasse *IfcBuildingElementLayer* werden Verlegebauteile zu Schichten gruppiert. Diese „nachträgliche“ Gruppierung ist notwendig und nicht etwa redundant zur Leitschichtung, da zum einen nicht zwangsläufig alle Bauteile einer Schicht aus demselben Bauprodukt bestehen, und zum anderen die Bauart und die Funktion einer Schicht nicht eindeutig aus seinen Bauprodukten allein ableitbar sind. Über *IfcRelAssignsToGroup* werden die Bauteile der Schicht zugeordnet.

IfcBuildingElementLayer verfügt über die Attribute nach Tabelle 6.8, wobei die ausführlichen Typisierungen aus Platzgründen lediglich in **Anhang A** aufgeführt sind.

<i>IfcBuildingElementLayer</i>	
<u>Tag:</u>	Positionsnummer (Zeichenkette)
<u>Description:</u>	Beschreibung der Schicht in Textform (Zeichenkette)
<u>LayerNumber:</u>	Schichtnummer bezogen auf die Leitschichtung (ganze Zahl)
<u>LayerConstructionType:</u>	Bauart der Schicht (Aufzählungstyp)
<u>LayerFunctionType:</u>	Funktion der Schicht innerhalb des Bauelementes (Aufzählungstyp)

Tabelle 6.8: *IfcBuildingElementLayer*

6.5.3.5 Befestigung der Verlegebauteile: *IfcFasteningElementProxy*

Die Befestigungen von Verlegebauteilen werden in zweierlei Hinsicht vereinfacht dargestellt: Erstens wird auf die aufwendige Darstellung von wirklichen Verbindungen zugunsten von Befestigungen gemäß Abschnitt 6.3.3 (*IfcRelFixes*, *IfcRelFixesSemantic*, *IfcRelFixesGeometric*) verzichtet. Wird dabei das Anordnungsmuster der Befestigungsmittel lediglich in Textform angegeben, sollte dies in einer für den Ingenieur und Zeichner verständlichen Form geschehen, zum Beispiel „Befestigung in jeder Rippe und über jedem Binder“. Die Zahl der dazu verwendeten Befestigungsmittel wird im Attribut *Quantity* von *IfcRelFixesSemantic* angegeben. Befestigungsmuster können beliebig überlagert werden. Es ist die Aufgabe des Konstrukteurs, die zu befestigenden Bauteile, die implizierte Unterkonstruktion sowie die Art und Anordnung der Befestigungsmittel sinnvoll zu kombinieren.

Die zweite Vereinfachung besteht in der Darstellung der Befestigungsmittel durch die Klasse *IfcFasteningElementProxy*, die über die folgenden, von ihrer Oberklasse *IfcProxy* geerbten Eigenschaften verfügt.

- 1) *IfcFasteningElementProxy* hat keine geometrische Darstellung.
- 2) Das Befestigungsmittel wird durch eine standardisierte Bezeichnung im Attribut *Tag* beschrieben. Dies ist notwendig, wenn Befestigungsmittel ohne bestimmte Bauprodukte dargestellt werden sollen. Ein Entwurf für eine entsprechende Standardisierung ist im **Anhang C** zu dieser Arbeit zu finden.
- 3) Zusätzlich zu dieser Bezeichnung kann ein Bauprodukt für das Befestigungsmittel durch *IfcProductSpecification* definiert und über *IfcRelAssociatesMaterial* zugeordnet werden.

Außerdem sind die folgenden Attribute gemäß Tabelle 6.9 definiert. Für die vollständigen Definitionen siehe **Anhang A** zu dieser Arbeit.

<i>IfcFasteningElementProxy</i>	
<u>FasteningElementType:</u>	Art des Befestigungsmittels (Aufzählungstyp)
<u>Specification:</u>	Beschreibung des Befestigungsmittels in Textform (Zeichenkette)
<u>PositionNumber:</u>	Positionsnummer (Zeichenkette)

Tabelle 6.9: *IfcFasteningElementProxy*

6.5.3.6 Palettierung

Die Verlegebauteile werden zum Transport auf Paletten verpackt. Die optimale, montagerechte Zuordnung der Bauteile zu den Paletten erfolgt in der Regel durch den Konstrukteur beim Erstellen der Verlegeplanung im CAD-System. Aus diesem Grunde darf eine entsprechende Klasse zur Palettierung nicht etwa in der Baubetriebsdomäne der IFC2x (*IFCCONSTRUCTIONMANAGEMENTDOMAIN*) angesiedelt sein. Statt dessen wird auf die in den IFC bereits vorhandene Klasse *IfcGroup* zurückgegriffen, die domänenübergreifend angewandt werden kann. Die Bauteile werden ihr über *IfcRelAssignsToGroup* zugeordnet. Im Attribut *Description* wird eine Beschreibung der Palette angegeben, und im Attribut *ObjectType* muß die Zeichenkette „Package“ stehen.

6.5.4 Beispiel: Kassettenwand

Im folgenden wird das Beispiel der Wand aus Abschnitt 6.4.2.3 bis zum Detaillierungsgrad der Verlegeplanung weiterentwickelt. Aus Platzgründen wird lediglich eine der drei Wandscheiben behandelt (#3).

a) In Bauteile aufgelöste Wand: *IfcSlabsDecomposed*

Abbildung 6.21 zeigt die Auflösung der Wandscheibe in Bauteile. Dazu wurde die ursprüngliche Instanz #3 (*IfcSlab*) durch eine Instanz der Klasse *IfcSlabsDecomposed* ersetzt. Gemäß der im ursprünglichen Beispiel aufgestellten Anforderung, daß die Wand in Kassettenbauart ausgeführt werden soll (#8=*Pset_RoofAndWallRequirements*), hat sie eine entsprechende Leitschichtung (#11=*IfcMaterialLayer*

Set) mit Angabe von Bauprodukten (*IfcProductSpecification*) erhalten. Aus Platzgründen sind nur die ersten zwei der insgesamt vier Schichten aufgeführt: Es fehlen die thermische Trennung zwischen den Kassettenstegen und die Trapezprofildeckschicht. In der Instanz #14 für die Wärmedämmschicht ist die Dicke Null angegeben, da die Dämmplatten innerhalb der Kassettenstege liegen und somit nicht zur Gesamtdicke des Bauelementes gerechnet werden dürfen.

Die neue Instanz #3 ist in Verlegebauteile aufgelöst, von denen stellvertretend für jede Schicht die Instanzen #106 (Kassetteprofil), #211 (Dämmung) und #331 (Trapezprofil) aufgeführt sind. Auf die explizite Darstellung der thermischen Trennstreifen kann vereinfachend verzichtet werden. Das Beispiel zeigt außerdem die Instanz #401 (*IfcConstructionalElement*), die, stellvertretend für die zahlreichen Zubehörteile der Wand, eine Attikakappe darstellt.

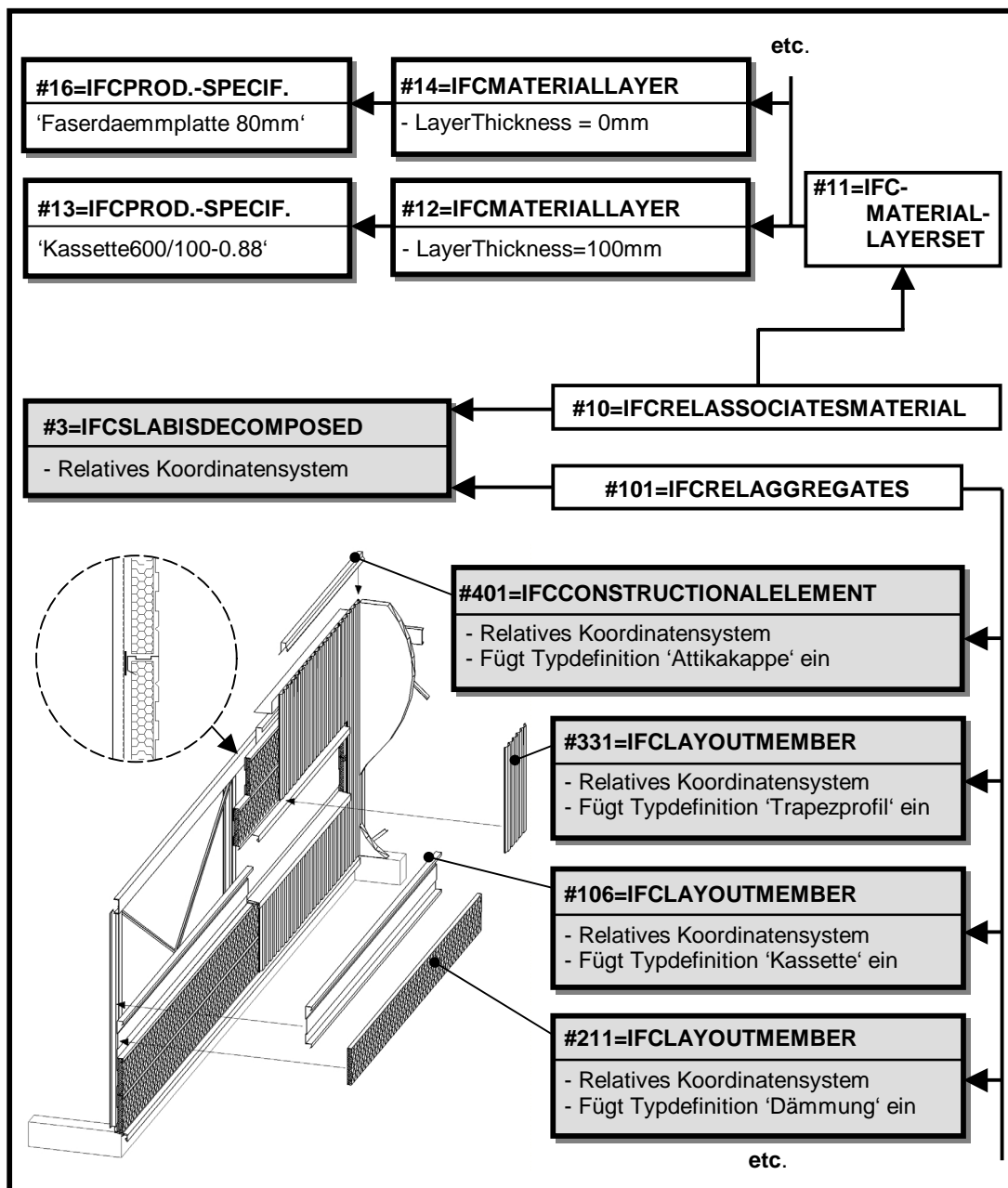


Abbildung 6.21: In Bauteile aufgelöste Wand

b) Verlegen der Kassettenelemente: *IfcLayoutMember*, *IfcLayoutMemberStyle*

In Abbildung 6.22 sind alle Verlegebauteile der Kassettenschicht aufgeführt. In der Austauschdatei werden sie durch die Instanzen #101 bis #108 (*IfcLayoutMember*) dargestellt, die die Typdefinitionen der Klasse *IfcLayoutMemberStyle* einfügen. Es existieren drei verschiedene Typen, die sich lediglich durch ihre Länge unterscheiden. Die Typdefinitionen benutzen jeweils ein in der Leitschichtung des übergeordneten Bauelementes bereits definiertes Bauprodukt (#13, siehe Abbildung 6.21). Die Zuordnung von Typdefinitionen zu den einfügenden Instanzen ist vereinfacht durch die gestrichelten Pfeile angedeutet.

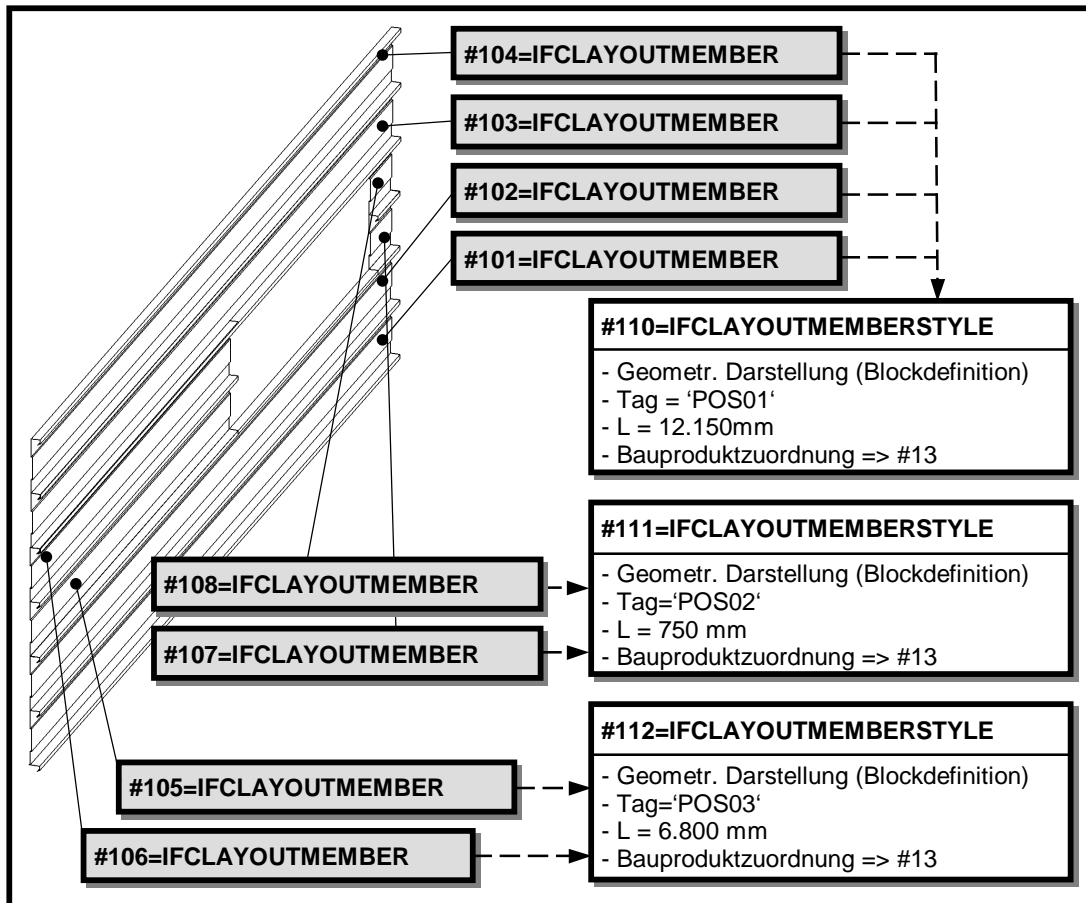


Abbildung 6.22: Verlegen der Kassettenelemente

c) Schichtzuordnung und Befestigung: *IfcBuildingElementLayer* und *IfcFasteningElementProxy*

In Abbildung 6.23 werden die Verlegebauteile schließlich zu einer Schicht gruppiert, die durch die Instanz #114 der Klasse *IfcBuildingElementLayer* dargestellt ist. Sie entspricht der Schicht 1 in der Leitschichtung, hat die Bauart 'Kassettprofil' und dient zur Lastabtragung innerhalb der Wand.

Durch die Instanz #116 wird eine Bohrschraube mit standardisierter Bezeichnung dargestellt, die den Verlegebauteilen über die Instanz #115 zugeordnet wird. Das Anordnungsmuster der Befestigungsmittel wird in Textform definiert; es werden 36 Schrauben benötigt.

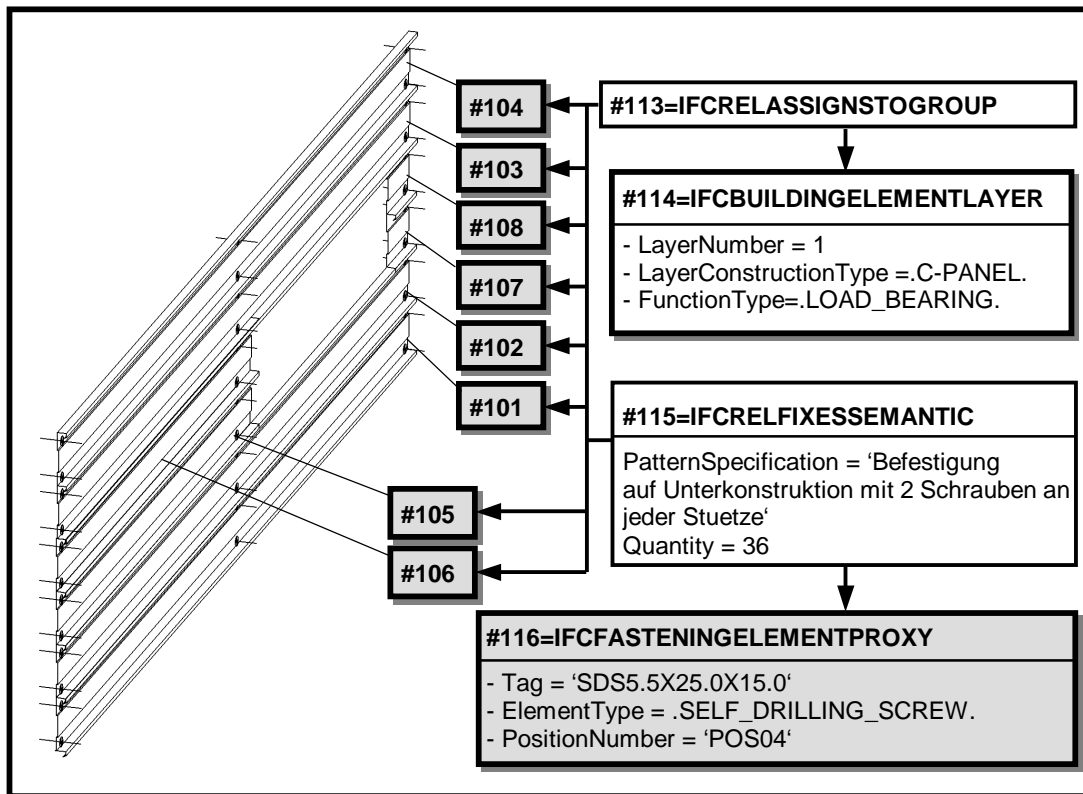


Abbildung 6.23: Schichtzuordnung und Befestigung

6.6 Dach- und Wandöffnungen

6.6.1 Allgemeines

In diesem Abschnitt wird das Domänenmodell für die Konstruktion der Öffnungen in Dach- und Wand für das Schema *IFCOPENINGCONSTRUCTIONDOMAIN* definiert. Die in Kapitel 4 geforderte Übersichtsplanung ist bereits durch das gewerkeübergreifende Modell abgedeckt, so daß an dieser Stelle lediglich der Detaillierungsgrad der Ausführungsplanung darzustellen ist. Im Gegensatz zu Dach und Wand würde eine Auflösung der Bauelemente in Bauteile aufgrund der Vielzahl der Komponenten, aus denen die Öffnungselemente bestehen, eine ausufernde Datenmenge bewirken. Anstelle der Auflösung wird deshalb die vereinfachte Darstellung mit parametrischen Eigenschaftsklassen aus dem gewerkeübergreifenden Modell übernommen. Die dort vorhandene Typisierung ist für die Bauelemente Türen, Fenster, Industrietore und RWA-Einrichtungen bereits ausreichend, während sie für die größere Formenvielfalt von Dachbelichtungselementen erweitert werden muß. Abbildung 6.24 zeigt die entsprechenden Klassen zur Typdefinition für die Dachbelichtungselemente.

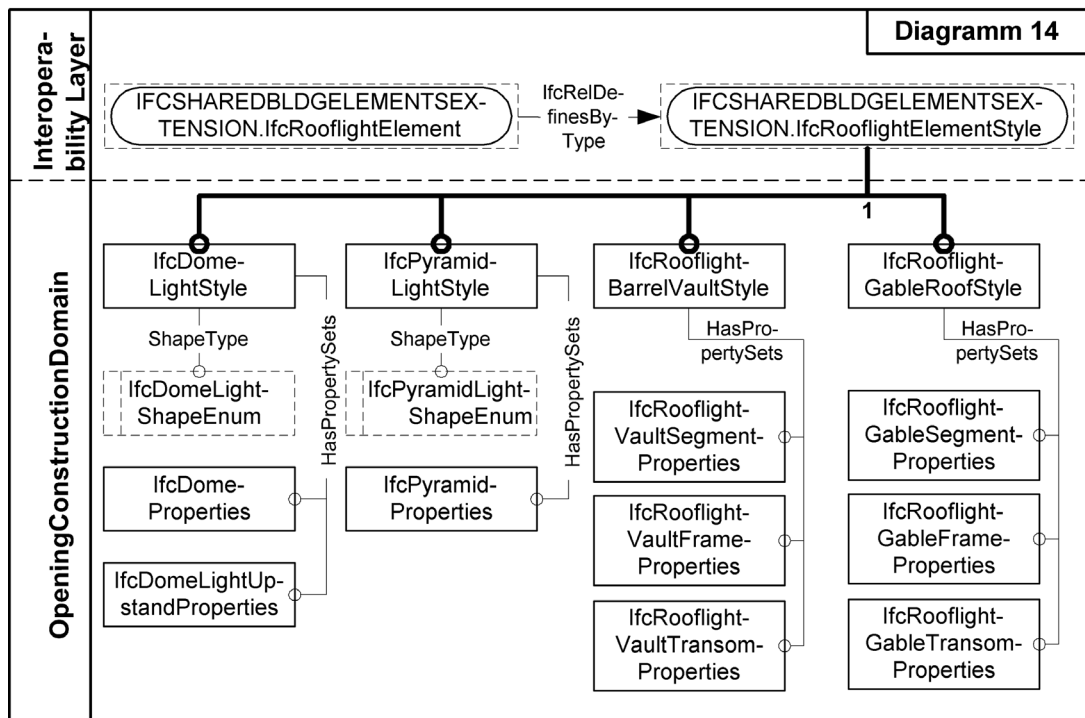


Abbildung 6.24: Typdefinitionen für Dachbelichtungselemente (EXPRESS-G)

Trotz dieser vergleichsweise einfachen Darstellung soll es möglich sein, bei Bedarf eine vollständige Auflistung der verwendeten Bauprodukte einschließlich Zubehör angeben zu können. Da die einzelnen Bestandteile der Elemente jedoch nicht explizit dargestellt werden, müssen die verwendeten Bauprodukte dem „übergeordneten“ Element zugeordnet werden, also der Typdefinition. Die in Abschnitt 6.3.2 beschriebene Darstellung ist hierfür ungeeignet, weil sie es nicht erlaubt, die Stückzahlen der jeweils verwendeten Produkte anzugeben. Aus diesem Grunde erhalten die Typdefinitionen der Öffnungselemente für jedes ihrer Hauptbestandteile ein Listenpaar, in dem jeweils die verwendeten Bauprodukte und die davon verwendeten Stückzahlen angegeben werden können. Aus Platzgründen sind diese Listen nicht in Abbildung 6.24 eingezeichnet. Das Prinzip wird jedoch im nachfolgenden Abschnitt erläutert.

Ebenfalls aus Platzgründen wird von den neuen Klassen lediglich der Typ „Lichtkuppel“ erläutert. Die anderen Dachbelichtungselemente „Lichtpyramide“ (*IfcPyramidLightStyle*), „Tonnengewöbelichtband“ (*IfcRooflightBarrelVaultStyle*) und „Satteldachlichtband“ (*IfcRooflightGableRoofStyle*) sind vollkommen analog dazu. Ihre formale Definition ist im **Anhang A** dieser Arbeit zu finden.

6.6.2 Typdefinition für Lichtkuppeln: *IfcDomeLightStyle*

Die Klasse *IfcDomeLightStyle* dient zur Typdefinition von Lichtkuppel-elementen. Wie ihre Oberklasse *IfcRooflightElementStyle* kann sie von beliebig vielen Instanzen *IfcRooflightElement* eingefügt werden. Sie verfügt über die Attribute gemäß Tabelle 6.10. Für jeden der Hauptbestandteile des Elementes (im Fall der Lichtkuppel sind dies das eigentliche Lichtkuppel-element, gegebenenfalls ein Aufsetzkranz, ein Antrieb zum Öffnen und sonstiges Zubehör) existiert ein *indexparalleles* Listenpaar. Das bedeutet, daß beide Listen die gleiche Länge haben. In der ersten Liste werden jeweils die Bauprodukte in Form von Instanzen der Klasse *IfcProductSpecification* aus dem Schema *IFCMATERIALRESOURCEEXTENSION* referenziert, und in der

zweiten Liste werden die davon verwendeten Stückzahlen angegeben. Sofern ein Hauptbestandteil nicht vorhanden ist, zum Beispiel bei einer Lichtkuppel ohne Aufsetzkranz, wird das entsprechende Listenpaar weggelassen.

<i>IfcDomeLightStyle</i>	
<u>ShapeType:</u>	Grundrißform: kreisförmig oder rechteckig (Aufzählungstyp)
<u>DomeProducts:</u>	Indexparalleles Listenpaar für Produkte der eigentlichen Kuppel
<u>DomeProductNumbers:</u>	
<u>UpstandProducts:</u>	Indexparalleles Listenpaar für Produkte des Aufsetzkranzes
<u>UpstandProductNumbers:</u>	
<u>DriveProducts:</u>	Indexparalleles Listenpaar für Produkte des Antriebs
<u>DriveProductNumbers:</u>	
<u>EquipmentProducts:</u>	Indexparalleles Listenpaar für Zubehörprodukte
<u>EquipmentProductNumbers:</u>	

Tabelle 6.10: *IfcDomeLightStyle*

IfcDomeLightStyle verfügt über keine geometrische Darstellung. Die Formen der Hauptbestandteile, sofern vorhanden, werden statt dessen über die Eigenschaftsklassen *IfcDomeProperties* und *IfcDomeLightUpstandProperties* gemäß Abbildung 6.25 beschrieben.

6.6.3 Geometrie der Kuppel: *IfcDomeProperties*

Die Klasse *IfcDomeProperties* beschreibt die Form der eigentlichen Lichtkuppel. Sie verfügt über keine wirkliche geometrische Darstellung, sondern definiert die Parameter, aus denen entsprechende Software-Applikationen eine Darstellung erzeugen können, siehe Abbildung 6.25. Alle Parameter beziehen sich auf das in der Abbildung eingezeichnete Koordinatensystem der einfügenden Instanz *IfcRooflightElement*.

6.6.4 Geometrie des Aufsetzkranzes: *IfcDomeLightUpstandProperties*

Analog zur Darstellung der Kuppel werden in der Klasse *IfcDomeLightUpstandProperties* die geometrischen Parameter von Lichtkuppelaufsetzkranzen angegeben, siehe Abbildung 6.25. Die Parameter beziehen sich ebenfalls auf das Koordinatensystem der einfügenden Instanz *IfcRooflightElement*.

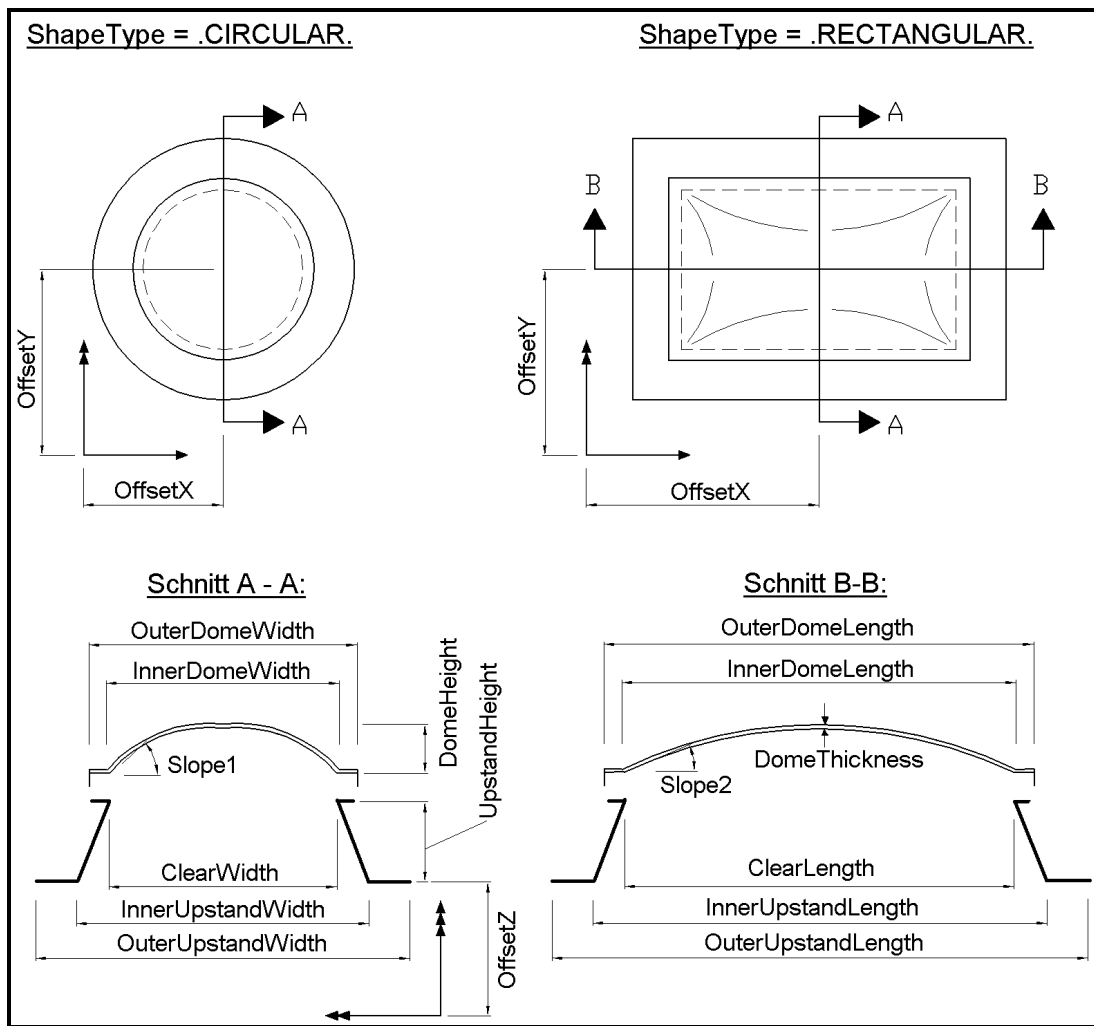


Abbildung 6.25: Geometrische Parameter von Lichtkuppeln

6.6.5 Beispiel

Das in Abbildung 6.26 skizzierte Beispiel verdeutlicht die Anwendung der Klasse *IfcDomeLightStyle* zur Typdefinition einer Lichtkuppel mit Bauprodukten. In der Instanz #10 selbst sind Informationen über den Elementtyp, die Bauart von Kuppel und Aufsetzkranz, die Betätigungsart und die Grundrißform angegeben. Die beiden Instanzen #11 und #12 spezifizieren die für Kuppel und Aufsetzkranz verwendeten Bauprodukte. Sie werden in den Listen *DomeProducts* und *UpstandProducts* aus Instanz #10 referenziert. Die Listen *DomeProductNumbers* und *UpstandProductNumbers* geben an, daß jeweils nur ein einziges dieser Produkte verwendet wird. Unabhängig davon werden die geometrischen Parameter der Kuppel und des Aufsetzkranzes in den Instanzen #13 und #14 angegeben. Die Einfügung dieser Typdefinition ist im Beispiel nicht dargestellt, da dies genauso erfolgt, wie es in Abschnitt 6.4.7 im gewerkeübergreifenden Modell bereits gezeigt wurde.

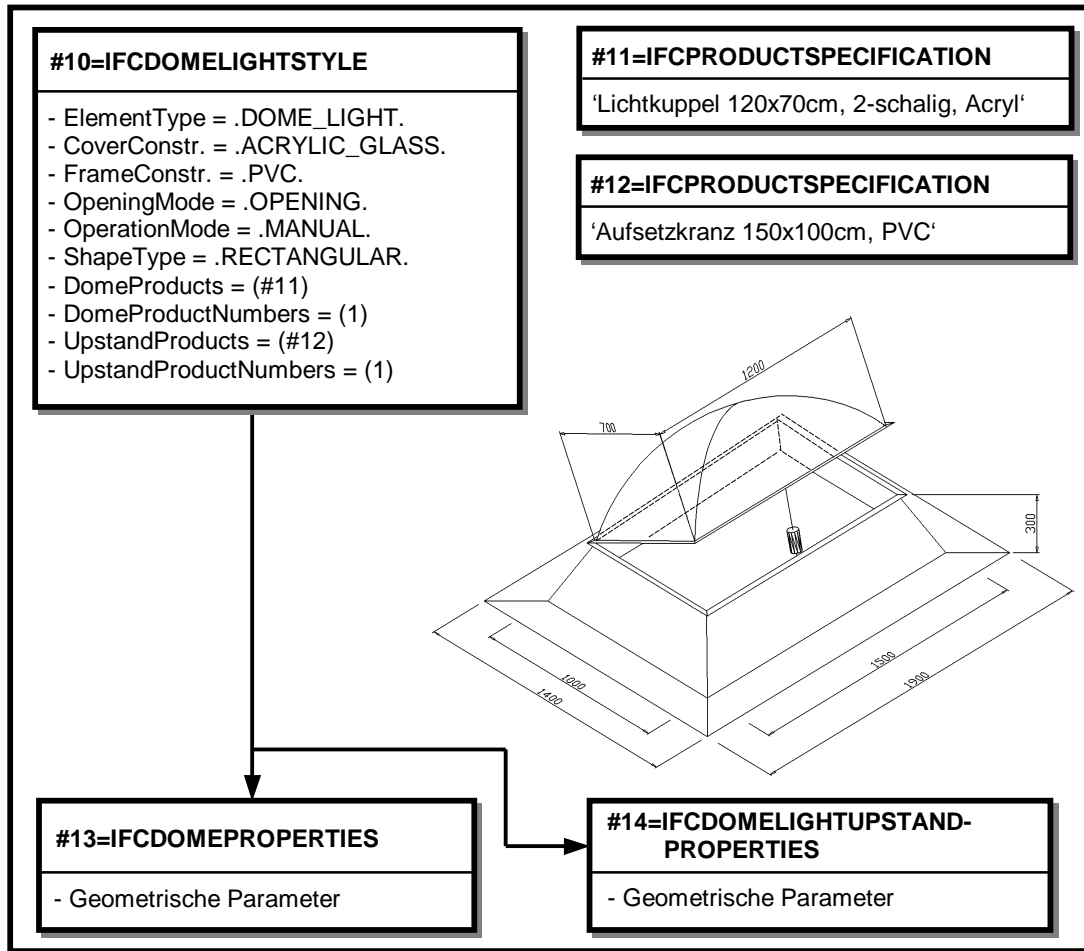


Abbildung 6.26: Typdefinition eines Lichtkuppel-elementes

6.7 Glasbau

6.7.1 Grundzüge der Darstellung

Von den in Kapitel 4 für den Glasbau geforderten zwei Detaillierungsgraden ist die Übersichtsplanung durch das gewerkeübergreifende Modell bereits abgedeckt. Ein wesentliches Merkmal der in diesem Abschnitt zu modellierenden Ausführungsplanung ist die Auflösung der globalen Bauelemente in Bauteile, ähnlich dem Gewerk Dach und Wand. Im Unterschied zu Dach und Wand existieren beim Glasbau jedoch keine Schichtungen. Das heißt, daß Doppelverglasungen oder Verbundgläser als ein einziges Bauprodukt und ein einziges Bauteil aufgefaßt werden. Der zweite wesentliche Unterschied ist die Bedeutung der Befestigungen im Glasbau. Die Art der Befestigung (punktförmige Halterung, ideale Punkthalterung, mechanische Linienlagerung und *Structural Sealant Glazing (SSG)*) bestimmt die gesamte übrige Konstruktion. Außerdem sind die topologischen Zusammenhänge zwischen Verglasung, Befestigung und Unterkonstruktion unter Umständen sehr komplex, und die Befestigungen selbst bestehen oft aus hochentwickelten Bauteilen. Aus diesen Gründen kann das Thema der Befestigung im Glasbau nicht so vereinfacht modelliert werden wie in der Domäne für Dach und Wand.

Als Vorüberlegung wird deshalb die Topologie von Glasfassaden genauer untersucht, besonders im Hinblick auf die Darstellung der Befestigungen. Die IFC2x ver-

fügen über die Klasse *IfcRelConnectsElements*, die genau zwei Bauteile auf unmittelbare Weise miteinander verbindet (1:1-Verbindung). Dies funktioniert immer dann eindeutig, wenn alle beteiligten Bauteile explizit dargestellt werden, also auch die Befestigungsbauteile selbst. Probleme treten dagegen dann auf, wenn ein Befestigungsbauteil mehr als zwei Teile verbindet, es selbst aber vereinfachend nicht dargestellt werden soll. Die Modellierung der Befestigungen ist also eng mit der Frage verbunden, welche Bestandteile explizit dargestellt werden sollen und welche nicht. Da diesbezüglich grundsätzliche Unterschiede bestehen, wird im folgenden zwischen linien- und punktgelagerten Glaskonstruktionen unterschieden. Liniengelagert sind die Bauarten „mechanische Linienlagerung“ und „SSG“, punktgelagert die Bauarten „punktförmige Halterung“ und „ideale Punkthalterung“.

Abbildung 6.27 zeigt diese Unterschiede anhand jeweils einer Konstruktion mit Verglasungsprofil und einem Mehrfachpunkthalter. Für die Befestigungen mit Verglasungsprofil (Abbildung 6.27a) kann mit der 1:1-Verbindung eine brauchbare Darstellung erzielt werden, da in dem hier betrachteten Detaillierungsgrad die Profile in der Regel explizit dargestellt werden. Dagegen ist der Aufwand für die vollständige Darstellung des Mehrfachpunkthalters sehr groß, zumal das in Abbildung 6.27b) gezeigte Beispiel noch nicht einmal der komplexeste Fall des Vierfachhalters ist. Hinzu kommt, daß die Punkthalter in der Regel bereits in der Stahlbaufertigung an die Fassadenträger geschweißt werden, so daß sie auch modelltechnisch eher dem Gewerk des Stahltragwerkes zuzuordnen sind.

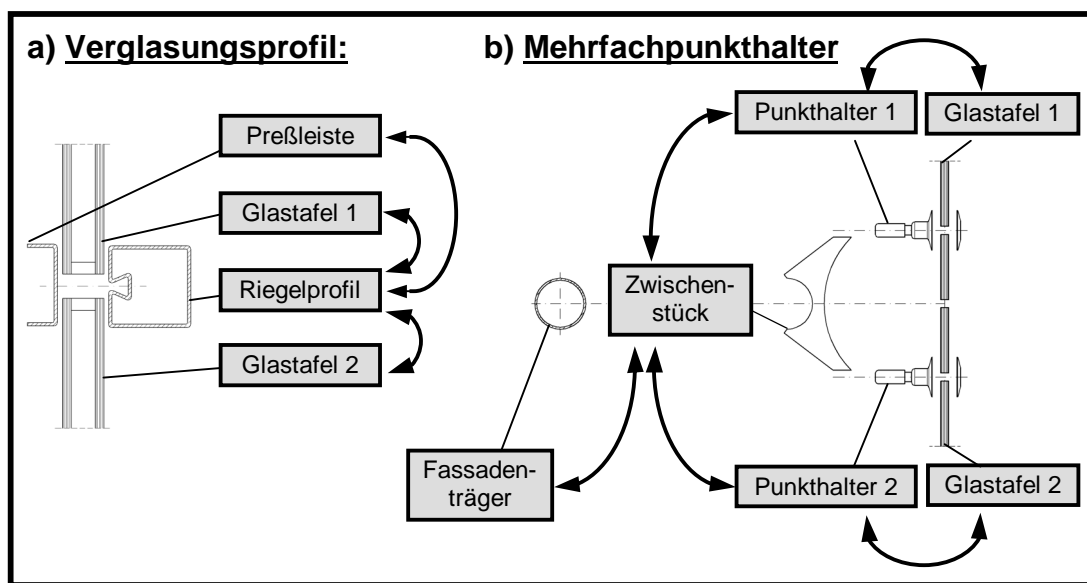


Abbildung 6.27: Topologie von Glasbefestigungen mit a) Verglasungsprofil und b) Mehrfachpunkthalter

Die Modellierung der Dekompositionshierarchie und der Befestigungen soll deshalb gemäß der folgenden, pragmatischen Grundsätze erfolgen.

a) Liniengelagerte Konstruktionen:

Die Hauptkomponenten bei der Auflösung von Bauelementen des Glasbaus mit Linienlagerung sind die eigentlichen Verglasungsbauteile und die Linienlager, also in der Regel die Verglasungsprofile für Pfosten und Riegel. Sie sollen jeweils explizit mit Blockmechanismus und vollständiger Geometrie dargestellt werden. Dies geschieht mit den Klassen *IfcGlazingMember* und *IfcGlazingMemberStyle* für Vergla-

sungsbauteile sowie *IfcGlazingSupportMember* und *IfcGlazingSupportMemberStyleLinear* für die Linienlager. Zur Darstellung aller weiteren Zubehörteile stehen die Klassen *IfcConstructionalElement* oder *IfcProxy* zur Verfügung. Nicht alle Zubehörteile müssen explizit dargestellt werden, sondern können vereinfacht auch lediglich in der Materialliste des übergeordneten Bauelementes *IfcGlassFaceIsDecomposed* aufgeführt werden. Die Verbindungen zwischen jeweils zwei Bauteilen werden mit der Klasse *IfcRelConnectsElements* dargestellt. Dieses Dekompositionsschema ist in Abbildung 6.28 skizziert.

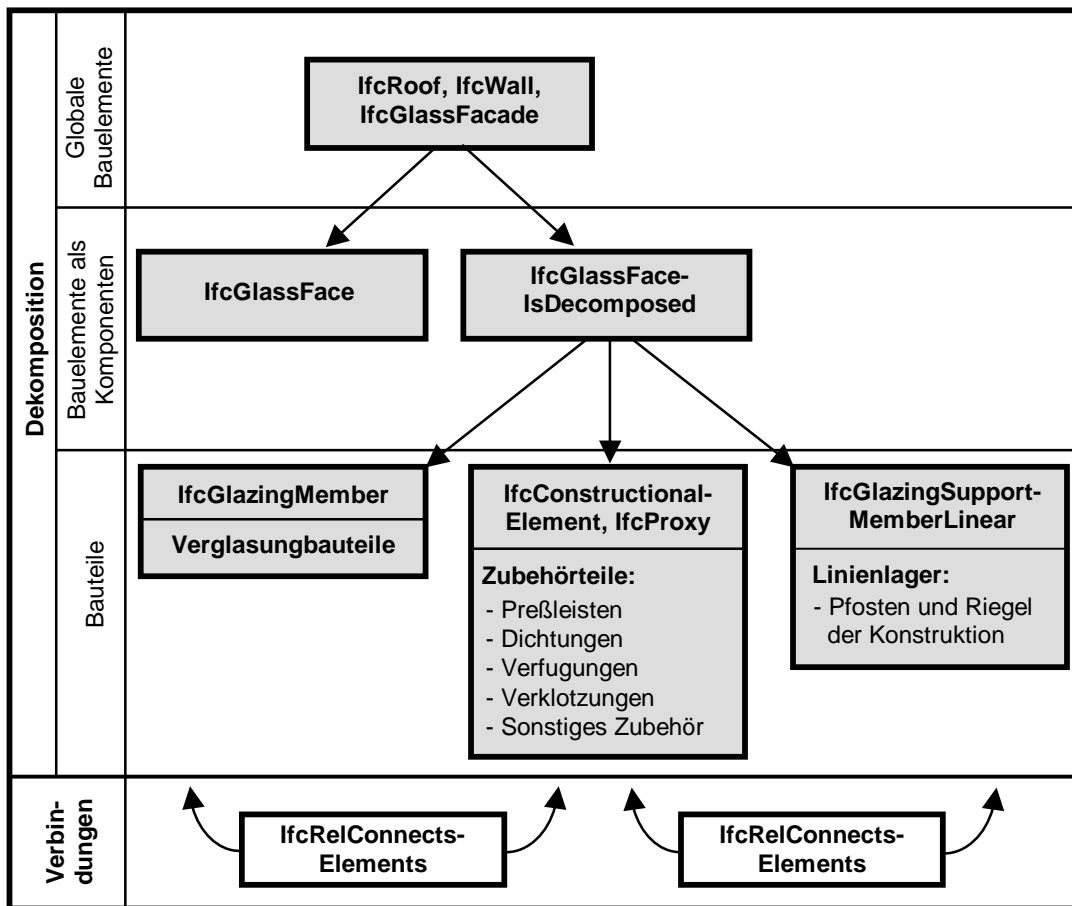


Abbildung 6.28: Dekompositionsschema liniengelagerter Glaskonstruktionen

b) Punktgelagerte Konstruktionen:

Bei der Auflösung von Bauelementen mit punktgelagerter Verglasung werden lediglich Verglasungs- und Zubehörteile explizit dargestellt, was analog zu Punkt a) geschieht. Die Punkthalter sollen dagegen als Teile der Unterkonstruktion dargestellt werden, zum Beispiel als Anbauteil an einen Fassadenträger durch *IfcProxy*. Sie gehören modelltechnisch deshalb nicht zur Glasbaudomäne. Abbildung 6.29 zeigt das Dekompositionsschema für punktgelagerte Glaskonstruktionen und die Verbindung zur Unterkonstruktion, die gestrichelt gezeichnet ist.

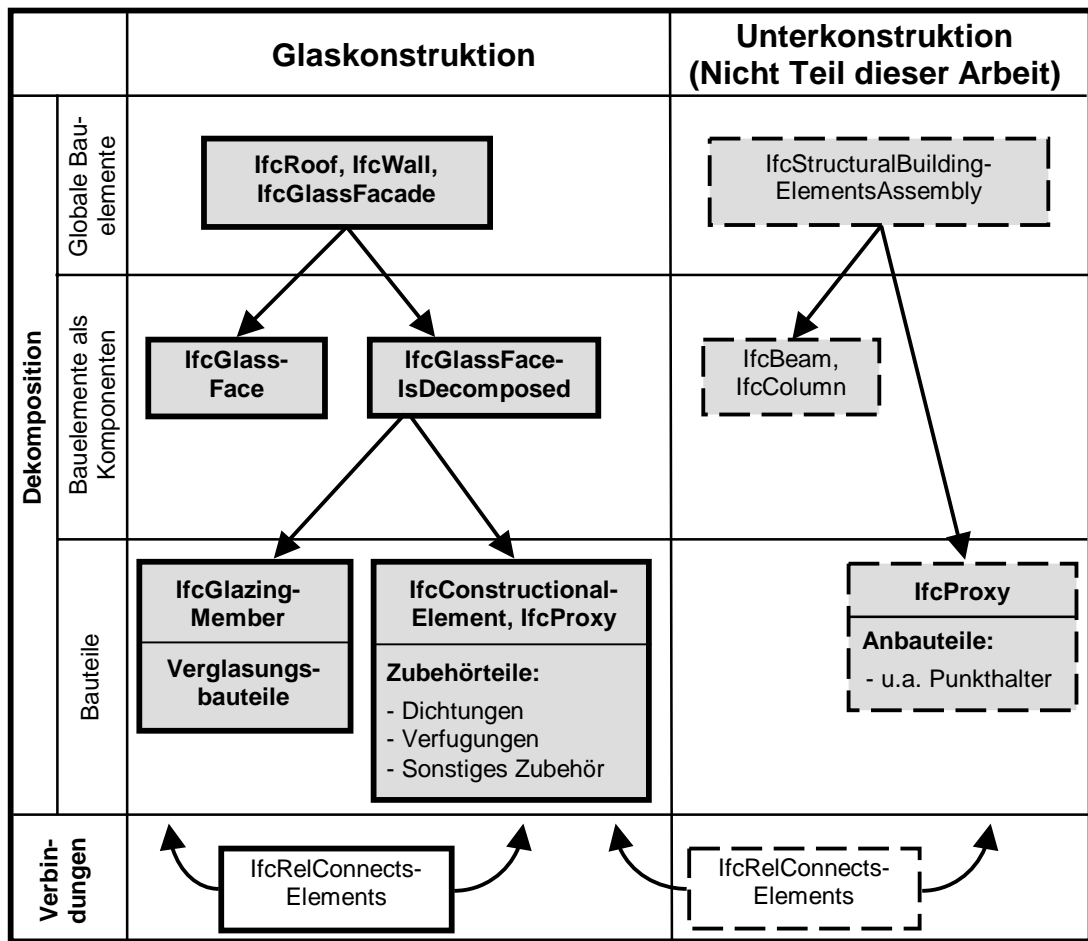


Abbildung 6.29: Dekompositionsschema punktgelagerter Glaskonstruktionen

Abbildung 6.30 zeigt eine Übersicht über das Schema *IFCGLAZINGCONSTRUCTIONDOMAIN*, in dem die neuen Klassen definiert sind. Es sind wiederum die Achsen „Detaillierungsgrad“ und „Auflösungsgrad“ eingetragen.

6.7.2 Auflösung der Bauelemente in Bauteile: *IfcGlassFaceIsDecomposed*

Die Auflösung der Bauelemente in Bauteile erfolgt nach dem in Abschnitt 6.3.5 erläuterten Prinzip, wobei ausschließlich Bauelemente der eigens dazu definierten Klasse *IfcGlassFaceIsDecomposed* aufgelöst werden. Das bedeutet, daß Dächer, Wände und Glasfassaden (*IfcRoof*, *IfcWallIsDecomposed* und *IfcGlassFacade*) ihrerseits zuvor in Bauelemente der Klasse *IfcGlassFaceIsDecomposed* aufgelöst werden müssen. Für diese Klasse sind die folgenden Eigenschaften definiert.

- 1) Einer Instanz von *IfcGlassFaceIsDecomposed* werden Bauprodukte als Materialliste (*IfcMaterialList*) zugeordnet. Dies bedeutet, daß diese Produkte in einer nicht weiter definierten Art und Weise in der Glaskonstruktion verwendet werden. Es wird jedoch keine Vorschrift darüber gemacht, bis zu welchem Detaillierungsgrad diese Liste vollständig sein muß. Wenn keine Materialliste referenziert wird, gilt diejenige des übergeordneten Bauelementes *IfcGlassFacade*.
- 2) Eine Positionsnummer kann im Attribut *Tag* der Oberklasse *IfcElement* angegeben werden.
- 3) *IfcGlassFaceIsDecomposed* verfügt über ein Einfügekoordinatensystem.

- 4) *IfcGlassFaceIsDecomposed* verfügt über keine geometrische Darstellung. Statt dessen wird sie gemäß der Abbildungen 6.28 oder 6.29 in Verlege-, Befestigungs- oder Zubehörbauteile aufgelöst.

Es ist die Aufgabe des Konstrukteurs, die Art und Anordnung der Bauteile mit Unterstützung seiner Software in Einklang mit gegebenenfalls im Stadium der Übersichtsplanung in Form von *Pset_GlassFaceRequirements* an das übergeordnete Bauelement gestellten Anforderungen zu bringen.

6.7.3 Verglasungsbauteile: *IfcGlazingMember*

Verglasungsbauteile werden unter Anwendung des Typmechanismus dargestellt. Es gilt wieder der Grundsatz, daß für identische Bauteile dieselbe Instanz zur Typdefinition verwendet werden muß.

Die Klasse *IfcGlazingMember* fügt dazu Instanzen der Klasse *IfcGlazingMemberStyle* im Koordinatensystem der übergeordneten Instanz *IfcGlassFaceIsDecomposed* ein. *IfcGlazingMember* verfügt über ein relatives Koordinatensystem, jedoch über keine eigene geometrische Darstellung. Die komplizierte Beziehung zwischen einfügender Instanz und Blockdefinition wird der Übersichtlichkeit halber wiederum vereinfacht als „Blockmechanismus“ bezeichnet.

6.7.4 Typdefinition für Verglasungsbauteile: *IfcGlazingMemberStyle*

Die Klasse *IfcGlazingMemberStyle* dient zur Typdefinition von Verglasungsbauteilen, wozu sie über die folgenden, von ihrer Oberklasse geerbten Eigenschaften verfügt:

- 1) Über *IfcRelAssociatesMaterial* wird ein Bauprodukt (*IfcProductSpecification*) zugeordnet. Die dazu referenzierte Instanz zur Produktspezifikation soll auch in der Materialliste des übergeordneten Bauelementes geführt werden.
- 2) *IfcGlazingMemberStyle* verfügt immer über eine vollständige geometrische Darstellung, die als Blockdefinition „gekapselt“ ist. Die Standardgeometrie für un bearbeitete Bauteile wird als *IfcSweptAreaSolid* unter Verwendung rechteckiger Profile dargestellt. Darauf basierend können CSG-Geometrien (*IfcCsgSolid*) für Bauteile erzeugt werden, die nicht rechteckig sind, oder die zum Beispiel Löcher für Punkthalter aufweisen.
- 3) Eine Positionsnummer wird im Attribut *Tag* der Oberklasse *IfcTypeProduct* angegeben.

Außerdem verfügt sie über die Attribute gemäß Tabelle 6.11. Für die entsprechenden Typisierungen siehe **Anhang A**.

<i>IfcGlazingSupportMember</i>	
SupportType:	Lagerungsart (Aufzählungstyp)
GlazingType:	Glasart (Aufzählungstyp)

Tabelle 6.11: *IfcGlazingMemberStyle*

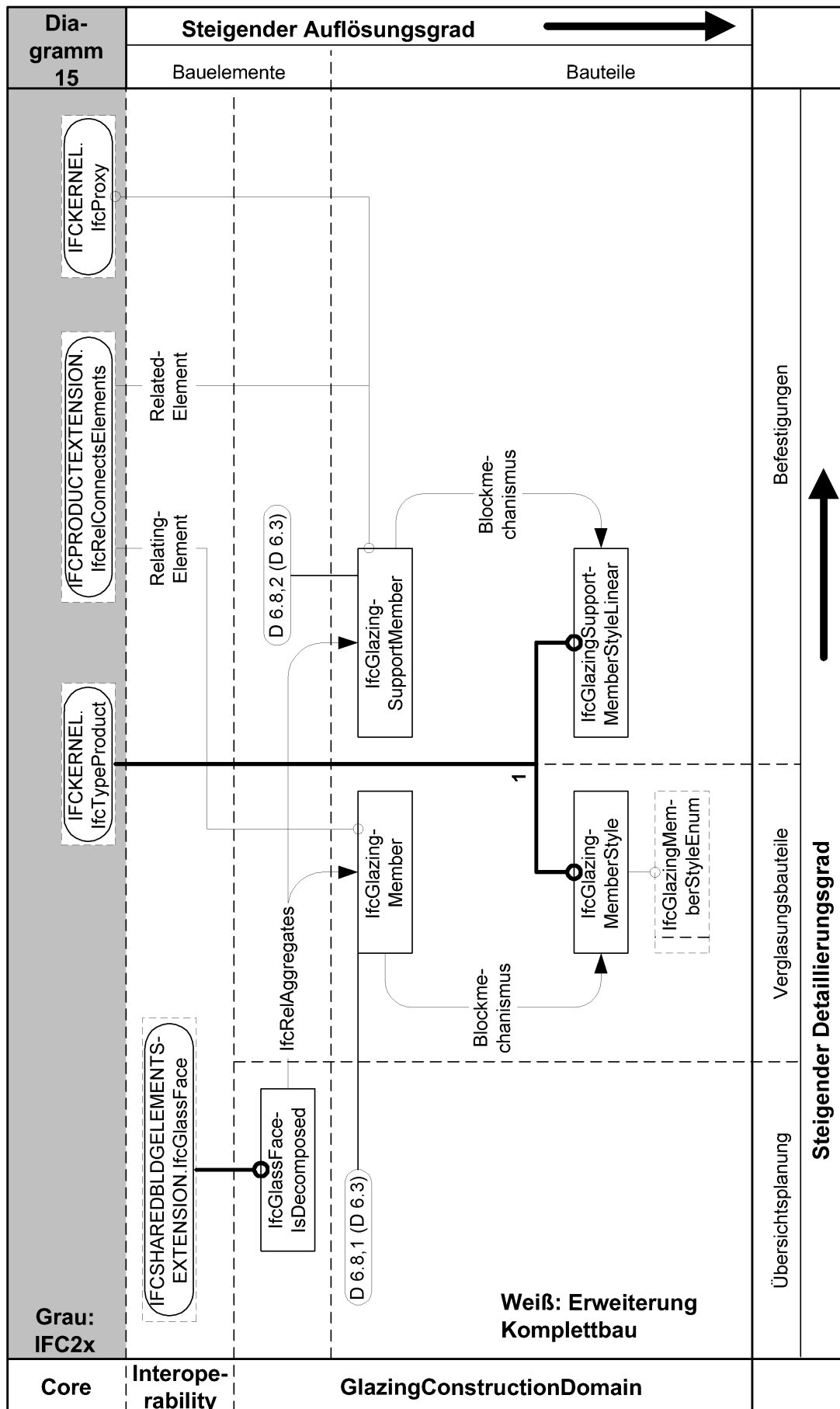


Abbildung 6.30: IFCGLAZINGCONSTRUCTIONDOMAIN (EXPRESS-G)

6.7.5 Befestigungsbauteile: *IfcGlazingSupportMember*

Die Klasse *IfcGlazingSupportMember* fügt Instanzen der Klasse *IfcGlazingSupportMemberStyleLinear* im Koordinatensystem der übergeordneten Instanz *IfcGlassFaceIsDecomposed* ein. Sie verfügt dazu über ein relatives Koordinatensystem, jedoch über keine eigene geometrische Darstellung.

6.7.6 Typdefinition für Befestigungsbauteile: *IfcGlazingSupportMemberStyleLinear*

Die Klasse *IfcGlazingSupportMemberStyleLinear* dient zur Typdefinition von linienförmigen Befestigungsbauteilen, wozu sie über folgende Eigenschaften verfügt:

- 1) Über *IfcRelAssociatesMaterial* wird ein Bauprodukt (*IfcProductSpecification*) zugeordnet. Die dazu referenzierte Instanz zur Produktspezifikation soll auch in der Materialliste des übergeordneten Bauelementes geführt werden. Wenn das Bauteil aus mehreren Produkten besteht, beispielsweise bei einem Verglasungsprofil samt Dichtungen, wird ihm eine Materialliste (*IfcMaterialList*) zugeordnet.
- 2) *IfcGlazingSupportMemberStyleLinear* verfügt immer über eine vollständige geometrische Darstellung, die als Blockdefinition „gekapselt“ ist. Die Standardgeometrie für unbearbeitete Bauteile wird als *IfcSweptAreaSolid* unter Verwendung beliebiger Profilformen dargestellt. Darauf basierend können CSG-Geometrien (*IfcCsgSolid*) erzeugt werden, zum Beispiel für Endbearbeitungen der Verglasungsprofile.
- 3) Eine Positionsnummer wird im Attribut *Tag* der Oberklasse *IfcTypeProduct* angegeben.

Es gilt auch hier, daß identische Bauteile mit derselben Instanz *IfcGlazingSupportMemberStyleLinear* dargestellt werden müssen.

6.7.7 Beispiel 1: Pfosten-Riegel-Verglasung

In Abbildung 6.31 ist ein Ausschnitt aus der Verglasungsfläche gezeigt, die bereits in Abschnitt 6.4.3.4 im Detaillierungsgrad der Übersichtsplanung dargestellt wurde. Gegenüber dem ursprünglichen Beispiel wird die Verglasung hier gemäß der Vorgaben aus Instanz #10 (*Pset_GlassFaceRequirements*) bis auf die Befestigungsmittel der Preßleisten durchkonstruiert.

Die ursprüngliche Instanz des Bauelementes *IfcGlassFace* wurde hierzu vom CAD-System des Glaskonstruktors, das die Austauschdatei erzeugt, durch *IfcGlassFaceIsDecomposed* ersetzt (#5). Sie besitzt eine ausführliche Materialliste (#12) mit Bauprodukten für Glastafeln, Verglasungsprofile, Preßleisten und Dichtungen. Anstelle der ursprünglichen geometrischen Darstellung wurde das Bauelement in Bauteile aufgelöst, von denen aus Platzgründen lediglich die Instanzen #21 bis #25 aufgeführt sind. Alle Arten von Bauteilen, also Verglasungs-, Befestigungs- und Zubehörbauteile, werden gleichwertig behandelt und im Koordinatensystem des übergeordneten Bauelementes angeordnet. Die aufgeführten Instanzen der Bauteile fügen Typdefinitionen ein, von denen wiederum auf die Instanzen der übergeordneten Materialliste verwiesen wird. Vereinfachend sind in den Typdefinitionen für Pfosten, Riegel und Preßleisten jeweils die Geometrie der Verglasungsprofile und der Dichtungen zusammengefaßt. Aus diesem Grunde werden in den entsprechenden Instanzen (#21, #22, #24 und #25) jeweils zwei Bauproduktspezifikationen referenziert.

Die Instanzen #26 bis #29 der Klasse *IfcRelConnectsElements* stellen die Verbindung zwischen jeweils zwei Bauteilen her.

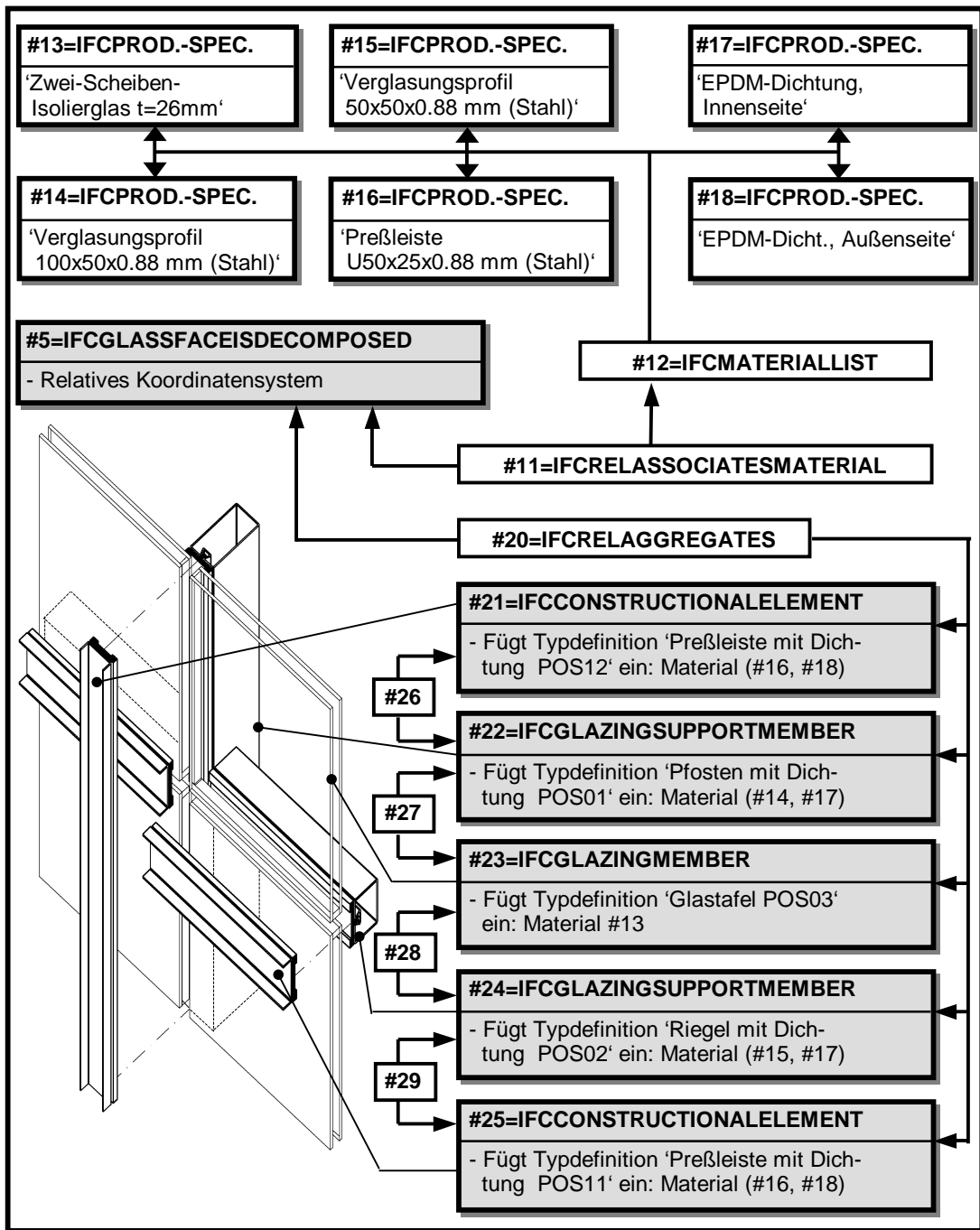


Abbildung 6.31: Beispiel für eine Pfosten-Riegel-Verglasung (Ausschnitt)

6.7.8 Beispiel 2: Punktgestützte Verglasung

Das in Abbildung 6.32 gezeigte Beispiel verdeutlicht die Darstellung von punktgestützten Verglasungen. Aufgrund der Regel, daß die Befestigungsbauteile modelltechnisch zur Unterkonstruktion gehören, ist in der Abbildung außer der eigentlichen Fassade ein Horizontalträger aus Stahl eingezeichnet. Die zu dessen Darstellung verwendeten Instanzen befinden sich oberhalb der gestrichelten Linie: Die Baugruppe *IfcStructuralElementsAssembly* (#1) wird in die verschiedenen Stahlbauteile aufgelöst. Von diesen Komponenten ist aus Platzgründen nur ein einzelner Punkthalter aufgeführt, der als *IfcProxy* (#3) ohne Geometrie dargestellt ist.

Die Instanzen der eigentlichen Glasfassade sind in der Abbildung unterhalb der gestrichelten Linie aufgeführt. Wie bei der liniengelagerten Bauart wird einer Instanz *IfcGlassFaceIsDecomposed* (#4) eine Materialliste (#6) zugeordnet, in der die verwendeten Bauprodukte geführt werden. Neben der Verglasung ist dies im Beispiel nur das Abdichtungsband für die Fugen zwischen den Glastafeln. Die Instanz #4 verfügt über keine eigene geometrische Darstellung, sondern wird in Bauteile aufgelöst. Im Beispiel ist aus Platzgründen lediglich ein Verglasungsbauteil dargestellt (#10). Die zur Typdefinition dienende Instanz #11 des Verglasungsbauteils verfügt über eine Bauproduktdefinition (#7), die auch in der globalen Materialliste geführt wird, und über eine CSG-Geometrie mit Löchern für die Punkthalter. Die Zuordnung dieser Typdefinition zur einfügenden Instanz ist stark vereinfacht durch den gestrichelten Pfeil skizziert. Über die Instanz #12 wird schließlich die Befestigung der Glastafel am Punkthalter des Fassadenträgers dargestellt.

Das Beispiel zeigt außerdem sehr gut die Funktionsweise des gewerkeübergreifenden Modells im Komplettbau: Der zum Gewerk des Stahltragwerkes gehörende Punkthalter wird durch die allgemeinverständliche Klasse *IfcProxy* dargestellt, so daß er auch von den Bauteilen der gewerkespezifischen Modelle aus referenziert werden kann.

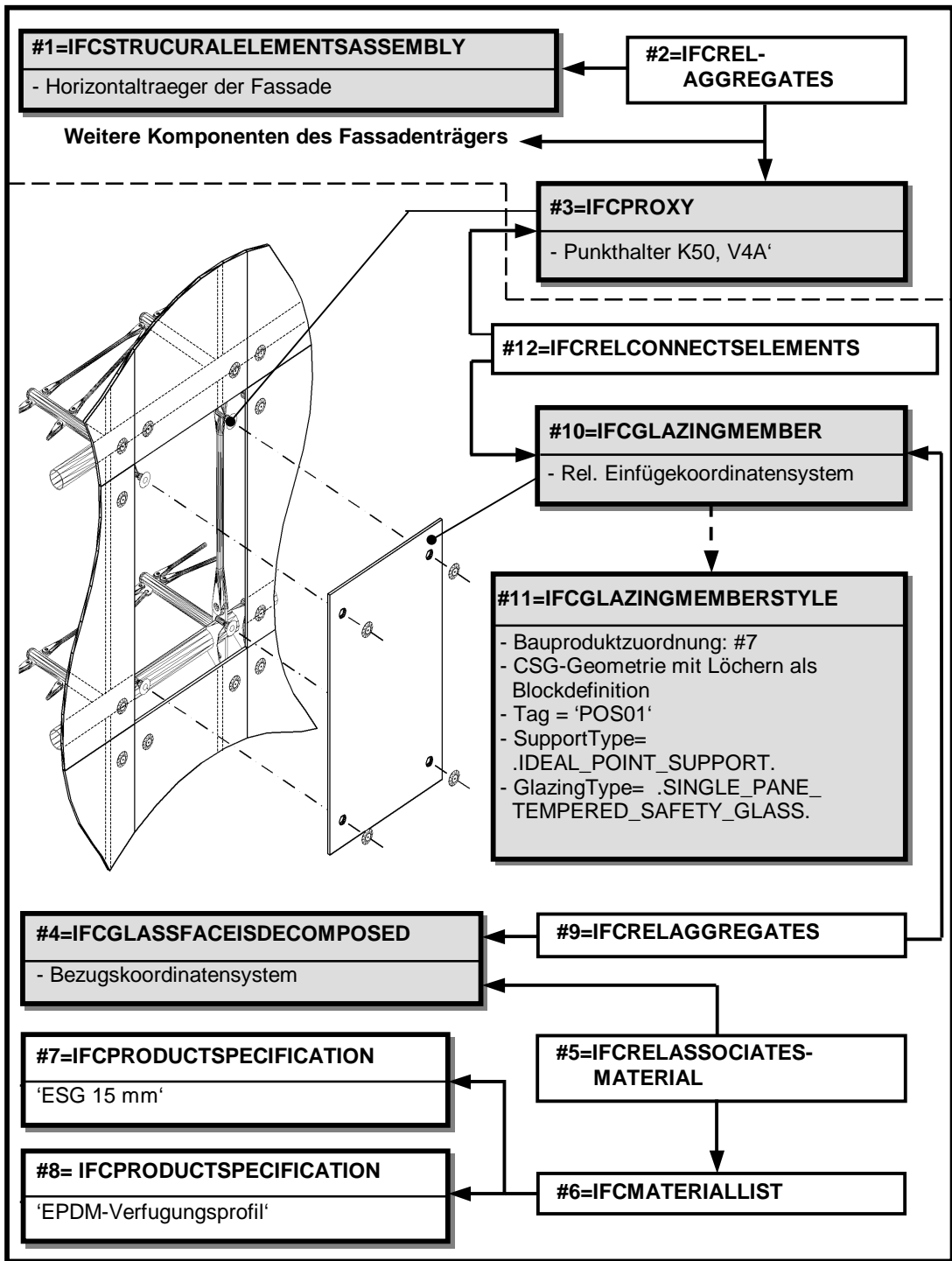


Abbildung 6.32: Punktgestützte Verglasung mit Unterkonstruktion

7 Sichten

7.1 Allgemeines

Das Datenmodell stellt die für den Produktdatenaustausch benötigten Sprachmittel zur Verfügung. Da jedoch keine Anwendungs-Software alle Sprachmittel interpretieren kann, müssen Untermengen gebildet werden, die sogenannten *Sichten*. Ein Software-Hersteller kann wählen, welche Sicht sein Programm unterstützen soll, und dadurch dessen Fähigkeiten zum Produktdatenaustausch bestimmen. Die für jede Sicht definierten *Konformitätsanforderungen* dienen dem Schnittstellenprogrammierer als Anleitung dafür, welche Klassen mit welchen Funktionen er zu implementieren hat.

Natürlich muß auch der Software-Anwender wissen, welche Sichten sein Programm unterstützt, um daraus die zur Verfügung stehenden Möglichkeiten zum Produktdatenaustausch mit anderen Programmen zu erkennen. Dabei können nicht nur Programme gekoppelt werden, die dieselbe Sicht des Datenmodells unterstützen, sondern auch solche mit unterschiedlichen Sichten. Die austauschbaren Dateninhalte sind in diesem Fall die Schnittmenge der verschiedenen Sichten. Abbildung 7.1 veranschaulicht dieses Prinzip, wenn man sich eine Sicht als Lampe vorstellt. Der Lichtkegel der Lampe erhellt einen Bereich auf einer Fläche, die dem Datenmodell entspricht. Die zwischen unterschiedlichen Sichten austauschbaren Daten befinden sich innerhalb der Verschneidung der Lichtkegel.

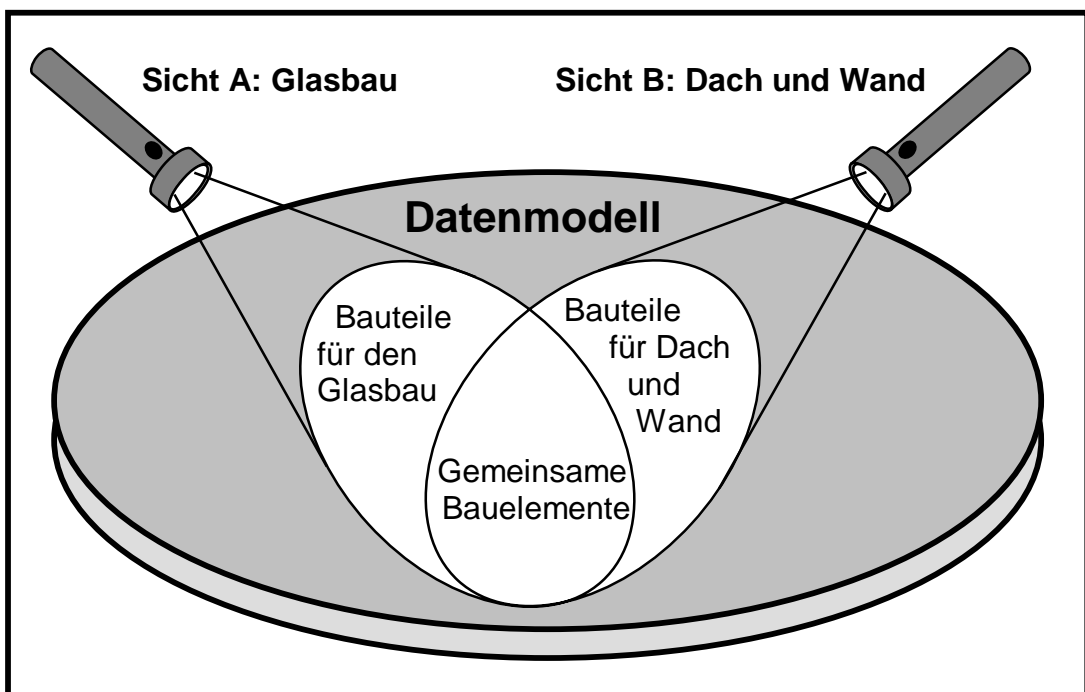


Abbildung 7.1: Veranschaulichung des Sichtenprinzips

Die Frage, welche Sichten für das Kompletbaumodell definiert werden sollen, kann derzeit nicht endgültig beantwortet werden, da für das übergeordnete Datenmodell der IFC2x noch keine offiziellen Sichten existieren. Die im folgenden vorgestellten Sichten müssen deshalb gegebenenfalls mit zukünftigen, offiziellen Definitionen

harmonisiert werden. Dies ist allerdings kein Problem, da durch die Domänenstruktur des Komplettbaumodells zwar eine gewisse Einteilung bereits nahegelegt wird, die Sichtdefinitionen jedoch schemaübergreifend sein können und somit unabhängig von der Modellarchitektur sind.

Aufgrund ihres vorläufigen Charakters werden die Sichten in diesem Abschnitt nicht bis ins Detail formuliert. Alle vorgeschlagenen Sichten gelten für den Produktdatenaustausch mit Dateien.

Erfahrungen mit der Implementierung und Anwendung der Produktschnittstelle Stahlbau haben gezeigt, daß für die vertikale Kopplung der Planungsprozesse eine höhere Akzeptanz seitens der Software-Hersteller zu erwarten ist als für die horizontale. Die Gründe hierfür sind, daß die vertikale Integration vom Standpunkt der Software-Hersteller den größeren Nutzen durch Schaffung von Prozeßketten hat, daß ihre Implementierung weniger aufwendig ist, und daß die horizontale Kopplung eine stärkere Öffnung der Systeme zu unmittelbar konkurrierenden Software-Produkten bedeutet. Letzteres wird nach wie vor skeptisch beurteilt. Diese Argumente gelten zwar nicht für den Standpunkt des Anwenders, aus pragmatischen Gründen ist die vertikale Kopplung der Planungsschritte dennoch das vorrangige Ziel dieses Kapitels.

7.2 Vertikale Systemkopplung

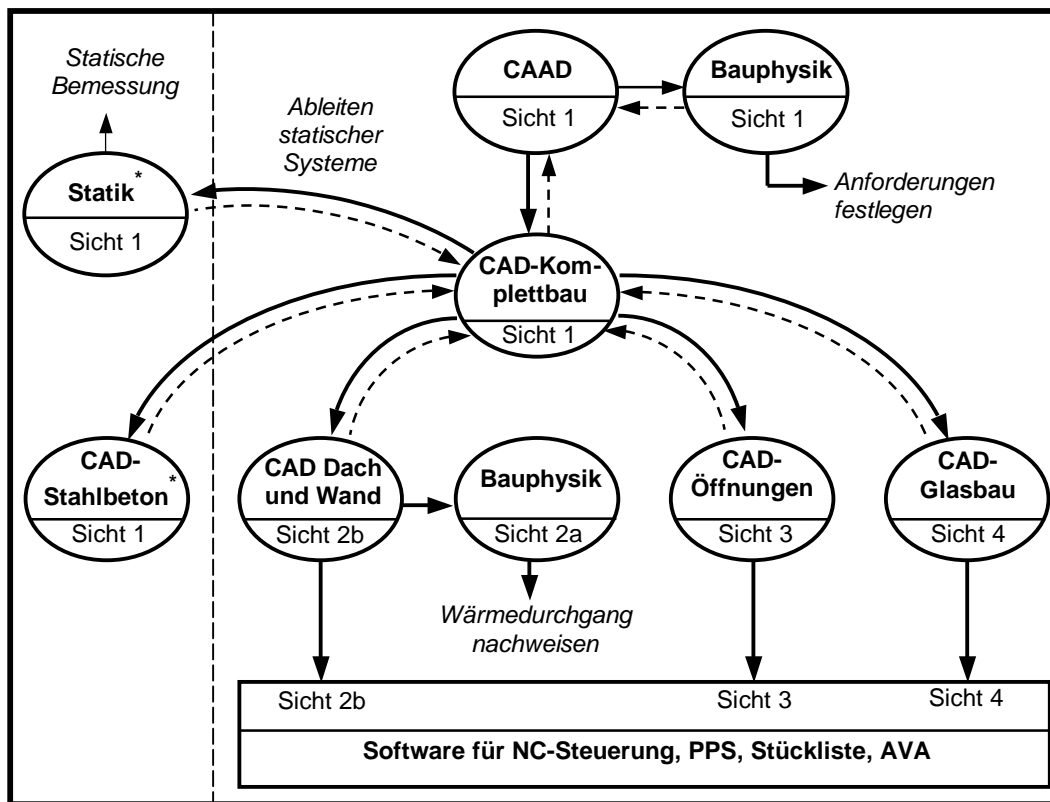


Abbildung 7.2: Anwendungsszenario mit notwendigen Sichten

In Abbildung 7.2 wird das gewerkeübergreifende Anwendungsszenario aus Abschnitt 4.2.2 aufgegriffen, bei dem die Koordinierung der Fachplanungen im Objektmodell eines CAD-Systems für den Stahl- oder Komplettbau geschieht. Die vertikalen Austauschwege sind, wie gehabt, mit durchgezogenen Pfeilen dargestellt. Die Akteure des ursprünglichen Szenarios wurden in der Abbildung dagegen durch

die Anwendungsprogramme mitsamt der von ihnen zum Datenaustausch unterstützten Sichten ersetzt.

Die Anwendungsbereiche der beiden links von der gestrichelten Linie dargestellten, mit einem Stern gekennzeichneten Programme (Statik und Stahlbeton-CAD) sind nicht Bestandteil des Datenmodells. Trotzdem können sie in das Szenario eingebunden werden, indem das CAD-System die Bauelemente der Übersichtsplanung interpretiert, und das Statikprogramm aus den Bauelementen für das Tragwerk geeignete statische Systeme oder zumindest Schwerelinienmodelle ableitet. Eine weitergehende Integration ist für diese Programme jedoch nicht möglich.

Tabelle 7.1 zeigt, welche Dateninhalte die verschiedenen Sichten zur Realisierung dieses Szenarios umfassen müssen.

Nr.	Name	Software	Dateninhalte
Sicht 1	<i>Komplettbau – Übersichtsplanung</i>	CAAD, CAD für Komplett- und Stahlbetonbau*, Bauphysik, Statik*	- Projekt- und Bauwerkstruktur - alle Bauelemente des gewerkeübergreifenden Modells - Anforderungen an Bauelemente
Sicht 2a	<i>Dach und Wand – Detailplanung</i>	CAD für Dach und Wand, Bauphysik	- Projekt- und Bauwerkstruktur - einfache Bauelemente: Tragwerk, Türen, Fenster, Tore, Dachbelichtung, RWA - Bauelemente mit Schichtaufbau und Bauprodukten: Dächer und Wände
Sicht 2b	<i>Dach und Wand – Verlegeplanung</i>	CAD für Verlegeplanung, PPS, Stückliste, AVA	- Umfang von Sicht 2a - Verlegebauteile, Zubehörteile, Gruppierungen zu Schichten, Befestigungen, Palettierung
Sicht 3	<i>Öffnungen – Ausführungsplanung</i>	CAD, NC-Steuerung, PPS, Stückliste, AVA	- Projekt- und Bauwerkstruktur - einfache Bauelemente: Dächer, Wände - Bauelemente mit Bauprodukten: Türen, Fenster, Tore, Dachbelichtung, RWA
Sicht 4	<i>Glasbau – Ausführungsplanung</i>	CAD für Glasbau, NC-Steuerung, PPS, Stückliste, AVA	- Projekt- und Bauwerkstruktur - einfache Bauelemente: Dächer, Wände, Tragwerk, Türen, Fenster - Bauelemente mit Bauprodukten: Glasfassaden, Glasflächen - Verglasungsbauteile, Befestigungsbauteile, Zubehörteile für den Glasbau

Tabelle 7.1: Zur vertikalen Kopplung notwendige Sichten

Für alle fünf Sichten gelten außerdem die folgenden, allgemeinen Konformitätsanforderungen.

1. Beim Datenexport:

- a) Die Daten müssen in Austauschdateien gemäß ISO 10303, Teil 21, und gemäß der Klassendefinitionen des gewerkeübergreifenden Komplettbaumodells exportiert werden.
- b) Alle Objekte müssen über eine einzigartige Objektidentität verfügen.
- c) Alle Bauelemente müssen auf Instanzen der Klasse *IfcOwnerHistory* verweisen.

2. Beim Datenimport:

- a) Das Programm muß Austauschdateien gemäß ISO 10303, Teil 21, einlesen können. Die Dateien können Instanzen aller Klassen der IFC2x und des Komplettbaumodells enthalten.
- b) Das Programm darf diesen Datenbestand jedoch selektieren. Klassen, die in der vom Programm unterstützten Sicht nicht gefordert werden, dürfen ignoriert werden. Außerdem dürfen Bauelemente ignoriert werden, die außerhalb des Funktionsumfangs des Programms liegen, auch wenn sie in der Sicht gefordert sind.
- c) Alle übrigen Instanzen müssen vom Programm interpretiert werden, so daß sie dem Benutzer zur weiteren Bearbeitung zur Verfügung stehen.

3. Zusätzliche Anforderungen beim Export zuvor importierter Daten:

- a) Die zuvor importierten Daten der Projekt- und Bauwerkstruktur müssen erhalten bleiben. Sie müssen an eventuelle Änderungen des übrigen Datenbestandes angepaßt werden.
- b) Die Objektidentitäten aller zuvor interpretierten Objekte müssen unverändert erhalten bleiben.
- c) Änderungen an allen zuvor importierten Bauelementen müssen in den entsprechenden Instanzen von *IfcOwnerHistory* protokolliert werden.

Mit diesen Sichten lassen sich alle in Abschnitt 4.2 dargestellten, vertikalen Kopplungen realisieren. Dies schließt sowohl die Variante des gerade gezeigten Szenarios ein, bei der das CAAD-System des Architekten die Fachplanungen koordiniert, als auch die gewerkespezifischen Prozesse für Dach und Wand, Öffnungen und den Glasbau. Ein vereinfachtes Szenario, das ausschließlich den Datenaustausch bis zu den Fachplanern ermöglicht, läßt sich erreichen, indem auch die Programme der Fachplaner lediglich die Sicht 1 implementieren.

7.3 Horizontale Kopplung von Teilprozessen

Eine vollständige, sowohl vertikale als auch horizontale Integration der Planungsschritte kann nur mit Datenbankunterstützung erfolgen. Über die oben definierten Sichten hinausgehend ist jedoch ein Minimum an horizontaler Systemkopplung auch ohne weitere Datenbank- oder Software-Unterstützung möglich und sinnvoll. Dies betrifft die Datenaustauschwege, die zur Kontrolle und Koordination der Fachplanungen im zentralen Objektmodell benötigt werden. Sie sind in Abbildung 7.2 mit gestrichelten Pfeilen dargestellt. Die dazu vorgeschlagene Lösung soll mit möglichst geringem zusätzlichem Aufwand implementiert und angewandt werden können. Es wird deshalb nicht verlangt, daß die Programme bei der Weitergabe von Daten den gesamten Datenbestand konsistent erhalten müssen, den sie zum großen Teil gar nicht interpretieren können.

Die für die beabsichtigten Austauschwege notwendigen Dateninhalte sind durch die Sichten aus Tabelle 7.1 bereits abgedeckt. In der Regel kann jedoch weder das koordinierende CAD-System die Detaillierungsgrade der Fachplanungen interpretieren,

noch können die Programme der Fachplanungen alle Bauelemente des zentralen Objektmodells verstehen. Deshalb wird vom koordinierenden CAD-System zusätzlich gefordert, daß es die Teildatenbestände der Fachplanungen in den Gesamtbestand integrieren kann. Diese Anforderungen werden in einer weiteren Sicht definiert:

Nr.	Name	Software	Dateninhalte (wie Sicht 1)
Sicht 5	<i>Komplettbau – Koordination</i>	CAAD, CAD für den Komplettbau	- Projekt- und Bauwerkstruktur - alle Bauelemente des gewerkeübergreifenden Modells - Anforderungen an Bauelemente

Tabelle 7.2: Sicht zur horizontalen Integration von Teilprozessen

Für diese Sicht gelten ebenfalls die Punkte 1 bis 3 der Konformitätsanforderungen aus Abschnitt 7.2, sowie der folgende, zusätzliche Punkt:

4. Fähigkeit zur Integration von verändertem Datenbestand:

- a) Das Programm muß in einer Austauschdatei, die ursprünglich von ihm selbst erzeugte, jedoch in anderen Programmen veränderte Daten enthält, das Projekt anhand der Objektidentität identifizieren können. Außerdem muß es den ursprünglichen Datenbestand parallel wieder zur Verfügung stellen können.
- b) Vom Programm ursprünglich selbst exportierte Bauelemente müssen anhand ihrer Objektidentität identifiziert werden können.
- c) Das Programm muß die veränderten Daten für diese Bauelemente in geeigneter Weise interpretieren können. Insbesondere muß es aus gewerkespezifischen Unterklassen der ursprünglichen Bauelemente die für sich selbst relevanten Informationen erkennen können. Bauelemente, die aus gewerkeübergreifender Sicht beispielsweise nicht in Komponenten aufgelöst werden, können in den gewerkespezifischen Sichten Unterklassen besitzen, die eine Auflösung in Bauteile vorschreiben. Das Programm muß dann in der Lage sein, eine Geometrie aus allen Bauteilen des aufgelösten Bauelementes abzuleiten.
- d) Das Programm darf die betroffenen Bauelemente selektiv interpretieren. Die Bauteile, aus denen Geometrien abgeleitet werden, brauchen selbst nicht eingelesen zu werden, da die Kopplung zu Fertigung, Stücklisten und AVA bei den Fachplanern verbleibt.
- e) Das Programm muß den veränderten Datenbestand der betrachteten Bauelemente in den ursprünglichen, globalen Datenbestand integrieren können. Dies beinhaltet, daß Verbindungen zu und Gruppierungen mit anderen Bauelementen erhalten bleiben müssen, aus denen die Struktur und die Topologie des Bauwerks besteht.
- f) Falls die Integration nicht oder nicht vollständig möglich ist, muß das Programm dem Anwender dies anzeigen und die Möglichkeit geben, darauf zu reagieren. Mögliche Reaktionen sind das Löschen bestehender Verknüpfungen zwischen Bauelementen oder im schlimmsten Fall der Abbruch der Funktion.

Die beteiligten Programme der Fachplaner erfüllen alle Voraussetzungen für diese horizontale Integration bereits durch die in den Sichten 2 bis 4 gestellten Konformitätsanforderungen, wonach sie die für sich relevanten Bauelemente, die Projekt- und die Bauwerkstruktur interpretieren und wieder exportieren müssen. Der nicht interpretierte Datenbestand muß dagegen nicht erhalten bleiben.

Beispiel:

Abbildung 7.3 zeigt den aus dem gesamten Szenario nach Abbildung 7.2 herausgelösten Austauschweg vom CAD-Komplettbau zur Fachplanung für Dach und Wand mit Rückgabe der veränderten Daten (gestrichelter Pfeil). Die dazu erzeugten Austauschdateien sind in den grau hinterlegten Kästchen mit der in Kapitel 6 bereits verwendeten Notation für die Daten dargestellt.

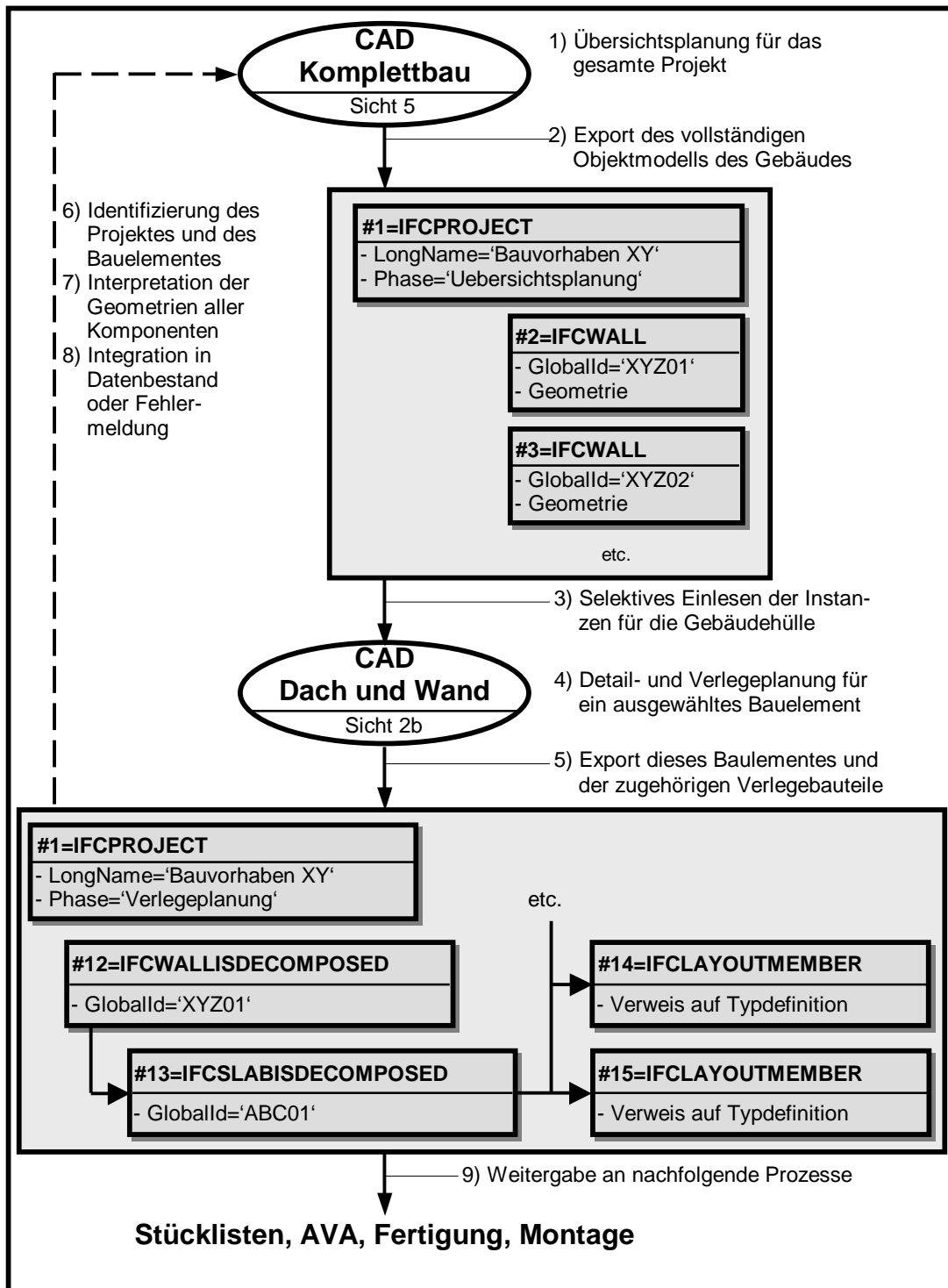


Abbildung 7.3: Ablauf des integrierten Teilprozesses

Im CAD-System für den Komplettbau wird die gesamte Übersichtsplanung durchgeführt (Schritt 1)). Das Programm übernimmt außerdem die Koordination der Fachplanungen und unterstützt dazu die Sicht 5 des Komplettbaumodells. Es exportiert eine Austauschdatei mit dem vollständigen Objektmodell des Gebäudes (Schritt 2)), das aus allgemeinen Daten zu Projekt- und Bauwerkstruktur sowie den Bauelementen besteht. Diese Datei wird an die Fachplaner weitergegeben. Im Beispiel wird sie in das CAD-System für Dach und Wand importiert, das dazu die Sicht 2b unterstützt. Es darf demnach die Bauelemente der Gebäudehülle selektiv einlesen, so daß die übrigen Bauelemente verloren gehen (Schritt 3)). Anschließend wird für eine einzelne Wand die Detail- und Verlegeplanung durchgeführt (Schritt 4)), deren Ergebnis in eine neue Austauschdatei exportiert wird (Schritt 5)). Es genügt dabei, nur das ausgewählte Bauelement zu schreiben. Allerdings muß dessen Objektidentität und die globale Projektstruktur erhalten bleiben. Das Bauelement wurde im Vergleich zur ursprünglichen Austauschdatei jedoch stark verändert: Die Instanz #12 ist vom Typ *IfcWallIsDecomposed*. Sie ist also eine Unterklasse des ursprünglichen Typs *IfcWall* und ist anstelle einer Geometrie in Komponenten aufgelöst (Instanz #13, die ihrerseits in die Verlegebauteile #14 und #15 aufgelöst ist). Die Instanznummern der ursprünglichen Datei werden nicht beibehalten.

In den Schritten 6) bis 8) wird die neue Datei wieder an das koordinierende CAD-System zurückgegeben, das die Daten importiert und in das ursprüngliche Projekt integriert. Dazu werden zuerst das Projekt und das betroffene Bauelement anhand ihrer Objektidentitäten identifiziert. Danach interpretiert das Programm die für die Detaillierung der Übersichtsplanung notwendigen Daten: Es muß die Geometrie für die Instanz der Klasse *IfcWallIsDecomposed* aus der Gesamtheit der Bauteile ableiten. Dies ist aufwendig, jedoch grundsätzlich möglich, da die Dekomposition mit allgemeingültigen Datenstrukturen der IFC dargestellt ist. Falls die Integration nicht gelingt, wird eine Fehlermeldung ausgegeben.

In Schritt 9) wird der in der Fachplanung erzeugte Datenbestand an die nachfolgenden Prozesse weitergegeben. Dies geschieht in Verantwortung des Fachplaners und unabhängig von der oben beschriebenen Datenrückgabe.

Die Grenzen dieses pragmatischen Verfahrens werden erreicht, sobald *Planungskonflikte* auftreten. Dies ist der Fall, wenn ein Fachplaner bei der Ausführungsplanung „fremde“ Bauelemente verändert, oder wenn der ursprüngliche Datenbestand in der Übersichtsplanung geändert wird, während die Fachplaner die Ausführungsplanungen dazu erstellen. Die Nutzung dieses Verfahrens bedarf also einer genauen Kenntnis der Möglichkeiten durch den Generalplaner und einer entsprechend genauen Aufgabenverteilung.

Weitere, horizontale Kopplungen wären unter Umständen unmittelbar zwischen den CAD-Programmen der Fachplaner sinnvoll. Die dazu notwendigen Dateninhalte sind durch die Forderung der Sichten 2 bis 4 nach Unterstützung gemeinsamer Bauelemente bereits abgedeckt. So soll ein CAD-System, mit dem Fenster konstruiert werden, nicht nur die unmittelbaren Bauelemente für Fenster interpretieren können, sondern auch betroffene Teile der Gebäudehülle. Zusätzlich dazu wären für diese Kopplung jedoch ähnliche Programmfähigkeiten erforderlich, wie sie in Sicht 5 definiert wurden. Es ist jedoch fraglich, ob und wie die auf Spezialgebiete fokussierten Programme der Fachplaner eine Integration von Teilbeständen der Projektdaten in den globalen Datenbestand überhaupt leisten können. Deshalb wird auf die Definition entsprechender Sichten an dieser Stelle verzichtet. Dies muß bei Bedarf unter Einbeziehung der betroffenen Software-Hersteller geschehen.

7.4 Bestandteile der Sichtdefinitionen

Die Definition einer Sicht soll die folgenden Bestandteile enthalten:

- 1) Eine Erläuterung der Funktion, des Anwendungsgebietes und der Dateninhalte der Sicht. Dies sollte auch für den Software-Anwender verständlich sein.
- 2) Allgemeine, für die Schnittstellenprogrammierung verbindliche Konformitätsanforderungen in Textform, wie es in den Punkten 1) bis 4) der Abschnitte 7.2 und 7.3 gezeigt wurde.
- 3) Für die Schnittstellenprogrammierung verbindliche Festlegung der Dateninhalte.

Die Dateninhalte wurden in den Tabellen 7.1 und 7.2 anschaulich in Form der zu unterstützenden Bauelementen und Bauteile festgelegt. Für Sichtdefinitionen, die tatsächlich implementiert werden sollen, muß dies jedoch sehr viel genauer anhand der Klassen des EXPRESS-Schemas geschehen. Es liegt nahe, dies in formaler Art und Weise zu tun, wozu EXPRESS selbst die sogenannten *Use-Interfaces* bietet. Sie ermöglichen es, in einem Schema einzelne Klassen aus anderen Schemata zu „benutzen“, und implizit dadurch auch deren Oberklassen sowie alle Datentypen und Klassen, zu denen Verweise bestehen. Somit kann eine Sicht in einem eigenen EXPRESS-Schema definiert werden, von dem aus die benötigten Klassen benannt werden.

Abbildung 7.4 zeigt dies beispielhaft für die Sicht 2b (*Komplettbau – Verlegeplanung*). Im oberen Teil ist ein Fragment des Schemas *IfcOuterShellDomain* des Datenmodells gegeben, im unteren Teil das Schema *IfcRoofAndWallLayoutView*, das die Sicht definiert. Es enthält die mit (1) gekennzeichnete *Interface*-Deklaration, das die Klasse *Ifc-LayoutMemberStyle* (2) und dadurch implizit auch deren Oberklasse *IfcTypeProduct* (3) zur Benutzung benennt.

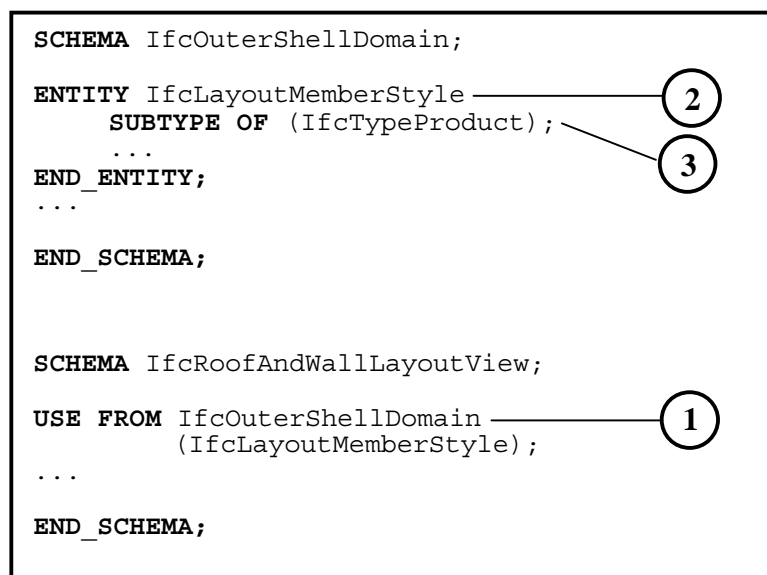


Abbildung 7.4: Sichtdefinition mit *Use-Interface* in EXPRESS

Die Vorteile dieser formalen Definition liegen in der Eindeutigkeit der Klassenbenennung, in der Nutzbarkeit des Schemas für die Schnittstellenprogrammierung und in der Möglichkeit, Austauschdateien maschinell auf die Zulässigkeit ihrer Dateninhalte zu prüfen.

Auf die IFC2x angewandt können auf diese Art zwar sämtliche verwendeten Klassen angesprochen werden, es kann jedoch nicht mit ausreichender Genauigkeit be-

stimmt werden, welche der generischen Datenstrukturen, die der Klasse durch Vererbung zur Verfügung stehen, konkret unterstützt werden sollen. So sind beispielsweise alle im Schema vorkommenden geometrischen Darstellungsarten implizit erfaßbar. Für die Implementierung einer konkreten Sicht ist es darüber hinaus jedoch notwendig, daß für jede Klasse diese Vielfalt an Möglichkeiten wieder auf das notwendige eingeschränkt wird. Da dies mit den *Interfaces* nicht möglich ist⁶, wird statt dessen eine tabellarische Festlegung vorgeschlagen.

Die Zahl der möglichen, generischen Datenstrukturen ist für die Leaf Node Classes der IFC2x sehr groß. Entsprechend umfangreich sind die notwendigen Konkretisierung dieser Möglichkeiten. In Tabelle 7.3 wird dies deshalb nur auszugsweise gezeigt, und zwar wiederum am Beispiel der Klasse *IfcLayoutMemberStyle* für die Sicht 2b.

In der ersten Spalte der Tabelle ist die vollständige Attributliste der Klasse aufgeführt, einschließlich der geerbten Attribute. In der zweiten Spalte folgen die in den Attributen referenzierten Datentypen und Klassen. Fett gedruckt sind die Konkretisierungen und Einschränkungen. Funktionen, die nicht ausdrücklich als optional ausgewiesen sind, sind zwingend erforderlich.

Die Attribute *HasAssociations* und *RepresentationsMaps* benutzen Verweisketten, innerhalb derer es vielfältige Verzweigungen gibt, die also ebenfalls konkretisiert werden müssen. Die weiterführenden Verweise werden in der Tabelle nicht ausführlich mit Attributen angegeben, sondern vereinfacht mit Pfeilen. Pfeile, die sich teilen, bedeuten, daß mehrere Varianten zur Darstellung möglich sind. Beim Datenimport muß ein Programm alle diese Varianten interpretieren können, beim Datenexport hat es dagegen die „Wahl“. Die Verweisketten werden in der Tabelle soweit verfolgt, bis die Festlegungen innerhalb der entsprechenden Klassendefinition selbst genügend konkret sind.

Die beiden Attribute *HasAssociations* und *RepresentationsMaps* sind sogenannte *inverse Attribute*. Das heißt, daß von ihnen keine Verweise ausgehen, sondern daß auf sie verwiesen wird. Sie tauchen in Austauschdateien nicht auf und dienen dazu, die Art und Anzahl der Instanzen zu beschränken, die auf eine Klasse verweisen können. Außerdem können Verweise mit ihnen rückwärts verfolgt werden.

Die in der Verweiskette des Attributes *RepresentationsMaps* aufgeführten Klassen wurden aus Platzgründen nicht in Kapitel 5 erläutert. Für ihr Verständnis wird deshalb auf die Online-Dokumentation der IFC2x verwiesen [59]. Das Prinzip dieser Art der Sichtdefinition wird jedoch auch ohne dieses Verständnis deutlich. Die Komplexität der Kette zeigt außerdem, wie unzureichend eine Sichtdefinitionen mit *Use-Interfaces* allein wäre.

⁶ Möglich wäre dies mit zusätzlichen *Domain-Rules* innerhalb der Klassendefinitionen im Datenmodell, also nicht auf Ebene der Sichten. In den IFC2x wird dies jedoch nicht praktiziert, zumindest nicht für geometrische Darstellungen.

Leaf Node Class, Attribute	Referenzierte Datentypen oder Klassen und Konkretisierung der verwendeten Datenstrukturen
	...
IfcLayout-MemberStyle:	Typdefinition für Verlegebauteile
<i>GlobalId:</i>	<i>IfcGlobalUniqueId</i> : Objektidentität
<i>OwnerHistory:</i>	<i>IfcOwnerHistory</i> : Besitz- und Benutzerstatus der Instanz
<i>Name:</i>	-
<i>Description:</i>	-
<i>HasAssociations:</i> (inverses Attribut)	<p><i>IfcRelAssociatesMaterial</i>: Optionale Zuordnung genau eines Bauproduktes</p> <p>↳ <i>IfcProductSpecification</i>: Bauprodukt. Sofern das Produkt Bestandteil der Leitschichtung ist, soll die dort verwendete Instanz referenziert werden.</p>
<i>ApplicableOccurrence:</i>	<i>IfcLabel</i> : Vorgeschriebener Wert = 'IfcLayoutMember'
<i>HasPropertySets:</i>	<i>IfcPropertySetDefinition</i> : Optionale Zuordnung von benutzerdefinierten Eigenschaften
<i>ObjectTypeOf:</i> (inverses Attribut)	<i>IfcRelDefinesByType</i> : Zuordnung dieser Typdefinition zu Bauelementen der Klasse <i>IfcLayoutMember</i>
<i>RepresentationMaps:</i>	<p><i>IfcRepresentationMap</i>: Zuordnung genau einer geometrischen Blockdefinition</p> <p>↳ <i>IfcAxis2Placement3D</i>: Räumliches Bezugskoordinatensystem der Blockdefinition</p> <p>↳ <i>IfcRepresentation</i>: Zuordnung genau einer geometrischen Darstellung</p> <p>↳ Variante 1: <i>IfcExtrudedAreaSolid</i>: In Längsrichtung gerader Extrusionskörper. Es können alle Profilformen des Schemas <i>IFCPROFILEEXTENSIONRESOURCE</i> sowie die Klasse <i>IfcArbitraryClosedProfileDef</i> verwendet werden.</p> <p>↳ Variante 2: <i>IfcSurfaceCurveSweptAreaSolid</i>: In Längsrichtung gekrümmter Extrusionskörper. Es können alle Profilformen des Schemas <i>IFCPROFILEEXTENSIONRESOURCE</i> sowie die Klasse <i>IfcArbitraryClosedProfileDef</i> verwendet werden.</p> <p>↳ Variante 3: <i>IfcCsgSolid</i>: Volumenkörper, der über boole'sche Operationen erzeugt wird.</p> <p>↳ <i>IfcBooleanClippingResult</i>: Boole'sche Operation</p> <p>↳ <i>IfcBooleanOperator</i>: Vorgeschriebener Wert = .DIFFERENCE.</p> <p>↳ Grundkörper der Operation. Darstellung mit einer der oben genannten Varianten 1 (<i>IfcExtrudedAreaSolid</i>), 2 (<i>IfcSurfaceCurveSweptAreaSolid</i>) oder 3 (<i>IfcCsgSolid</i>). Mit Variante 3 können mehrere Operationen geschachtelt werden.</p> <p>↳ Zweiter Körper, der vom Grundkörper abgezogen wird. Darstellung als <i>IfcExtrudedAreaSolid</i> mit freier Profilform (<i>IfcArbitraryClosedProfileDef</i>).</p>
<i>Tag:</i>	<i>IfcLabel</i> : Optionale Positionsnummer
<i>MemberType:</i>	<i>IfcLayoutMemberStyleEnum</i> : Art des Bauteils
	...

Tabelle 7.3: Ausführliche Sichtdefinition (Klasse *IfcLayoutMemberStyle*)

8 Zusammenfassung und Bewertung

8.1 Zusammenfassung

In der vorliegenden Arbeit wurde ein Produktmodellstandard entwickelt, mit dem Ingenieur-Software des Komplettbaus gekoppelt werden kann. Eine Voruntersuchung aus informationstechnischer Sicht, in der der Entwicklungsstand von Bau-Software, moderne Werkzeuge zur Produktmodellierung und vorhandene Produktmodelle des Bauwesens berücksichtigt wurde, lieferte die folgenden Ergebnisse:

- 1) Die Verbreitung von Anwendungs-Software ist seit den Untersuchungen von *Haller* in allen Bereichen des Bauwesens weiter fortgeschritten [47]. Dadurch ist auch der Bedarf weiter gestiegen, Produktdaten zwischen diesen Programmen auszutauschen.
- 2) Keines der untersuchten Produktmodelle des Bauwesens kann derzeit eine Kopplung der Anwendungsprogramme im Komplettbau leisten. Entweder sind die Anwendungsgebiete der Modelle zu beschränkt, oder ihre Detaillierungsgrade reichen nicht aus. Die in letzter Zeit sehr erfolgreichen, netzwerkbasierten IMS unterstützen zwar die Verwaltung und Verteilung der anfallenden Daten, auch sie ermöglichen jedoch keine Kopplung der Anwendungsprogramme. Es gibt deshalb keine Alternative zum Produktmodell als Mittel zur Integration.
- 3) Die IFC haben als einziges der betrachteten Produktmodelle berechnete Aussichten, eine universelle Bedeutung im Bauwesen zu erlangen. Aus diesem Grunde wurde beschlossen, daß das in dieser Arbeit zu entwickelnde Produktmodell mit deren aktueller Version IFC2x zu harmonisieren ist.
- 4) Die Modellierungs- und Implementierungsmethoden für Produktmodelle nach ISO 10303 sind nach wie vor Stand der Technik und wurden in dieser Arbeit verwendet. Aufgrund der unter Punkt 3) beschlossenen Harmonisierung mit den IFC schied jedoch die Unterstützung der Ressourcen nach ISO 10303 und damit die vollständige *STEP-Kompatibilität* aus.

In einer Voruntersuchung aus Ingenieursicht wurde festgelegt, welche Datenaustauschwege im Planungsprozeß des Komplettbaus vom Produktmodell abzudecken sind, und es wurde eine detaillierte Betrachtung der darzustellenden Gewerke ange stellt. Dadurch wurden bereits die Makrostruktur des Modells, die zu modellierenden Dateninhalte und die notwendigen Detaillierungsgrade vorbestimmt.

Anschließend wurde das Datenmodell der IFC2x im Hinblick auf die beabsichtigte Harmonisierung untersucht. Das Ergebnis war, daß die vorhandenen generischen Datenstrukturen für den Komplettbau zwar sehr gut geeignet sind, daß die konkreten Klassen zur Darstellung von Bauelementen und Bauteilen jedoch nicht ausreichen. Eine Harmonisierung konnte am besten erreicht werden, indem man das Komplettbaumodell als Erweiterung der IFC2x definierte, und zwar hauptsächlich in Form neuer Klassen für Bauelemente im *Interoperability Layer* und neuer Domänenmodelle im *Domain Layer*.

Die neuen Klassen wurden deshalb an geeigneter Stelle in den Vererbungsbaum der IFC2x eingeordnet. Bei der Definition ihrer Eigenschaften wurde so oft wie möglich auf die generischen Datenstrukturen der IFC zurückgegriffen. Diese Definitionen sind anschaulich und verkürzt in Kapitel 6 sowie formal und ausführlich in den Anhängen A und B zu finden. Die neuen Klassen bilden zusammen mit den relevanten

Bauelementen der IFC2x das Datenmodell, mit dem die im Komplettbau benötigten Sachverhalte dargestellt werden können.

Für dieses Datenmodell wurden schließlich vier Sichten entworfen, die die vertikale Kopplung der verschiedenen Anwendungsprogramme ermöglichen. Außerdem wurde eine fünfte Sicht definiert, die zusätzliche Konformitätsanforderungen an zentrale CAD-Programme stellt und dadurch auf einfache Art und Weise die horizontale Integration von Teilprozessen ermöglicht. Alle diese Sichten sind für den Einsatz von Austauschdateien ausgelegt.

8.2 Bewertung der Ergebnisse

Mit dem aus dem Datenmodell und den verschiedenen Sichtdefinitionen bestehenden Produktmodell existiert erstmals ein ganzheitlicher Standard zum Austausch von Ingenieurdaten für die Gewerke des Komplettbaus. Durch die IFC-Kompatibilität ist er in den Zusammenhang des gesamten Bauwesens eingebunden, und zwar mit internationaler Gültigkeit. Es besteht insbesondere die Möglichkeit zur Kopplung mit dem im ST-4 Projekt der IAI entwickelten Modell für Stahlkonstruktionen und bei Bedarf auch mit dem für die Statik. Das Datenmodell ist nach dem aktuellen Stand der Technik modelliert und damit auf die Nutzung der modernen Informations- und Kommunikationstechnologien vorbereitet.

Durch die Verwendung der generischen Datenstrukturen der IFC übernimmt das Komplettbaumodell jedoch nicht nur dessen Vorzüge, sondern auch seine Nachteile:

- 1) Das Datenmodell ist sehr umfangreich und komplex. Der Aufwand für die eigentliche Schnittstellenprogrammierung wird durch die Nutzung der Vererbungsstrukturen und bei Einsatz moderner Programmierwerkzeuge noch begrenzt. Es bleibt jedoch der Aufwand für die Einarbeitung des Programmierers in das Modell.
- 2) Jede Klasse im Datenmodell kann sehr viele, von ihren Oberklassen geerbte generische Datenstrukturen verwenden. Die Schwierigkeit bei der Konkretisierung, welche dieser Strukturen für eine Sicht tatsächlich zu unterstützen sind, wurde bereits in Kapitel 7.4 deutlich, insbesondere im Bezug auf die Geometrie. Wünschenswert wäre die Möglichkeit zur vollständigen Spezifikation in formaler Art und Weise.

Ein weiterer, kritisch zu bewertender Punkt ist unabhängig von den IFC die Anwendung des Komplettbaumodells: Das den Sichtdefinitionen aus Kapitel 7 zugrundeliegende Anwendungsszenario zur rein vertikalen Systemkopplung wäre genauso möglich, wenn die Weitergabe der Daten aus den CAD-Systemen der Fachplanungen über herkömmliche Schnittstellen bewerkstelligt würde. Da solche Schnittstellen zum Teil bereits existieren, könnte man sich die Implementierung der Domänenmodelle streng genommen ersparen. Der volle Nutzen des Produktmodells dieser Arbeit wird also erst bei Anwendungsszenarien auftreten, die stärker auf eine horizontale Integration der Planungsschritte abzielen. Dies wäre zum Beispiel bei der Kopplung von gleichartigen CAD-Programmen oder bei stärker ausgeprägter, zentraler Koordinierung der Fachplanungen der Fall.

Dennoch, das Ziel dieser Arbeit wurde erreicht: Auf der Grundlage der in Kapitel 7 entworfenen, pragmatischen Sichten können die kurzfristig realistischen Anwendungen des Komplettbaumodells durchgeführt werden. Durch das Prinzip der Trennung von Datenmodell und Sichtdefinition ist es darüber hinaus möglich, bei Bedarf neue Sichten für weitergehende Anwendungsszenarien zu schaffen, die mittelfristig realisiert werden können.

9 Ausblick

9.1 Umsetzung der Ergebnisse

Zur unmittelbaren Umsetzung der Ergebnisse dieser Arbeit muß das Produktmodell in die entsprechenden Gremien der IAI eingereicht werden. Dazu müssen die Klassendefinitionen in das Format der Online-Dokumentation der IFC übertragen werden [59]. Da in diesem Stadium keine endgültige Version erforderlich ist, kann der Schwerpunkt auf der Beschreibung der Klassen und ihrer Anwendung in Textform liegen, und zwar in englischer Sprache. Formale Darstellungen, wie die vollständigen EXPRESS-G-Diagramme oder die Vererbungs bäume, können dagegen zu einem späteren Zeitpunkt mit spezieller Software automatisch aus dem EXPRESS-Schema erzeugt werden. Nach Durchlaufen einer Rezension innerhalb der IAI und gegebenenfalls daraufhin durchgeführten Anpassungen erhält das Modell einen offiziellen Status. Es ist dann in den Gesamtzusammenhang der IFC eingebunden und kann von Software-Herstellern implementiert werden.

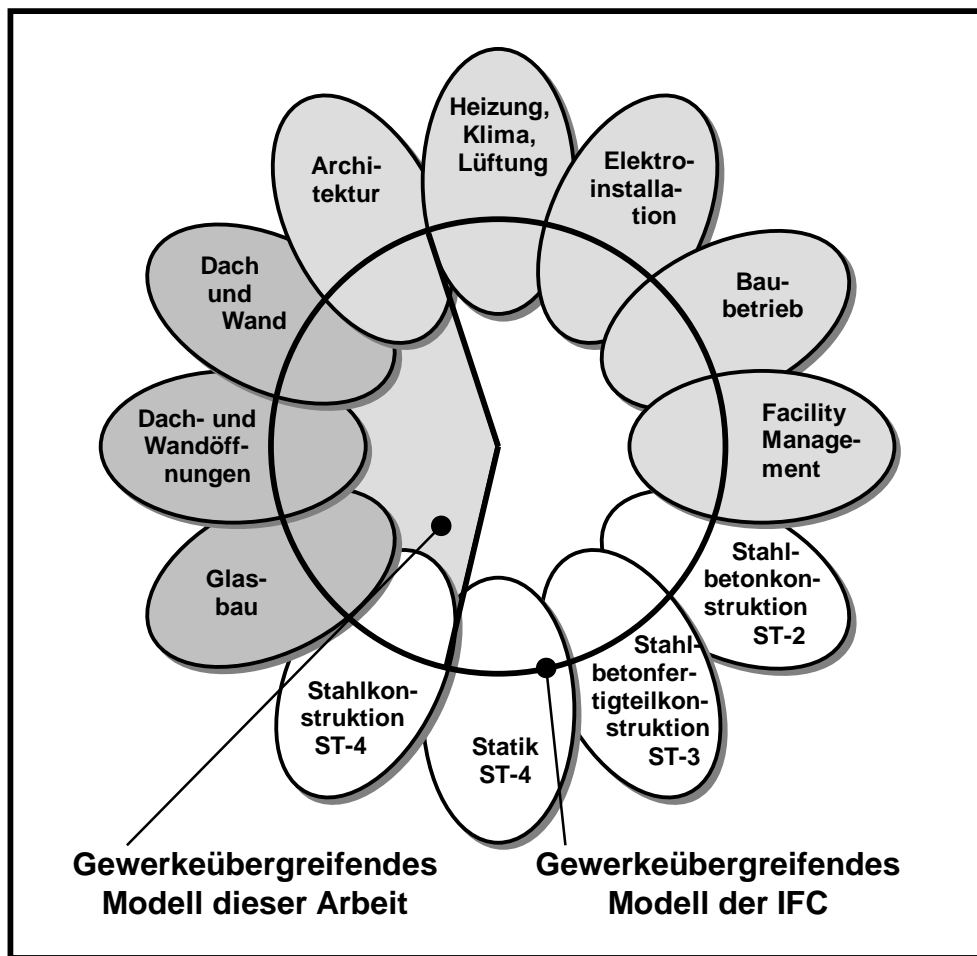


Abbildung 9.1: Erweitertes Datenmodell der IFC

Abbildung 9.1 zeigt das derart erweiterte Modell der IFC. Die Ellipsen symbolisieren die Domänenmodelle, wobei die in dieser Arbeit entwickelten dunkelgrau dargestellt sind. Die in den IFC2x bereits vorhandenen Domänen sind dagegen hellgrau

und die in den Erweiterungsprojekten ST-2, ST-3 und ST-4 derzeit in Entwicklung befindlichen weiß dargestellt. Für eine konsequente Weiterführung der Systemintegration in Richtung Schlüsselfertigbau sind besonders die Domänen für Heizung, Klima und Lüftung sowie für die Elektroinstallation von Bedeutung. Der Bereich des *Facility Managements* ist ebenfalls durch eine entsprechende Domäne vertreten. Zur Kopplung mit Software aus diesen Bereichen müssen neue Sichten innerhalb des erweiterten, gewerkeübergreifenden Modells der IFC definiert werden.

9.2 Längerfristige Ziele

Die längerfristigen Ziele der Produktmodellierung innerhalb einer ganzheitlichen Planung können zu den folgenden drei Punkten zusammengefasst werden.

Modellierung neuer Anwendungsbereiche:

Abbildung 9.2 zeigt die Betätigungsfelder eines großen, modernen Baukonzerns nach *Schroeder* [107], und zwar von unten nach oben mit steigender Rentabilität. Neben den klassischen Aufgaben der Ingenieurplanung und Bauausführung sind dies immer mehr die Bereiche der Abwicklung und der Entwicklung von Projekten. Mit Systemführung ist außerdem der Betrieb von Bauwerken gemeint, an dem moderne Baukonzerne als Mitglieder in Betreibergesellschaften zunehmend teilhaben. Das langfristige Ziel ist es, alle Betätigungsfelder miteinander zu koppeln. Die Betätigungen, für die das Produktmodell dieser Arbeit dies bereits leistet, sind mit schwarzen Haken markiert. Die mit grauen Haken gekennzeichneten Betätigungen sind dagegen durch die übrigen Domänen der gemäß Abbildung 9.1 erweiterten IFC abgedeckt.

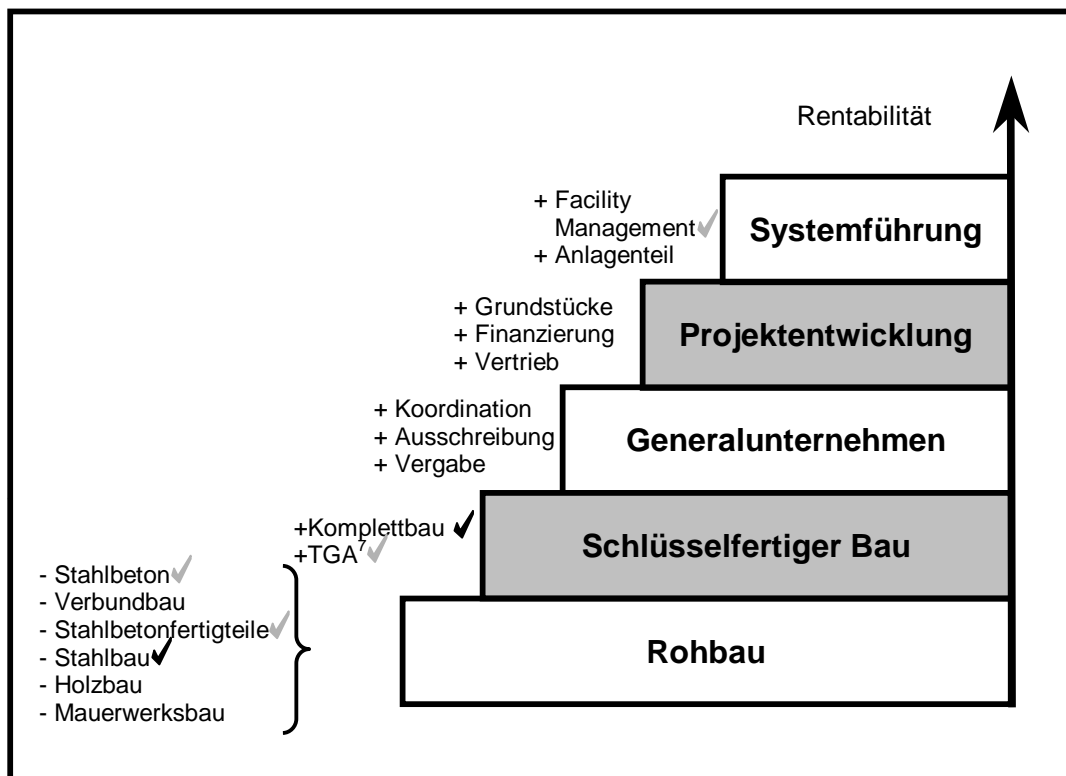


Abbildung 9.2: Betätigungsfelder großer Baukonzerne [107]

⁷ TGA: Technische Gebäudeausrüstung

Integration der Baubehörden:

Die bisher behandelten Anwendungsszenarien betreffen ausschließlich Baufirmen und Ingenieurbüros. Es ist jedoch sinnvoll, auch die Baubehörden in den Datenaustausch mit einzubeziehen. Dabei stellt sich das Problem, daß der herkömmliche Ablauf der Genehmigungsverfahren auf Papierzeichnungen basiert, die Informationsmedium und Dokument in einem sind. Im Gegensatz dazu ist es der ausdrückliche Zweck eines Produktmodells, den Datenbestand eines Projektes für jeden Planer und jede Applikation zu öffnen, so daß der Dokumentencharakter zwangsläufig verloren geht. Herkömmliche Zugriffsbeschränkungen in Form von Benutzerrechten greifen zu kurz, da genehmigte Daten beispielsweise auch von ihren Erzeugern nachträglich nicht mehr manipuliert werden dürfen.

Anwendung des Produktmodells:

Eines der größten, noch ungelösten Probleme stellt sich bei der Anwendung eines so umfassenden Produktmodells, wie es die gemäß Abbildung 9.1 erweiterten IFC sind. Nicht alle Funktionen zur Wahrung der Datenkonsistenz und zur Lösung von Planungskonflikten können von der Anwendungs-Software geleistet werden, so wie es in den Sichten in Kapitel 7 dieser Arbeit vorgesehen ist. Der statt dessen lange Zeit propagierte Einsatz zentraler Datenbanken, die ein Projekt in Form eines einzigen, der Struktur des Produktmodells entsprechenden Datenbestandes abbilden, hat aufgrund des Umfanges solcher Datenbanken wenig Anklang in der Baupraxis gefunden. Eine interessante Alternative dazu ist die Nutzung der in letzter Zeit sehr erfolgreichen Technologie der IMS. Durch die dabei praktizierte Handhabung der Daten in kleinen Einheiten, in der Regel vom Anwender selbst erzeugte Dokumenten- oder Zeichnungsdateien, können diese Systeme individuell und flexibel an die Bedürfnisse der Anwender angepaßt werden. Die Frage in diesem Zusammenhang ist, ob und wie sich die derzeit voneinander unabhängigen Bestrebungen der Produktmodellierung und der IMS vereinen lassen. Lösungsvorschläge hierfür existieren bereits, wie der von *Rezgui e.a.* präsentierte Ansatz [96], [97]; dennoch besteht noch Forschungsbedarf.

10 Schrifttum

- [1] AFNOR: Automatisation industrielle; Représentation externe des donnée de définition des produits; Spécification du standard d'échange et de transfert (SET), Version 85-08; Z68-300, Association française de normalisation (afnor), Paris, France; 1985
- [2] ARCHmatic-Glossar und –Lexikon; Online-Glossar zu den Themen EDV und Internet; <http://www.glossar.de/>
- [3] Armstrong, S. G., Lockley, S. R.: Modelling of Generic Document Structures and the Development of an Integrated Document Data Environment; Proceedings of the 1st European Conference on Product and Process Modelling in the Building Industry ECPPM'94, Dresden; 1994; S. 19 - 28
- [4] Baehre, R.: Raumabschließende Bauelemente; Stahlbau Handbuch, Band1 Teil A, Dritte, neu bearbeitete Auflage; Stahlbau-Verlagsgesellschaft, Köln; 1993; S. 901 – 940
- [5] Banke, F.: Vergleich der Produktschnittstelle Stahlbau mit dem Logical Product Model des CIMSteel-Projects, Vertieferarbeit; Universität Karlsruhe (TH), Karlsruhe; 1996
- [6] Bauen mit Stahl: Stahlbau Arbeitshilfen, Stand Januar 97; Bauen mit Stahl e.V., Düsseldorf; 1997
- [7] Björk, B.-C.: Document Management – a Key Technology for the Construction Industry; Proceedings of the Second Worldwide ECCE Symposium – Information and Communication Technology (ICT) in the Practice of Building and Civil Engineering, Espoo, Finland; 2001; S. 35 – 43
- [8] Bretschneider, D.: Modellierung rechnerunterstützter, kooperativer Arbeit in der Tragwerksplanung, Fortschr.-Ber. VDI Reihe 4 Nr.151; Düsseldorf, VDI-Verlag; 1998
- [9] Bucak, Ö.: Glas im konstruktiven Ingenieurbau; Stahlbau-Kalender 1999; Ernst und Sohn, Berlin; 1999; S.515 – 643
- [10] Charlier, H.: Bauaufsichtliche Anforderungen an Glaskonstruktionen; Der Prüflingenieur, Heft 1997; 1997; S. 44 – 54
- [11] CIMSteel Integration Standards - Release One (CIS/1.0); Computer Aided Engineering Group, Department of Civil Engineering, The University of Leeds, Leeds, England; 1995
- [12] CIMSteel Integration Standards - Release Two (CIS/2); The Steel Construction Institute, Berkshire, England; 2000; Kostenlose Vorschau unter: <http://www.cis2.org/>
- [13] Cziesielski, E., Maerker, B.: Bauphysikalisches Verhalten von Stahl-Kassettenwänden; Der Stahlbau 4 (1982); 1982; S. 109 – 115
- [14] DASt: DASt-Richtlinie 016 - Bemessung und konstruktive Gestaltung von Tragwerken aus dünnwandigen kaltgeformten Bauteilen, 2. überarbeitete Auflage; Deutscher Ausschuß für Stahlbau (DASt); Stahlbau-Verlagsgesellschaft, Köln; 1992

- [15] DFG: Schwerpunktprogramm „Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau“; Deutsche Forschungsgemeinschaft (DFG); Übersicht der Projekte unter:
<http://www.iib.bauing.tu-darmstadt.de/dfg-spp1103/index.html>
- [16] DIBt: Technische Regeln für die Verwendung von linienförmig gelagerten Überkopfverglasungen; Mitteilungen des DIBt, Heft 5/1996; Ernst und Sohn, Berlin; 1996
- [17] DIBt: Technische Regeln für die Verwendung von linienförmig gelagerten Vertikalverglasungen; Mitteilungen des DIBt, Heft 4/1997; Ernst und Sohn, Berlin; 1997
- [18] DIN EN 1013: Lichtdurchlässige profilierte Platten aus Kunststoff für einschalige Dacheindeckungen;
Teil 1: Allgemeine Anforderungen und Prüfverfahren; Ausgabe Januar 1998
Teil 2: Besondere Anforderungen und Prüfmethoden für Platten aus glasfaserverstärktem Polyesterharz (GF-UP); Ausgabe März 1999
Teil 3: Besondere Anforderungen und Prüfmethoden für Platten aus Polyvinylchlorid (PVC); Ausgabe Januar 1998
Teil 4: Besondere Anforderungen, Prüfverfahren und -verhalten für Platten aus Polycarbonat (PC); Ausgabe Februar 2000
Teil 5: Besondere Anforderungen, Prüfverfahren und -verhalten für Platten aus Polymethylmethacrylat (PMMA); Ausgabe Februar 2000
- [19] DIN V ENV 1627: Fenster, Türen, Abschlüsse - Einbruchhemmung - Anforderungen und Klassifizierung; Ausgabe April 1999
- [20] DIN EN 12433: Tore – Terminologie;
Teil 1: Bauarten von Toren; Ausgabe Februar 2000
Teil 2: Bauteile von Toren; Ausgabe Februar 2000
- [21] DIN EN 12519 (Entwurf): Türen und Fenster - Terminologie; Dreisprachige Fassung, Ausgabe November 1996
- [22] DIN 18055: Fenster – Fugendurchlässigkeit, Schlagregendichtigkeit und mechanische Beanspruchung, Anforderung und Prüfung;
Ausgabe Oktober 1981
- [23] DIN 18056: Fensterwände – Bemessung und Ausführung;
Ausgabe Juni 1966
- [24] DIN 18095: Türen – Rauchschutztüren;
Teil 1: Begriffe und Anforderungen; Ausgabe Oktober 1988
Teil 2: Bauartprüfung der Dauerfunktionstüchtigkeit und Dichtheit;
Ausgabe März 1991
Teil 3: Anwendungen von Prüfergebnissen; Ausgabe Juni 1999
- [25] DIN 18323: Baulicher Brandschutz – Rauch- und Wärmeabzugsanlagen;
Teil 1: Begriffe und Anwendungen; Ausgabe September 1981
Teil 2: Rauchabzüge - Bemessung, Anforderungen und Einbau;
Ausgabe November 1989
Teil 3: Rauchabzüge – Prüfungen; Ausgabe September 1984

- [26] DIN 18516: Außenwandbekleidungen, hinterlüftet;
Teil 1: Anforderungen, Prüfgrundsätze; Ausgabe Dezember 1999
Teil 3: Naturwerkstein; Anforderungen, Bemessung;
Ausgabe Dezember 1999
Teil 4: Einscheiben-Sicherheitsglas; Anforderungen, Bemessung, Prüfung;
Ausgabe Februar 1990
Teil 5: Betonwerkstein; Anforderungen, Bemessung;
Ausgabe Dezember 1999
- [27] DIN 18531: Dachabdichtungen - Begriffe, Anforderungen, Planungsgrundsätze; Ausgabe September 1991
- [28] DIN 18807: Trapezprofile im Hochbau;
Teil 1: Stahltrapezprofile; Allgemeine Anforderungen, Ermittlung der Tragfähigkeitswerte durch Berechnung; Ausgabe Juni 1987
Teil 2: Stahltrapezprofile; Durchführung und Auswertung von Tragfähigkeitsversuchen; Ausgabe Juni 1987
Teil 3: Stahltrapezprofile; Festigkeitsnachweis und konstruktive Ausbildung; Ausgabe Juni 1987
Teil 6: Aluminium-Trapezprofile und ihre Verbindungen; Ermittlung der Tragfähigkeitswerte durch Berechnung; Ausgabe September 1995
Teil 7: Aluminium-Trapezprofile und ihre Verbindungen; Ermittlung der Tragfähigkeitswerte durch Versuche; Ausgabe September 1995
Teil 8: Aluminium-Trapezprofile und ihre Verbindungen; Nachweise der Tragsicherheit und Gebrauchstauglichkeit; Ausgabe September 1995
Teil 9: Aluminium-Trapezprofile und ihre Verbindungen; Anwendung und Konstruktion; Ausgabe Juni 1998
- [29] DStV: Materialbezeichnung für den Datenaustausch im Stahlbau, Teil 1: Halbzeuge und Verbindungsmittel, Version März 1998; Empfehlungen des DSTV-Arbeitsausschusses EDV, Düsseldorf; 1998
- [30] DStV: Blockdarstellung der EDV im Stahlbau – Standardbeschreibung von Stahlbau-Teilen für die NC-Steuerung, Version Juli 1998; Empfehlungen des DSTV-Arbeitsausschusses EDV, Düsseldorf; 1998
- [31] DStV: Blockdarstellung der EDV im Stahlbau – Schnittstellenbeschreibung CAD - Stückliste, Version September 1999; Empfehlungen des DSTV-Arbeitsausschusses EDV, Düsseldorf; 1999
- [32] DStV: Standardbeschreibung Produktschnittstelle Stahlbau, Version 4/2000 - Teile 1 bis 3; Arbeitsausschuß EDV des Deutschen Stahlbau-Verbandes (DStV), Düsseldorf; 2000;
Zu beziehen unter:<http://www.deutscherstahlbau.de>
- [33] DStV: Erhebung EDV-Systeme im Stahl- und Komplettbau; Deutscher Stahlbau-Verband (DStV), Düsseldorf; 2000
- [34] ECCS: European Recommendations for Steel Construction: The Stressed Skin Design of Steel Structures; ECCS Technical Committee 7, European Convention for Constructional Steelwork; 1978
- [35] ECCS: European Recommendations for Steel Construction: The Design of Profiled Sheeting; ECCS Technical Committee 7, European Convention for Constructional Steelwork; 1983

- [36] ECCS: Europäische Empfehlungen für das Bauen mit Fassaden- und Dach-
elementen aus Stahl: Dünnwandige kaltgeformte Stahlbleche im Hochbau;
EKS Technische Kommission 7, Europäische Konvention für Stahlbau;
1985
- [37] ECCS: Preliminary Recommendations for Sandwich Panels - Part I: Design,
Part II: Good Practice; ECCS Technical Committee 7, Working Group 7.4,
European Convention for Constructional Steelwork; 1991
- [38] Egner, R.: Zum Tragverhalten von Fassadenelementen aus Aluminium bei
Windbelastung, Dissertation; Universität Karlsruhe (TH), Karlsruhe; 1993
- [39] ENV 1090, Part 2: Execution of Steel Structures - Supplementary Rules for
Cold Formed and Thin Gauge Components and Sheeting; Release July 1998
- [40] ENV 1993-1-3: Eurocode 3: Bemessung und Konstruktion von Stahlbauten,
Teil 1.3: Allgemeine Bemessungsregeln – Ergänzende Regeln für kaltge-
formte dünnwandige Bauteile und Bleche; Ausgabe 1996
- [41] EOTA: Guideline for European Technical Approval for Structural Sealant
Glazing Systems (SSGS), Draft ETAG n° 00X, WG 04.04/13 – Structural
Sealant Glazing, European Organisation for Technical Approvals (EOTA),
Brussels, Belgium; Release October 1997
- [42] Fischer, M., Liston, K: Information Modeling for Construction Project
Teams; Additional Paper to the Proceedings of the Second Worldwide
ECCE Symposium – Information and Communication Technology (ICT) in
the Practice of Building and Civil Engineering, Espoo, Finland; 2001;
- [43] GAEB: Regelungen für Informationen im Bauvertrag – Aufbau Leistungs-
verzeichnis und GAEB-Datenaustausch 2000, Version 1.0; Gemeinsamer
Ausschuß Elektronik im Bauwesen (GAEB); Beuth Verlag, Berlin; 1999
- [44] Grabowski, H., Anderl, R., Schilli, B., Schmitt, M.: STEP – Entwicklung
einer Schnittstelle zum Produktdatenaustausch; VDI-Z 131,
Ausgabe September 1989; 1989; S. 68 - 76
- [45] Haas, W.: CAD-Datenaustauschkonzepte, STEP-2DBS für Architekten und
Ingenieure; Springer-Verlag, Berlin; 1993
- [46] Hahn, D., Huhn, M.: Von der Einzellösung zur Integration; Der Stahlbau 68
(1999), Heft 10; 1999; S. 785-790
- [47] Haller, H.-W.: Ein Produktmodell für den Stahlbau, Dissertation; Universi-
tät Karlsruhe (TH), Karlsruhe; 1994
- [48] Haller, H.-W.: Projektschnittstelle Holzbau, Version 1.01; Universität Stutt-
gart, Stuttgart; 1994
- [49] Haller, H.-W., Osterrieder, P. Saal, H.: Product Models for Structural
Steelwork; Proceedings of the Sixth International Conference on Computing
in Civil Engineering; Baalkema, Rotterdam, The Netherlands; 1995;
S. 419 – 427
- [50] Hannus, M., Karstila, K.: Requirements on Standardised Building Product
Data Models; Proceedings of the 1st European Conference on Product and
Process Modelling in the Building Industry ECPPM'94, Dresden; 1994;
S. 43 – 50

- [51] Hartmann, D.: Grundlegende Betrachtungen zur Anwendung der Objektorientierung; Deutsche Forschungsgemeinschaft (DFG), Forschungsbericht Objektorientierte Modellierung in Planung und Konstruktion; Wiley-VCH, Weinheim; 2000; S.1 – 25
- [52] Hartmann, D.: Vorwort zum Forschungsbericht Objektorientierte Modellierung in Planung und Konstruktion; Deutsche Forschungsgemeinschaft (DFG); Wiley-VCH, Weinheim; 2000; S. XVII - XX
- [53] Heinze Bau-Office; Bau-Informationssystem der Heinze GmbH;
<http://heinze.baunetz.de>
- [54] Hörenbaum, C., Saal, H.: Konformitätsprüfung der Produktschnittstelle Stahlbau; Im Auftrag des Deutschen Stahlbau-Verbandes (DStV) entwickeltes Verfahren; Universität Karlsruhe (TH), Karlsruhe; 2002; Voraussichtlich einzusehen unter <http://www.deutscherstahlbau.de>
- [55] Hörenbaum, C., Saal, H.: Integration of Product Interface Steel Construction and Industry Foundation Classes; Proceedings of the Second Worldwide ECCE Symposium – Information and Communication Technology (ICT) in the Practice of Building and Civil Engineering, Espoo, Finland; 2001; S. 131 – 136
- [56] Hörenbaum, C., Liebich, Th., Weise, M.: Model Extensions of IFC2x, Status Report on ST-4 Project; International Alliance for Interoperability (IAI); Aktuelle Versionen unter: <http://cib.bau.tu-dresden.de/icss/structural>
Volume I: Resource Extensions
Volume II: Steel Construction Domain
Volume III: Structural Analysis Domain
- [57] Hörenbaum, C., Saal, H.: Weiterentwicklung der Produktschnittstelle Stahlbau für die Anforderungen im Komplettbau; Abschlußbericht zum gleichnamigen AiF-Forschungsprojekt FV-12386N; Universität Karlsruhe (TH), Karlsruhe; 2002
- [58] IAI: Industry Foundation Classes (IFC) – Release 1.5; International Alliance for Interoperability (IAI); 1997; Online-Dokumentation zu beziehen unter: <http://iaiweb.lbl.gov/>
- [59] IAI: Industry Foundation Classes (IFC) – Release 2x Platform; International Alliance for Interoperability (IAI); 2000; Online-Dokumentation zu beziehen unter: <http://iaiweb.lbl.gov/>
- [60] IAI: IFC2x Extension Modelling Guide; International Alliance for Interoperability (IAI); 2001; <http://iaiweb.lbl.gov/>
- [61] IAI: IFC2x Implementation Guide; International Alliance for Interoperability (IAI); 2001; <http://iaiweb.lbl.gov/>
- [62] IAI: Verzeichnis der aktuellen Erweiterungsprojekte für die IFC; International Alliance for Interoperability (IAI);
http://cic.vtt.fi/niai/technical/IFC_3/R3_projects.html
- [63] IAI: AecXML; International Alliance for Interoperability (IAI); 2001;
<http://www.aecxml.org>
- [64] IAI: XML Schema Language Binding of EXPRESS for ifcXML; International Alliance for Interoperability (IAI); 2001; <http://iaiweb.lbl.gov/>
- [65] IAI: The ifcXML Schemas; International Alliance for Interoperability (IAI); 2001; <http://iaiweb.lbl.gov/>

- [66] IFBS: Stahltrapezprofile im Hochbau; Institut zur Förderung des Bauens mit Bauelementen aus Stahlblech e.V. (IFBS), Düsseldorf; 1980
- [67] IFBS: Stahlkassettenprofile für Dach und Wand, IFBS-Info 3.01; Industrieverband zur Förderung des Bauens mit Stahlblech e.V. (IFBS), Düsseldorf; 1999
- [68] IFBS: Stahltrapezprofile für Dach, Wand und Decke, IFBS-Info 3.02; Industrieverband zur Förderung des Bauens mit Stahlblech e.V. (IFBS), Düsseldorf; 2000
- [69] IFBS: Stahl-PUR-Sandwichelemente für Dach und Wand, IFBS-Info 3.03; Industrieverband zur Förderung des Bauens mit Stahlblech e.V. (IFBS), Düsseldorf; 1999
- [70] IFBS: Stahl-Sonderprofile für Dach, Wand und Decke, IFBS-Info 3.04; Industrieverband zur Förderung des Bauens mit Stahlblech e.V. (IFBS), Düsseldorf; 1999
- [71] IFBS: Zulassungsbescheid Verbindungselemente zur Verwendung bei Konstruktionen mit „Kaltprofilen“ aus Stahlblech - insbesondere bei Stahlprofiltafeln, IFBS-Info 7.01; Industrieverband zur Förderung des Bauens mit Stahlblech e.V. (IFBS), Düsseldorf; 1999
- [72] ISO 8879: Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML); 1986;
Zu beziehen unter <http://www.iso.ch/>
- [73] ISO 10303: Industrial Automation Systems and Integration - Product Data Representation and Exchange; Zu beziehen unter: <http://www.iso.ch/>
- [74] ISO 10303: Step on a Page; Übersicht zu ISO 10303 und aktuelles Verzeichnis aller Teile; Stand: 06/2001; <http://www.mel.nist.gov/sc5/soap/>
- [75] ISO CD 10303, Part 225 - Application Protocol: Building Elements Using Explicit Shape Representation; TC 184/SC4 N338; 1995
- [76] Jeckle, M.: Übersicht und Links zum Thema XML-Schema;
<http://www.jeckle.de/xml/schema.html>
- [77] Kech, J., Ladwein, Th.: Anleitung zur Bemessung von Sandwichelementen, Entwurf; Industrieverband zur Förderung des Bauens mit Stahlblech e.V. (IFBS), Düsseldorf; 1992
- [78] Klinkenberg, A.: Computergestützte plastische Bemessung von Anschlüssen im Stahlbau, Dissertation; Universität Karlsruhe (TH), Karlsruhe; 1998
- [79] Krüner, H.: Wissensbasierte Ermittlung statischer Systeme aus CAD-Daten von Stahlbaukonstruktionen mit interaktiver Eingriffsmöglichkeit, Dissertation; Ruhr-Universität Bochum, Bochum; 1987
- [80] Kutsch, P., Pegels, G.: Wettbewerbsfähigkeit und CAD-Konstruktion; Der Stahlbau 66 (1997), Heft 5; 1997; S. 276-279
- [81] Liebich, Th., Adachi, Y.: Exchanging IFC Content Information using the XML-Protocol; Proceedings of the 3rd European Conference on Product and Process Modelling in Building and Construction ECPPM 2000, Lissabon, Portugal; 2000; S. 33 - 38
- [82] Liebich, Th.: IFC – Die nächsten Schritte; Bauinformatik-Journal, Heft 3/2001; 2001; S. 11 – 15;
Auszugsweise einzusehen unter <http://www.iai-ev.de/>

- [83a] Lubinski, F.: Die Stahl-Kassettenwand, Bauphysikalisches Verhalten; Der Bauingenieur 57 (1982); 1982
- [83b] Loose, T., Saal, H.: Ermittlung der Wärmeverluste an zweischaligen Wand-aufbauten in Leichtbauweise; Voraussichtlich 2002
- [84] Meißner, U., F., Möller, B., Diaz, J., Rüppel, U., Peters, F., Kirchner, V., Petersen, M.: Objektorientierte Tragwerksmodelle für die Systemintegration von Planungs- und Konstruktionsvorgängen im Bauwesen; Deutsche Forschungsgemeinschaft (DFG), Forschungsbericht Objektorientierte Modellierung in Planung und Konstruktion; Wiley-VCH, Weinheim; 2000; S.266 – 286
- [85] Mitchell, W.J.: Computer-Aided Architectural Design; School of Architecture and Urban Planning, University of California, Los Angeles, USA; 1977
- [86] NBS: The Content, Plan and Schedule of the First Version of the Product Data Exchange Specification (PDES), Internes IGES-Papier; National Bureau of Standards, Gaithersburg, USA; 1985
- [87] NBS: Initial Graphics Exchange Specification (IGES), Version 4.0; National Bureau of Standards, Washington DC, USA; 1988
- [88] NIST: Integration Definition for Function Modeling (IDEF0), Processing Standards Publication 183; National Institute of Standards and Technology (NIST), USA, 1993;
Zu beziehen unter: <http://www.qd.cl/WEBPAGES/IDEF/IDEF0/idef0.html>
- [89] Osterrieder, P., Haller, H.-W., Saal, H.: Bauspezifische, fertigungsorientierte Produktmodellierung im Stahlbau und Holzbau; Der Bauingenieur 72 (1997); 1997; S. 489-496
- [90] Pegels, G.: Leistungsstärke durch Informations- und Kommunikationstechnologie – erfolgreiche Strategien von Konstruktions- und Ingenieurbüros; Der Stahlbau 68 (1999), Heft 10; 1999; S. 777 - 784
- [91] Peter, W.: Untersuchung zur Berechnung des Wärmewiderstandes von zweischaligen Dach- und Wandkonstruktionen in Stahlleichtbauweise, Diplomarbeit; Universität Karlsruhe (TH), Karlsruhe; 1998
- [92] Pflugbeil, M.: CAD – Werkzeug des Architekten; Vieweg, Braunschweig; 1995
- [93] PMI: Project Management Software Survey; The Project Management Institute (PMI), Philadelphia, USA; 1999; Kontakt unter: <http://pmi.org>
- [94] RAL: Bauelemente aus Stahlblech - Gütesicherung RAL - RG 617; Gütegemeinschaft Bauelemente aus Stahlblech e.V., Deutsches Institut für Gütesicherung und Kennzeichnung e.V., Sankt-Augustin; 1997
- [95] Rezmann, Stefan: Einsatz moderner Informations- und Kommunikationstechnologie bei der Projektkoordination und im Baustellenbereich, Vertiefearbeit; Institut für Siedlungswasserwirtschaft der Universität Karlsruhe (TH), Karlsruhe; 1998
- [96] Rezgui, Y., Debras, P.: An Integrated Approach for a Model-Based Document Production and Management; Electronic Journal of Information Technology in Construction (ITcon), Vol.1; 1996; S. 1 – 24;
Zu beziehen unter: <http://www.itcon.org/1996/1/>

- [97] Rezgui, Y., Cooper, G.: A Proposed Open Infrastructure for Construction Project Document Sharing; *Electronic Journal of Information Technology in Construction (ITcon)*, Vol.3; 1998; S. 11 – 24;
Zu beziehen unter: <http://www.itcon.org/1998/2/>
- [98] Rice, P., Dutton, H.: *Structural Glass*; London, England; 1995
- [99] Rumbaugh, J., Booch, G.: *Das UML-Benutzerhandbuch (1. Auflage der deutschen Übersetzung)*; Addison-Wesley, Bonn; 1999
- [100] Saal, H., Haller, H.-W.: Feature Based Product Model for Steel Structures; *Proceedings of the 1st European Conference on Product and Process Modelling in the Building Industry ECPPM'94*, Dresden; 1994; S. 155 - 162
- [101] Schäufele, N.: Anbindung des „Programmsystems zur Konstruktion im Stahlbau“ PKS an das Logical Product Model des CIMSteel-Projektes, Diplomarbeit; Universität Karlsruhe (TH), Karlsruhe; 1995
- [102] Scherer, R.J., Katranuschkov, P.: Integration sollte mehr sein als reiner Datenaustausch; Beitrag zur Festschrift für Prof. Mang und Prof. Steinhardt, Universität Karlsruhe (TH), Karlsruhe; 1994; S. 749 - 762
- [103] Schiffer, J., Pegels, G.: Erschließung neuer Märkte durch CAD-Konstruktion – Erfahrungen eines Konstruktionsbüros; *Der Stahlbau* 66 (1997), Heft 2; 1997; S. 85-86
- [104] Schittich, C., Staib, G., Balkow, D., Schuler, M., Sobek, W.: *Glasbau Atlas*; Institut für internationale Architektur-Dokumentation GmbH, München; 1998
- [105] Schmitt, G.: *Architektur mit dem Computer*; Vieweg Verlag, Braunschweig; 1996; <http://caad.arch.ethz.ch/projects/acm>
- [106] Schnellenbach-Held, M., Denk, H., Albert, A.: Agent-Based Virtual Market Place for AEC-Bidding; *Darmstadt Concrete*, V15; Eigenverlag, Darmstadt; 2000; S 111-118
- [107] Schroeder, F.: Aspekte des IT-Managements im globalen Umfeld; Vortrag am Institut für Maschinenwesen im Baubetrieb, Universität Karlsruhe (TH), Karlsruhe; 2002
- [108] Sedlacek, G.: *Glas im konstruktiven Ingenieurbau*; Ernst und Sohn, Berlin; 1999
- [109] Song, H.-C.: Kopplung von CAD und Statik bei der integrierten Projektbearbeitung im Stahlbau, Dissertation; Universität Stuttgart, Stuttgart; 1994
- [110] Sontheimer, R.: Ein Verfahren zur automatisierten Bestimmung von Schweißbrennerstellungen als Beitrag zum Roboterschweißen im Stahlbau, Dissertation; Universität Stuttgart, Stuttgart; 1997
- [111] Specification of a CAD*I Neutral File for CAD Geometry, Version 3.2; *Research Reports ESPRIT Project 322, CAD Interfaces*; Berlin; 1987
- [112] ST-2 Reinforced Concrete Structures and Foundation Structures; *International Alliance for Interoperability (IAI)*; Projektinformationen unter: http://cic.vtt.fi/niai/technical/IFC_3/R3_ST2.html
- [113] Stein, E.: Notwendige und erstrebenswerte weitere Entwicklungen in der Baumechanik und Bauinformatik; *Der Bauingenieur* 75 (2000); 2000; S. 568 – 572

- [114] Suchocki., M.: Successfully Adopting Collaborative Technology; Proceedings of the Second Worldwide ECCE Symposium – Information and Communication Technology (ICT) in the Practice of Building and Civil Engineering, Espoo, Finland; 2001; S. 259 – 264
- [115] Tolman, F., Stephens, J., Steinmann, R., van Rees, R., Böhms, M., Zarli, A.: BcXML, an XML Vocabulary for Building and Construction; Proceedings of the Second Worldwide ECCE Symposium – Information and Communication Technology (ICT) in the Practice of Building and Civil Engineering, Espoo, Finland; 2001; S. 47 – 54
- [116] VDA: VDA-Flächenschnittstelle (VDAFS), Version 2.0; Verband der Automobilindustrie e.V. (VDA), Frankfurt; 1987
- [117] Verordnung über einen energiesparenden Wärmeschutz bei Gebäuden (Wärmeschutzverordnung – WärmeschutzV) vom 16.August 1994; Auszug aus dem Bundesgesetzblatt vom 24. August 1994
- [118] Weichert, J., Osterrieder, P.: Produktmodell DtH für den Ingenieurholzbau, Zimmermannsbau und Fertigausbau; Schriftenreihe Statik und Dynamik 2/2000, Brandenburgische Technische Universität Cottbus, Cottbus; 2000; Zu beziehen unter: <http://www.statik.tu-cottbus.de/>
- [119] Weise, M.: Produktmodellierung in Bauwesen – Konzeption und Validierung eines objekt-orientierten Tragwerksmodells für den Bereich Hochbau auf der Basis des IFC-Projektmodells 2.0, Diplomarbeit; Technische Universität Dresden, Dresden; 1999
- [120] Wörner, J.-D., Reymendt, J., Gunnarson, S., Stürmer, R.: Objektorientierte Integration von Teilprozessen im Bauwesen mit Hilfe einer objektorientierten Datenbank für den Bereich der Bemessung und Konstruktion von Hochbauteilen im Massivbau; Deutsche Forschungsgemeinschaft (DFG), Forschungsbericht Objektorientierte Modellierung in Planung und Konstruktion; Wiley-VCH, Weinheim; 2000; S.321 – 345
- [121] World Wide Web Consortium: Extensible Markup Language (XML) 1.0; W3C Recommendation; 1998; <http://www.w3.org/TR/1998/REC-xml-19990210>
- [122] World Wide Web Consortium: XML Schema; W3C Candidate Recommendation; 2001; <http://www.w3.org/TR/xmlschema-0/>

11 Glossar

4D-Visualisierung:	→ Visualisierung mit einer zeitlichen Komponente
AAM:	Application Activity Model: Teil eines → APs nach ISO 10303
Abstrakte Klasse:	→ Klasse, die nicht in einer Austauschdatei instanziiert werden kann.
Abstrakter Testfall:	Teil des Verfahrens zur Konformitätsprüfung nach ISO 10303
aecXML:	„Vokabular“ für den internetbasierten Handel im Bauwesen mit → XML-Technologie
Agent:	Software-Einheit, die im Auftrag des Nutzers Anfragen im Netz durchführt und die Antworten bereitstellt.
AIM:	Application Interpreted Model: Teil eines → APs nach ISO 10303
Anforderung:	Hier: Eine Anforderung wird in der Übersichtsplanung an ein → Bauelement im Hinblick auf die spätere Fachplanung gestellt, z.B. eine erforderliche Feuerwiderstandsklasse. Anforderungen werden als → dynamische Eigenschaften definiert.
Animations-Software:	Software zur räumlichen Darstellung von Bewegungsabläufen
Anwendungsprogramm:	Software zur Lösung bestimmter Aufgaben sowie zur Erstellung von Dokumenten und Plänen, z.B. CAD- oder Statikprogramme.
Anwendungsprozeß:	Darstellung einzelner Bereiche der Planung mit dem Ziel, die darin vorkommenden Informationsflüsse zu veranschaulichen.
Anwendungs-Software:	→ Anwendungsprogramm
Anwendungsszenario:	Darstellung zusammenhängender Anwendungsprozesse der Planung mit dem Ziel, die Anwendung eines Produktmodells zu verdeutlichen.
AP:	Application Protocol: → Produktmodell innerhalb von ISO 10303
AP 225	Application Protocol 225 – Building Elements Using Explicit Shape Representation: In ISO 10303-225 definiertes Produktmodell für das Bauwesen
Applikation, Applikations-Software:	Anwendungsprogramm
Applikationsdomänenmodell:	Bestimmte → Modellarchitektur eines → Daten- oder → Produktmodells
ARM:	Das Application Reference Model ist nach ISO 10303 Teil eines → APs

ASCII:	American Standard Code for Information Interchange
Attribut:	Bestandteil eines → Entities zur Beschreibung einer Eigenschaft
Auflösung:	Datenstruktur, bei der ein Bauelement in mehrere Komponenten aufgelöst wird. Diese Komponenten sind entweder ebenfalls Bauelemente oder → Bauteile. In den IFC wird diese Struktur durch die Klasse <i>IfcRelDecomposes</i> dargestellt.
Austauschdatei:	Einfachste Form des → Produktdatenaustausches, in der die Klassen eines Produktmodells in → ASCII-Dateien → instanziiert werden.
AVA:	Ausschreibung, Vergabe und Abrechnung
B2B:	Business to Business: → Internetbasierter Handel zwischen Unternehmen
B2C:	Business to Customer: → Internetbasierter Handel zwischen Unternehmen und Endkunden
Bauelement:	Hier: Teil eines Bauwerkes, das eine globale Funktion besitzt, zum Beispiel eine Wand. In den IFC werden Bauelemente als Unterklassen von <i>IfcBuildingElement</i> dargestellt.
Bauinformatik:	Fachgebiet innerhalb des Bauingenieurwesens, das sich mit der bauingenieurspezifischen Anwendungen der Computerwissenschaften befaßt.
Bauinformationssystem:	System zur Recherche in bestimmten Themengebieten des Bauwesens
Bauteil:	Hier: Teil eines Bauwerkes, das keine globale Funktion besitzt, zum Beispiel eine Verlegetafel. Mehrere Bauteile können ein Bauelement bilden. In dieser Arbeit werden Bauteile als Unterklassen von <i>IfcConstructionalElement</i> dargestellt.
Bauwerksstruktur:	Struktur aller gegenständlichen und nichtgegenständlichen Teile im → Objektmodell eines Gebäudes
bcXML:	„Vokabular“ für den internetbasierten Handel im Bauwesen mit → XML-Technologie
Bilaterale Schnittstelle:	Herkömmliches Schnittstellenformat, dessen Ziel lediglich der Datenaustausch zwischen zwei → Anwendungsprogrammen ist.
Blockmechanismus, -definition, -einfügung:	Datenstruktur in CAD-Programmen, bei der ein einziges geometrisches Objekt (Blockdefinition) von beliebig vielen anderen Objekten (Blockeinfügungen) referenziert wird. Die Blockdefinition definiert dabei die geometrische Form, die Blockeinfügungen jeweils eine Lage und Orientierung im Raum.

	Auf die IFC bezogen wird mit Blockmechanismus ein → Typmechanismus bezeichnet, der über eine vollständige geometrische Darstellung verfügt.
Boole'sche Operation:	Vereinigungs-, Durchschnitts- und Abzugsverknüpfungen zweier geometrischer Regelkörper, aus denen ein neuer Körper erzeugt wird.
Boole'scher Typ:	Datentyp, der den Wert <i>WAHR</i> oder <i>FALSCH</i> haben kann.
BREP:	Boundary Representation: → Explizite Geometriedarstellung, bei der Volumenkörper durch ihre Begrenzungsflächen definiert sind.
Browser:	Software zur Darstellung von Informationen aus dem Internet, zum Beispiel in Format von → html-Dokumenten.
C/C++:	Programmiersprache. C++ ist die objektorientierte Variante von C.
CAAD:	Computer Aided Architectural Design: → CAD-Software für Architektur
CAD:	Computer Aided Draughting oder Computer Aided Design: Sammelbegriff für Software für technisches Zeichnen oder Entwerfen
CAD-CAM-Kette:	Koppelung von Software für → CAD und → NC-Fertigung
CAFM:	Computer Aided Facility Management
CAM:	Computer Aided Manufacturing: Sammelbegriff für Software zur computerunterstützten Fertigung, zum Beispiel mit NC-Maschinen.
CIS:	CIMSteel Integration Standards: vom → SCI herausgegebener Standard für den Produktdatenaustausch im Stahlbau. Kern der CIS ist das Produktmodell → LPM.
CORBA:	Common Object Request Broker Architecture: Spezifikation einer Software-Architektur für verteilte Systeme
Core Layer:	→ Funktionale Schicht der IFC
CSG:	Construction Solid Geometry: → Explizite Geometriedarstellung, bei der Volumenkörper über → bool'sche Operationen anderer Volumenkörper definiert sind. Die Entstehungshierarchie (CSG-Baum) ist dabei Teil der Darstellung, so daß alle Operationen jederzeit rückgängig gemacht werden können.
Dachbelichtung:	Oberbegriff für Lichtkuppeln, -pyramiden und -bänder
Dach und Wand:	Gewerk, das in dieser Arbeit die Dach- und Außenwandbekleidung in Elementbauweise enthält, nicht jedoch tragende Wände in Stahlbetonbauweise.

Datenmodell:	Hier: Formale, modellhafte Definition eines Datenformates. Das Datenmodell ist das Kernstück eines → Produktmodells.
DBMS:	Datenbankmanagementsystem
Dekomposition:	→ Auflösung
DGfH	Deutsche Gesellschaft für Holzforschung
DFG:	Deutsche Forschungsgesellschaft
Dekomposition:	Auflösung: Datenstruktur, bei der ein → Bauelement in mehrere Komponenten aufgelöst wird. Diese Komponenten sind entweder ebenfalls Bauelemente oder → Bauteile.
DMS:	Dokumentenmanagementsystem, → IMS
Domäne:	Hier: Bereich eines → Produktmodells, der eine bestimmte Anwendung betrifft.
Domain Layer:	→ Funktionale Schicht der IFC
Domain-Rule:	Nach ISO 10303: Sprachelement von → EXPRESS, mit dem der Wertebereich von → Attributen eingeschränkt wird.
DStV:	Deutscher Stahlbau-Verband
DTD:	Dokumenttyp-Definition: Formatdefinition für → XML-Dokumente, geschrieben in → SGML.
dxfl:	Drawing Exchange Format: herstellerabhängiges Zeichnungsformat des CAD-Programms AutoCAD
Dynamische Eigenschaftsdefinition:	→ Semantische Eigenschaft, deren Bedeutung durch Vereinbarung <u>außerhalb</u> des eigentlichen Datenmodells definiert wird. In den IFC wird dies durch die sogenannten → PSETs erreicht.
Dynamische Erweiterung:	→ dynamische Eigenschaftsdefinition
DtH:	Produktmodell Datentransfer im Holzbau: von der → DGfH herausgegebenes Produktmodell für den Ingenieurholz-, Zimmermanns- und Fertighausbau
E-Business:	→ Internetbasierter Handel zwischen Unternehmen und Lieferanten
E-Commerce:	→ Internetbasierter Handel
EDMS:	Engineering Data Management System, → IMS
Eigenschaftsdefinition, Eigenschaftsklasse:	Klasse der IFC, die → semantische Eigenschaften für ein Objekt definiert.
Einfachvererbung:	Festlegung in den IFC, daß eine → Unterklasse nur eine einzige unmittelbare → Oberklasse besitzen darf.

Entity:	Nach ISO 10303 der wichtigste Datentyp in → EXPRESS. Ein Entity wird definiert zur Darstellung von Objekten oder deren Zuständen. In den IFC tritt der Begriff der → Klasse anstelle des Entities.
ESG:	Einscheiben-Sicherheitsglas
Explizite Geometrie:	Geometriedarstellung, bei der alle Bestandteile auf geometrischen Grundobjekten beruhen. Sie ist also unabhängig von Typisierungen.
EXPRESS:	Formale Spezifikationsprache nach ISO10303-11 für → Datenmodelle
EXPRESS-G:	Graphische Variante von → EXPRESS
EXPRESS-I:	Formale Sprache nach ISO10303-12 zur Darstellung von instanziierten → Datenmodellen
EXPRESS-Schema:	Satz von Regeln und Datendefinitionen in der Sprache → EXPRESS
EXPRESS-X:	Formale Sprache nach ISO10303-14 zur Definition des → Mappings von Daten aus einem → EXPRESS-Datenmodell in ein anderes
External Mapping:	Nach ISO 10303-21: Besondere Form der → Instanziierung eines → Datenmodells mit Vererbungsstrukturen
Extrusion:	Geometrische Operation, bei der eine Kurve oder eine Fläche entlang eines Pfades „gezogen“ wird, so daß eine Oberfläche (Oberflächenextrusion) oder ein Volumenkörper (Volumenkörperextrusion) entsteht.
FE, FEM	Finite-Elemente, Finite-Elemente-Methode
Feature-basierte Modellierung:	Datenmodellierung, die einem Volumenkörper vereinfachend Anarbeitungsschritte zuordnet, wodurch eine → CSG-Operation lediglich impliziert wird.
Fensteranordnung:	Anordnung mehrerer Einzelfenster in einer Wandöffnung
FM:	Facility Management: Bewirtschaftung von Gebäuden
Funktionale Schicht:	Das Datenmodell der → IFC ist in vier funktionale Schichten eingeteilt: Resource, Core, Interoperability und Domain Layer.
GAEB:	Gemeinsamer Ausschuß für Elektronik im Bauwesen
Gebäudetopologie:	→ Topologische Strukturen im → Objektmodell eines Gebäudes, durch die z.B. Fenster in Wänden verankert sind, oder mit denen sich Räume über ihre angrenzenden Wände und Decken definieren.
Generische Datenstruktur:	Struktur innerhalb eines Datenmodells, die selbst zu allgemein ist, um angewandt zu werden.
Geometrische Modellierung:	Datenmodellierung, die lediglich geometrische Daten beinhaltet.
Horizontale Kopplung:	Kopplung von gleichartiger Anwendungssoftware, zum Beispiel mehrerer → CAD-Programme

HTML (Hypertext Markup Language):	→ SGML-basiertes Dokumentenformat für Internetseiten
IAI:	Internationale Allianz für Interoperabilität: Nicht kommerzielle Organisation mit dem Ziel, den Datenaustausch im gesamten Bauwesen zu ermöglichen.
IFC:	Industry Foundation Classes: Produktmodell der → IAI
ifcXML:	→ XML-Version der → IFC
IDEF0:	Integration Definition for Function Modeling: Graphische Notation für Prozeßmodelle
Implementierung:	Hier: Einbindung eines → Produktmodells in → Applikationssoftware
Implementation Level:	Nach ISO 10303: Art der Implementierung. Vorgesehen sind vier Stufen der Kooperation: über Austauschdateien, über → Repositories, über → DBMS und über → wissensbasierte Systeme.
Implementierungsbereich:	→ Sicht
IMS:	Informationsmanagementsystem: Der Begriff wird in dieser Arbeit als Sammelbegriff für Systeme zur Ablage, Verwaltung und Verteilung digitaler Informationen gebraucht.
Indexparallele Listen:	Listen mit gleicher Länge, bei denen sich die Elemente an jeweils derselben Stelle aufeinander beziehen.
Integrale Planung:	Vision der ganzheitlichen Planung von Bauwerken, die deren gesamten Lebenszyklus, alle am Bau Beteiligten sowie die Nutzer berücksichtigt.
Integration:	→ Produktdatenaustausch
Interoperabilität:	→ Produktdatenaustausch
Instanz:	Nach ISO 10303: Ein mit konkreten Werten belegtes → Entity in einer Austauschdatei, einem → Repository oder einer Datenbank.
Instanziierung:	Vorgang des Belegens eines → Datenmodells mit konkreten → Instanzen.
Interface:	→ Use-Interface
Internal Mapping:	Nach ISO 10303-21: Besondere Form der → Instanziierung eines → Datenmodells mit Vererbungsstrukturen
Interoperability Layer:	→ Funktionale Schicht der IFC
Inverses Attribut:	Nach ISO 10303: Sprachelement von → EXPRESS, von dem im Gegensatz zum normalen Attribut kein Verweis ausgeht, sondern <u>auf</u> das verwiesen wird. Inverse Attribute tauchen in Austauschdateien nicht auf und dienen dazu, die Art und Anzahl der Instanzen zu beschränken, die <u>auf</u> eine Klasse verweisen. Außerdem können Verweise mit ihnen rückwärts verfolgt werden.

IuK:	Informations- und Kommunikationstechnologie: Sammelbegriff für alle modernen Technologien, die zur Verwaltung und Verteilung von Informationen dienen - von der Datenverwaltung in Unternehmen über Netzwerk- und Internet-Technologien bis zum Mobiltelefon
Java:	Programmiersprache zur Programmierung plattformunabhängiger Software. Sie ist damit ideal für das Internet geeignet ist.
Klasse:	In Programmiersprachen ist eine Klasse eine Einheit, die sowohl über Eigenschaften als auch über Methoden verfügt. In der Terminologie der IFC, die in dieser Arbeit übernommen wurde, tritt der Begriff der Klasse anstelle des → Entities aus ISO 10303.
Komplettbau:	Sammelbegriff für die Gewerke des Tragwerks, für Dach und Wand, für Fenster, Tore, Türen etc. sowie den Glasbau
Konkrete Datenstruktur:	→ Konkrete Klasse
Konkrete Klasse:	Im Gegensatz zur → abstrakten Klasse kann eine konkrete Klasse in einer Austauschdatei instanziiert werden.
Konkreter Testfall:	Nach ISO 10303: Teil des Verfahrens zur Konformitätsprüfung
Konformität:	Nach ISO 10303: Soll-Zustand einer → Implementierung eines → Produktmodells, bei dem alle syntaktischen, formalen und inhaltlichen Vorgaben des Produktmodellstandards erfüllt sind.
Konformitätsanforderung:	Nach ISO 10303: Detaillierte Anforderung, die in einem → Produktmodellstandard für die Implementierung einer bestimmten → Sicht aufgestellt ist.
Konformitätsklasse:	→ Sicht
Konformitätsprüfung:	Überprüfen des Konformitätszustandes der → Implementierung eines → Produktmodells in Applikationssoftware
Konzeptuelles Produktmodell:	→ ARM
Kooperation:	→ Produktdatenaustausch
Leitschichtung:	Hier: Für Bauelemente der Gebäudehülle global definierter Schichtaufbau in Form einer Instanz <i>IfcMaterialLayerUsage</i> .
LNC:	Leaf Node Class: → Konkrete Klasse, die einen Ast im Vererbungsbaum eines Datenmodells beendet.
LPM:	Logical Product Model: Produktmodell für den Stahlbau, das den Kern der CIMSteel Integration Standards (→ CIS) bildet.

Makroprofilierung:	Bezeichnet bei Trapezprofilen und Sandwichelementen die Ausbildung von Längsrippen, bei Kassettenelementen die Ausbildung von Gurten und Stegen.
Mapping:	Nach ISO 10303: Übertragen von Daten aus einem Format in ein anderes. Das Mapping von programminternen Daten in eine Austauschdatei wird auch als → Instanziierung bezeichnet.
Mechanisches Modell:	Hier: Computerdaten, die ein statisches System modellhaft beschreiben. Dies können zum Beispiel die Daten eines Stabwerksprogramms sein.
Metamodell:	Teil eines → Datenmodells, in dem lediglich grundlegende und generische Datenstrukturen definiert sind, die durch → Vererbung in anderen Modellteilen konkretisiert werden.
Metasprache:	Eine Metasprache beschreibt andere Sprachen. Der Vorteil von Metasprachen besteht darin, daß sie den Aufwand für die Entwicklung von Parsern (Übersetzern) verringern.
Mikroprofilierung:	Bezeichnet bei Trapezprofilen, Kassetten- und Sandwichelementen die Ausbildung von Sicken, Nuten oder Linierungen zusätzlich zur → Makroprofilierung.
Modellierungsart:	Hier: Die Modellierungsart bestimmt den Inhalt und die Darstellungsart eines → Datenmodells. Es wird unterschieden in → geometrische, → topologische, → semantische, → featurebasierte und → objektorientierte Modellierung.
Modellkern:	Innerster Teil eines → Datenmodells
NC:	Numerical Control: Numerische Steuerung von Fertigungsmaschinen
Oberklasse:	→ Klasse, die ihre Eigenschaften an eine oder mehrere → Unterklassen vererbt.
Objectified Relationship:	Regel in den IFC, gemäß der Objekte nicht direkt aufeinander verweisen, sondern über → Relationsklassen verknüpft werden.
Objekt:	In Programmiersprachen ist ein Objekt eine instanziierte → Klasse. In dieser Arbeit werden davon abweichend die Unterklassen von <i>IfcObject</i> bezeichnet.
Objektidentität:	Attribut einer Klasse zur eindeutigen Erkennung. In den IFC geschieht dies mit dem Typ <i>IfcGlobalUniqueId</i> .
Objektmodell:	Darstellung eines Bauwerkes in einem Produktmodell oder einer objektorientierten Software anhand von bauteilbezogenen Objekten anstatt durch Zeichnungsdaten

Objektorientierung:	Zum Schlagwort mutiertes Paradigma in der Informatik, das konkrete Anwendung in der → o.-o. Programmierung, als → o.-o. Software und in der → o.-o. Datenmodellierung findet. Hier wird es nur in der letzten Bedeutung verwendet.
Objektorientierte Datenmodellierung:	Modellierungsart, bei dem reale Sachverhalte anstelle von EDV-orientierten Strukturen modelliert werden. Die Objekte enthalten alle Daten, die das reale Vorbild besitzt. Charakteristisch ist die Verwendung von Objektkapselung und Vererbung.
Objektorientierte Programmierung:	Programmierart, bei der die Objekte neben beschreibenden Daten (analog zur → o.-o. Datenmodellierung) auch Funktionen besitzen, die sog. Methoden.
Objektorientierte Software:	Anwendungs-Software, die mit einem → Objektmodell arbeitet. Für den Anwender eine CAD-Programms beispielsweise äußert sich dies dadurch, daß er nicht mit Strichen arbeitet, sondern mit „intelligenten“ Objekten (Wände, Decken, Stahlträger etc.).
Palettierung:	Montagegerechtes Packen von Verlegebauteilen auf Paletten
Parametrische Geometriedarstellung:	Geometriedarstellung, die auf einer Typisierung mit festgelegten Parametern basiert. Ein Beispiel ist die Festlegung von Profilformen im Komplettbaumodell.
Planungskonflikt:	Planungskonflikte entstehen, wenn mehrere Planer inkompatible Datenbestände eines Gebäudes oder von Gebäudeteilen erzeugen.
PMS:	Projektmanagementsystem
Polygonzug:	Linienzug, der zwischen mehreren → Scheitelpunkten definiert ist.
Portal:	Einstiegsseite eines Internet-Dienstleisters
PPS:	Produktionsplanung und Steuerung: Einsatz rechnerunterstützter Systeme zur Planung, Steuerung und Überwachung von Produktionsabläufen
Produktdaten:	EDV-Daten, die ein Produkt als Ganzes beschreiben, und nicht nur Teilaspekte.
Produktdatenaustausch:	Kommunikation von Anwendungsprogrammen über ein → Produktmodell, im Idealfall ohne Datenverlust.
Produktmodell:	Modellhafte Darstellung eines Bauwerkes zum Zweck des → Produktdatenaustauschs. In dieser Arbeit ist hiermit immer der gesamte Standard gemeint, dessen Kernstück das → Datenmodell ist.
Produktschnittstelle Stahlbau:	Produktmodell des → DStV für die Konstruktion und statische Berechnung von Stahlbauten
Projektstruktur:	Aus Projektbeschreibung, Darstellungskontext und Festlegung von Maßeinheiten bestehender, äußerster Rahmen eine IFC-Austauschdatei.

Prozeßmodell:	Modell zur Beschreibung von Aktivitäten und Informationsflüssen in Planung oder Ausführung
PSD:	Property Set Definition markup language: → XML-basierte Sprache zur Definition → dynamischer Eigenschaften in den IFC
PSET:	→ Dynamische Eigenschaftsdefinition in den IFC durch <i>IfcPropertySet</i>
Redundanz:	Mehrfach vorhandene Informationen
Relation, Relationsklasse:	Spezielle Klasse der IFC, mit der Beziehungen zwischen → Objekten dargestellt werden.
Repository:	Software-Archiv zur Aufbewahrung von Quellen für Programme und Dokumenten oder strukturierten Produktdaten
Resource Layer:	→ Funktionale Schicht der IFC
Ressource:	Hier: Teil eines → Datenmodell, in dem ein grundlegendes und allgemeingültiges Darstellungskonzept definiert ist, z.B. für Maßeinheiten oder Geometrie.
RWA:	Rauch- und Wärmeabzugsanlage
Scoping Statement:	Teil eines → APs nach ISO 10303
SDAI:	Standard Data Acces Interface: In ISO 10303 standardisierte Programmierschnittstelle zur Implementierung von Produktmodellen
Semantik:	Lehre von der Wortbedeutung. In der Datenverarbeitung werden nichtgeometrische und nichttopologische Eigenschaften als semantisch bezeichnet. Die Bedeutung dieser Eigenschaften kann nicht abgeleitet, sondern muß vereinbart werden.
Server:	Zentraler Rechner in einem → Netzwerk, der den Arbeitsstationen Daten, Speicher und Ressourcen zur Verfügung stellt.
SGML:	Standard Generalized Markup Language: In ISO 8879 standardisierte → Metasprache zur Definition von Dokumentenformaten, z.B. → HTML.
Selektiver Import:	Beim selektiven Import von Daten sucht ein Programm sich bestimmte → Instanzen aus einem Datenbestand heraus, die es → interpretiert. Die übrigen Daten werden ignoriert und gehen verloren.
Sicht:	Hier: Eine Sicht definiert eine Untermenge eines → Datenmodells und stellt Regeln zur Behandlung des Datenbestandes Gruppen bestimmter → Anwendungsprogramme. Eine „CAD-Sicht“ würde z.B. die für CAD-Programme relevanten Teile eines Modell beinhalten.
Software-Applikation:	→ Anwendungs-Software
SSG:	Structural Sealant Glazing

Stabwerksprogramm:	Software zur Berechnung von Schnittgrößen und Verformungen von Stabwerken im Konstruktiven Ingenieurbau
Statische Eigenschaftsdefinition:	→ Semantische Eigenschaft, deren Bedeutung <u>innerhalb</u> der Klassendefinitionen eines Datenmodells vereinbart wird.
STEP:	Standard for the Exchange of Product Data: Kurzname von ISO 10303
STEP-Kompatibilität:	Ein Produktmodell ist kompatibel zu → ISO 10303 (STEP), wenn es gemäß der dort vorgeschriebenen Modellierungsmethoden definiert ist und die entsprechenden → Ressourcen unterstützt.
STEP-Toolkit:	Programmierwerkzeug speziell zur → Implementierung von → Produktmodellen nach ISO 10303
Tagged Document:	→ SGML-basiertes Dokument, benannt nach den typischen Formatierungszeichen.
TGA:	Technische Gebäudeausrüstung
Top-Down-Ansatz:	Prinzip der IFC, nach dem ein Bauwerk als Ganzes dargestellt wird und mit steigender Detaillierung in seine Bestandteile aufgelöst wird.
Topologie:	Lehre von der Lage und Anordnung geometrischer Gebilde im Raum
Transformationsoperator:	Festlegung der Transformationen (Translation, Rotation, Spiegelung, Skalierung) für geometrische Elemente innerhalb ihres Einfügekoordinatensystems
TVG:	Teilweise vorgespanntes Glas
Typmechanismus, -definition, -einfügung:	Datenstruktur in den IFC, bei der eine einzige Instanz (Typdefinition) von beliebig vielen anderen Instanzen, den (Typeinfügungen) referenziert wird. Die Typdefinition definiert dabei beliebige Eigenschaften, die Typeinfügungen jeweils eine Lage und Orientierung im Raum. Ein Typmechanismus mit einer vollständigen geometrischen Darstellung wird als → Blockmechanismus bezeichnet.
UML:	Unified Modeling Language: Visuelle Modellierungssprache zur Software-Entwicklung
Unterklasse:	→ Klasse, die einen Teil ihrer Eigenschaften aus einer oder mehreren → Oberklassen erbt.
Use-Interface:	Nach ISO 10303: Sprachelement von → EXPRESS, das es ermöglicht, in einem → Schema Klassen und Datentypen aus anderen Schemata zu „benutzen“.
Vererbung:	Paradigma in der Datenmodellierung, bei dem → Klassen ihre Eigenschaften an → Unterklassen weitergeben, also sozusagen „vererben“.
Vererbungsbaum:	Darstellung einer Klassenhierarchie in Baumform

Verglasungsfläche:	Verglaster Teil innerhalb eines Daches oder einer Wand
Verklotzung:	Bauteil zur Vermeidung eines direkten Kontaktes zwischen Glas und Verglasungsprofil im Glasbau
Verteiltes Arbeiten:	Arbeiten in → verteilten Systemen. In der → Bauinformatik ist das verteilte Arbeiten derzeit ein vielbesprochenes Thema, da es den Gegebenheiten des Bauwesens besonders gut Rechnung trägt.
Verteiltes System:	Computer-System, bei dem die Computer oder die Software-Komponenten in Netzwerken verteilt sind.
Vertikale Kopplung:	Kopplung von unterschiedlicher Anwendungssoftware, zum Beispiel von → CAD-Programmen und → NC-Steuerung
VG:	Verbundglas
Visualisierung:	Photorealistische Darstellung von räumlichen Objekten im Computer
Wissensbasiertes System:	Ein wissensbasiertes System führt Operationen mit Unterstützung von gespeicherten Fakten und sonstigem Wissen aus. Ein Beispiel aus dem Bauwesen ist eine Schadensdatenbank, aus dem der Benutzer Informationen anhand ähnlicher Fälle erhält.
Workflow:	Ablauforganisation von Vorgängen und Geschäftsprozessen
XML:	Extensible Markup Language: dokumentenorientierte → Metasprache zur Definition von Dokumentenformaten. XML ist eine kompakte Untermenge von → SGML.
XML-Schema:	XML-Sprache zur Definition von XML-Vokabularen, durch die → XML von der Dokumentenorientierung zur wirklichen Datenorientierung erweitert werden kann.
XSD:	XML-Schema-Definition: Tritt in → XML-Schema an die Stelle der → DTD von XML. Die XSD wird im Gegensatz zur DTD selbst in XML geschrieben.
Zeichenprogramm:	→ CAD-Programm, das im Gegensatz zu objektorientierten CAD-Programmen lediglich mit geometrischen Objekten arbeitet.

A Klassendefinitionen

A.1 Schemainformationen

Dieser Anhang enthält das vollständige EXPRESS-Modell für den Komplettbau als Erweiterung der IFC2x. Ausgenommen sind aus Platzgründen lediglich die Klassen für die Profilformen. Entgegen der Aussage in der Dokumentation ist das EXPRESS-Modell der IFC2x nicht wirklich in separate Schemata getrennt, da die schemaübergreifende Instanziierung in ISO 10303-21 derzeit nicht ausdrücklich behandelt wird. Bei einer Aufteilung des Modells auf mehrere Schemata wäre deshalb mit Schwierigkeiten bei der Anwendung von Programmierwerkzeugen zu rechnen. Die folgende Teile des Komplettbaumodells sind dementsprechend in einem einzigen Schema definiert, das lediglich die Abschnitte für die Domänen unterteilt ist. Diese Abschnitte sind formal ohne Bedeutung.

Komplettbaumodell:

EXPRESS-Schema: IFCKOMPLETT_2002_07_08
Datei: Komplettbau_2002_07_08.exp
Stand: 08.07.2002

Zugrundeliegendes IFC-Modell (hier nicht enthalten):

EXPRESS-Schema: IFC2X_PLATFORM
Datei: Ifc2x_platform.exp
Stand: 26.10.2000

A.2 *IFCMATERIALRESOURCEEXTENSION*

A.2.1 Klasse *IfcProductSpecification*

EXPRESS-Beschreibung:

```

ENTITY IfcProductSpecification
  SUBTYPE OF (IfcMaterial);
  Producer      : IfcLabel;
  Description   : OPTIONAL IfcText;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
Producer:	Hersteller
Description:	Zusätzliche Beschreibung in Textform

A.3 *IFCKERNELEXTENSION*

A.3.1 Klasse *IfcRelFixes*

EXPRESS-Beschreibung:

```

ENTITY IfcRelFixes
  ABSTRACT SUPERTYPE OF (ONEOF (
    IfcRelFixesSemantic,
    IfcRelFixesGeometric))
  SUBTYPE OF (IfcRelationship);
  ConnectionElement : IfcProduct;
  RelatingElements : LIST [1:?] OF IfcElement;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
ConnectionElement:	Befestigungselement
RelatingElements:	Zu befestigende Elemente

A.4 IFCPRODUCTEXTENSIONEXTENSION

A.4.1 Klasse *IfcRelFixesSemantic*

EXPRESS-Beschreibung:

```

ENTITY IfcRelFixesSemantic
  SUBTYPE OF (IfcRelFixes);
  PatternSpecification      : IfcText;
  Quantity                  : OPTIONAL IfcCountMeasure;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
PatternSpecification:	Beschreibung des Anordnungsmusters in Textform. Dies muß für den Ingenieur oder Zeichner verständlich sein.
Quantity:	Anzahl der benötigten Befestigungselemente

A.4.2 Klasse *IfcRelFixesGeometric*

EXPRESS-Beschreibung:

```

ENTITY IfcRelFixesGeometric
  SUBTYPE OF (IfcRelFixes);
  PatternSpecification      : IfcGeometricSet;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
PatternSpecification:	Die Elemente des <i>IfcGeometricSet</i> bestimmen die Anordnung der Befestigungen. Ein Schraubenmuster wird beispielsweise durch Instanzen von <i>IfcPoint</i> bestimmt.

A.4.3 Klasse *IfcConstructionalElement*

EXPRESS-Beschreibung:

```

ENTITY IfcConstructionalElement
  SUPERTYPE OF (ONEOF (
    IfcLayoutMember,
    IfcGlazingMember,
    IfcGlazingSupportMember))
  SUBTYPE OF (IfcElement);
END_ENTITY;

```

A.5 *IFCSHAREDBLDGELEMENTSEXTENSION*

A.5.1 Klasse *IfcWallIsDecomposed*

EXPRESS-Beschreibung:

```
ENTITY IfcWallIsDecomposed
  SUBTYPE OF (IfcWall);
END_ENTITY;
```

A.5.2 Klasse *IfcGlassFacade*

EXPRESS-Beschreibung:

```
ENTITY IfcGlassFacade
  SUBTYPE OF (IfcBuildingElement);
END_ENTITY;
```

A.5.3 Klasse *IfcGlassFace*

EXPRESS-Beschreibung:

```
ENTITY IfcGlassFace
  SUBTYPE OF (IfcBuildingElement);
END_ENTITY;
```

A.5.4 Klasse *IfcShallowFoundation*

EXPRESS-Beschreibung:

```
TYPE IfcShallowFoundationTypeEnum = ENUMERATION OF
  ( ISOLATED_FOUNDATION, STRIP_FOUNDATION,
    FOUNDATION_SLAB, USERDEFINED, NOTDEFINED);
END_TYPE;

ENTITY IfcShallowFoundation
  SUBTYPE OF (IfcBuildingElement);
  PredefinedType : IfcShallowFoundationTypeEnum;
END_ENTITY;
```


Attributbeschreibung:

Attribut	Beschreibung
PredefinedType:	Art der Gründung: <ul style="list-style-type: none"> - Einzelfundament - Streifenfundament - Fundamentplatte - Benutzerdefiniert - Keine Angabe

A.5.5 Klasse *IfcWindowArrangement***EXPRESS-Beschreibung:**

```

ENTITY IfcWindowArrangement
  SUBTYPE OF (IfcBuildingElement);
END_ENTITY;

```

A.5.6 Klasse *IfcDoorIndustrial***EXPRESS-Beschreibung:**

```

ENTITY IfcDoorIndustrial
  SUBTYPE OF (IfcBuildingElement);
END_ENTITY;

```

A.5.7 Klasse *IfcDoorIndustrialStyle***EXPRESS-Beschreibung:**

```

TYPE IfcDoorIndustrialStyleOperationTypeEnum =
  ENUMERATION OF (SINGLE_LEAF_HINGED, DOU-
    BLE_LEAF_HINGED, SINGLE_LEAF_SWINGING,
    DOUBLE_LEAF_SWINGING, DOUBLE_LEAF_FOLDING_HINGED,
    TRIPLE_LEAF_FOLDING_HINGED, FOLDING,
    SLIDING_FOLDING, SINGLE_LEAF_SLIDING,
    BI-PARTING_SLIDING, MULTI-PASSING_SLIDING,
    SLIDING_SWINGING, ROUND_THE_CORNER_SLIDING,
    COLLAPSIBLE_LATTICE,
    VERTICAL_SLIDING_UP, VERTICAL_SLIDING_DOWN,
    SECTIONAL_OVERHEAD, VERTICAL_ROLLING,
    HORIZONTALLY_ROLLING, UP_AND_OVER,
    VERTICAL_FOLDING, USERDEFINED, NOTDEFINED);
END_TYPE;

```

```
TYPE IfcDoorIndustrialStyleFrameConstructionTypeEnum =  
  ENUMERATION OF (ALUMINIUM, HIGH_GRADE_STEEL, STEEL,  
  USERDEFINED, NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorIndustrialStyleLeafConstructionTypeEnum =  
  ENUMERATION OF (ALUMINIUM, ALUMINIUM_GLASS,  
  PLASTIC, SANDWICH, STEEL, STEEL_GLASS, USERDEFINED,  
  NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorIndustrialStyleOperationMethodEnum =  
  ENUMERATION OF (MANUAL, POWERED, POTENTIAL_ENERGY,  
  GRAVITY, POTENTIAL_ENERGY_DEVICE, AUTOMATIC,  
  EMERGENCY, NOTDEFINED);  
END_TYPE;
```

```
ENTITY IfcDoorIndustrialStyle  
  SUBTYPE OF (IfcTypeProduct);  
  OperationType      : IfcDoorIndustrialStyle-  
                        OperationTypeEnum;  
  FrameConstructionType: IfcDoorIndustrialStyle-  
                        FrameConstructioTypeEnum;  
  LeafConstructionType : IfcDoorIndustrialStyleLeaf-  
                        ConstructionTypeEnum;  
  OperationMethod    : IfcDoorIndustrialStyleOpe-  
                        rationMethodEnum;  
  SurfaceArea        : OPTIONAL IfcAreaMeasure;  
END_ENTITY;
```

Attributbeschreibung:

Attribut	Beschreibung
OperationType:	Bewegungsart; Typisierung und Terminologie nach DIN 12433
FrameConstructionType:	Rahmenbauweise: <ul style="list-style-type: none"> - Aluminium - Hochfester Stahl - Stahl - Benutzerdefiniert - Keine Angabe
LeafConstructionType:	Torblatt- oder Torsegmentbauweise: <ul style="list-style-type: none"> - Aluminium - Aluminium und Glas - Kunststoff - Sandwich - Stahl - Stahl und Glas - Benutzerdefiniert - Keine Angabe
OperationMethod:	Betätigungsart; Typisierung und Terminologie nach DIN 12433
SurfaceArea:	Torfläche

A.5.8 Klasse *IfcDoorIndustrialFrameProperties***EXPRESS-Beschreibung:**

```

ENTITY IfcDoorIndustrialFrameProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  FrameOffsetX      : IfcLengthMeasure;
  FrameOffsetY      : IfcLengthMeasure;
  OverallWidth      : IfcPositiveLengthMeasure;
  OverallHeight     : IfcPositiveLengthMeasure;
  ThresholdThickness : OPTIONAL IfcPositive-
                        LengthMeasure;
  ThresholdDepth    : OPTIONAL IfcPositive-
                        LengthMeasure;
  ThresholdOffsetY  : OPTIONAL IfcLengthMeasure;
  JambThickness     : OPTIONAL IfcPositive-
                        LengthMeasure;
  JambDepth         : OPTIONAL IfcPositive-
                        LengthMeasure;
  JambOffsetY       : OPTIONAL IfcLengthMeasure;
  HeadThickness     : OPTIONAL IfcPositive-
                        LengthMeasure;
  HeadDepth         : OPTIONAL IfcPositive-
                        LengthMeasure;
  HeadOffsetY       : OPTIONAL IfcLengthMeasure;
END_ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A1.

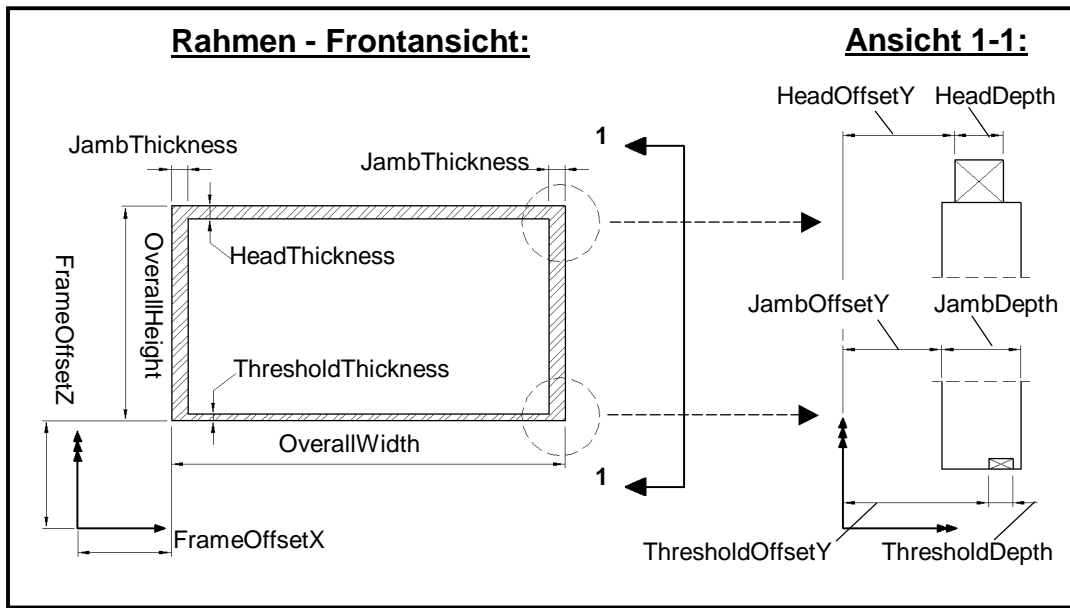


Abbildung A1: Geometrische Parameter des Torrahmens

A.5.9 Klasse *IfcDoorIndustrialLeafProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcDoorIndustrialLeafProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OffsetX      : IfcLengthMeasure;
  OffsetY      : IfcLengthMeasure;
  OffsetZ      : IfcLengthMeasure;
  OverallWidth : IfcPositiveLengthMeasure;
  OverallHeight : IfcPositiveLengthMeasure;
  PanelThickness : IfcPositiveLengthMeasure;
  SegmentNumber : OPTIONAL IfcInteger;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
SegmentNumber:	Anzahl der Torflügel (Falttor, Schiebefalttor) oder der Sektionen (Rundlauf-Schiebetor, Sektionaltor)
Alle weiteren Attribute gemäß Abbildung A2.	

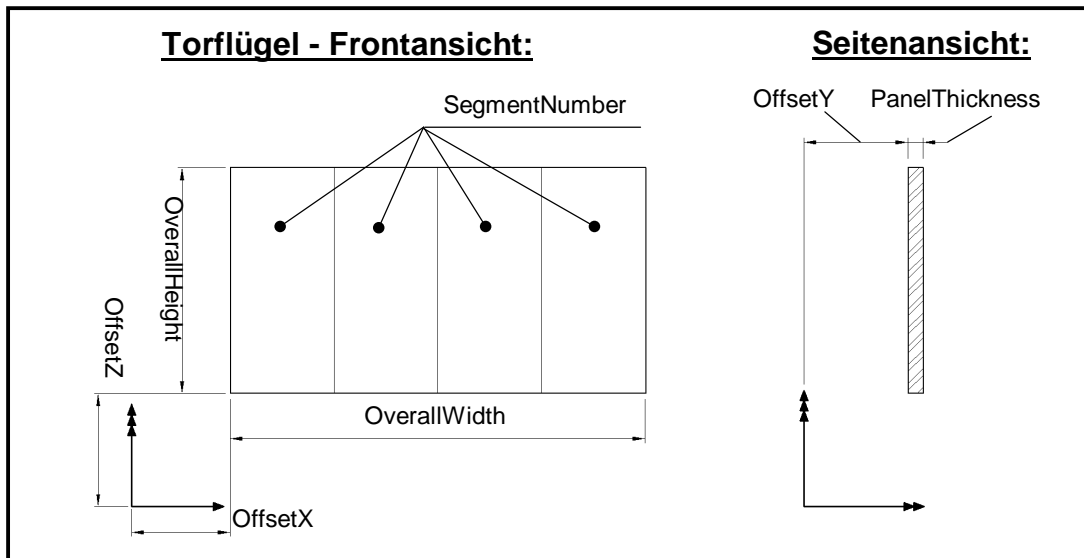


Abbildung A2: Geometrische Parameter des Torrahmens

A.5.10 Klasse *IfcRooflightElement*

EXPRESS-Beschreibung:

```
ENTITY IfcRooflightElement
  SUBTYPE OF (IfcBuildingElement);
END_ENTITY;
```

A.5.11 Klasse *IfcRooflightElementStyle*

EXPRESS-Beschreibung:

```
TYPE IfcRooflightElementTypeEnum = ENUMERATION OF
  (
    DOME_LIGHT, PYRAMID_LIGHT,
    CONTINUOUS_ROOFLIGHT_BARREL_VAULT,
    CONTINUOUS_ROOFLIGHT_GABLE_ROOF, FREEFORM);
END_TYPE;
```

```
TYPE IfcRooflightElementCoverConstructionTypeEnum =
  ENUMERATION OF (ACRYLIC_GLASS, ALUMINIUM_GLASS,
    POLYCARBONATE, STEEL_GLASS, USERDEFINED,
    NOTDEFINED);
END_TYPE;
```

```
TYPE IfcRooflightElementFrameConstructionTypeEnum =  
  ENUMERATION OF (ALUMINIUM,  
    GLASS_FIBRE_REINFORCED_PLASTIC, PVC,  
    STEEL,USERDEFINED, NOTDEFINED, NONE);  
END_TYPE;
```



```
TYPE IfcRooflightElementOpeningModeEnum = ENUMERATION  
  OF(FIXED, OPENING, NOTDEFINED);  
END_TYPE;
```



```
TYPE IfcRooflightElementOperationMethodEnum =  
  ENUMERATION OF (MANUAL, POWERED, NOTDEFINED);  
END_TYPE;
```



```
ENTITY IfcRooflightElementStyle  
  SUBTYPE OF(IfcTypeProduct);  
  ElementType : IfcRooflightElement-  
    TypeEnum;  
  CoverConstructionType : IfcRooflightElementCover-  
    ConstructionTypeEnum;  
  FrameConstructionType : IfcRooflightElementFrame-  
    ConstructionTypeEnum;  
  OpeningMode : IfcRooflightElement-  
    OpeningModeEnum;  
  OperationMethod : IfcRooflightElementOpera-  
    tionMethodEnum;  
  OpeningArea : OPTIONAL IfcAreaMeasure;  
END_ENTITY;
```

Attributbeschreibung:

Attribut	Beschreibung
ElementType:	Grundform: <ul style="list-style-type: none"> - Lichtkuppel - Lichtpyramide - Tonnengewölblichtband - Satteldachlichtband - Freie Form
CoverConstructionType:	Bauweise des Lichtelementes: <ul style="list-style-type: none"> - Acrylglas - Aluminium und Glas - PVC - Stahl und Glas - Benutzerdefiniert - Keine Angabe
FrameConstructionType:	Bauweise von Zarge oder Aufsetzkranz: <ul style="list-style-type: none"> - Aluminium - Aluminium und Glas - Kunststoff - Sandwich - Stahl - Stahl und Glas - Benutzerdefiniert - Keine Angabe
OpeningMode:	Öffnungsfunktion: <ul style="list-style-type: none"> - Starres Element - Zu öffnendes Element - Keine Angabe
OperationMode:	Betätigungsart: <ul style="list-style-type: none"> - Handbetätigung - Kraftbetätigung - Keine Angabe
OpeningArea:	Fläche der lichten Öffnung

A.5.12 Klasse *IfcSmokeAndHeatControlUnit***EXPRESS-Beschreibung:**

```

ENTITY IfcSmokeAndHeatControlUnit
  SUBTYPE OF (IfcBuildingElement);
END_ENTITY;

```

A.5.13 Klasse *IfcSmokeAndHeatControlUnitStyle*

EXPRESS-Beschreibung:

```

TYPE IfcSmokeAndHeatControlUnitTypeEnum =
  ENUMERATION OF (SMOKE_VENTILATOR, HEAT_VENTILATOR,
    SMOKE_AND_HEAT_VENTILATOR, USERDEFINED,
    NOTDEFINED);
END_TYPE;

```

```

TYPE IfcSmokeAndHeatControlUnitDriveEnum = ENUMERATION
  OF (MANUAL, ELECTRICAL, PNEUMATICAL, USERDEFINED,
    NOTDEFINED);
END_TYPE;

```

```

ENTITY IfcSmokeAndHeatControlUnitStyle
  SUBTYPE OF (IfcTypeProduct);
  UnitType      : IfcSmokeAndHeatControl-
                  UnitTypeEnum;
  DriveType     : IfcSmokeAndHeatControl-
                  UnitDriveEnum;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
UnitType:	Art des Elementes: <ul style="list-style-type: none"> - Rauchabzug - Wärmeabzug - Rauch- und Wärmeabzug - Benutzerdefiniert - Keine Angabe
DriveType:	Betätigungsart: <ul style="list-style-type: none"> - Manuell - Elektrisch - Pneumatisch - Benutzerdefiniert - Keine Angabe

A.6 *IFCOUTERSHELLDOMAIN*

A.6.1 Klasse *IfcSlabIsDecomposed*

EXPRESS-Beschreibung:

```
ENTITY IfcSlabIsDecomposed
  SUBTYPE OF (IfcSlab);
END_ENTITY;
```

A.6.2 Klasse *IfcLayoutMember*

EXPRESS-Beschreibung:

```
ENTITY IfcLayoutMember
  SUBTYPE OF (IfcConstructionElement);
END_ENTITY;
```

A.6.3 Klasse *IfcLayoutMemberStyle*

EXPRESS-Beschreibung:

```
TYPE IfcLayoutMemberStyleEnum = ENUMERATION OF
  (TRAPEZOIDAL_SHEET, TRAPEZIUM_LIKE_SHEET, C-PANEL,
  SANDWICH_ELEMENT, AERATED_CONCRETE_ELEMENT, FA-
  CADE_PANEL, SEALING_SHEET, THERMAL_INSULATION_BOARD,
  THERMAL_INSULATION_MAT, BOARD, USERDEFINED, NOTDE-
  FINED);
END_TYPE;
```

```
ENTITY IfcLayoutMemberStyle
  SUBTYPE OF (IfcTypeProduct);
  MemberType      :      IfcLayoutMemberStyleEnum;
END_ENTITY;
```

Attributbeschreibung:

Attribut	Beschreibung
MemberType:	Art des Bauteils: <ul style="list-style-type: none"> - Trapezprofiltafel - Tafel mit trapezförmlichem Profil - Kassettenpaneel - Sandwichelement - Gasbetonelement - Fassadentafel - Dichtungsbahn - Dämmungsbahn - Dämmungsmatte - Brett, Tafel oder Platte - Benutzerdefiniert - Keine Angabe

A.6.4 Klasse *IfcBuildingElementLayer***EXPRESS-Beschreibung:**

```

TYPE IfcBuildingElementLayerConstructionTypeEnum =
  ENUMERATION OF (TRAPEZOIDAL_SHEET,
    TRAPEZIUM_LIKE_SHEET, C-PANEL, SANDWICH,
    AERATED_CONCRETE, FACADE_PANEL, GENERAL_PROFILE,
    USERDEFINED, NOTDEFINED);
END_TYPE;

TYPE IfcBuildingElementLayerFunctionTypeEnum =
  ENUMERATION OF (LOAD_BEARING, ACOUSTIC_INSULATION,
    SEALING, SHEATHING, THERMAL_INSULATION,
    VAPOUR_BARRIER, DISTANCE, USERDEFINED, NOTDEFINED);
END_TYPE;

ENTITY IfcBuildingElementLayer
  SUBTYPE OF (IfcGroup);
  Tag : OPTIONAL IfcIdentifier;
  Description : OPTIONAL IfcText;
  LayerNumber : IfcInteger;
  LayerConstructionType : IfcBuildingElementLayer-
    ConstructionTypeEnum;
  LayerFunctionType : IfcBuildingElementLayer-
    FunctionTypeEnum;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
Tag:	Positionsnummer
Description:	Beschreibung der Schicht in Textform
LayerNumber:	Schichtnummer bezogen auf die Leitschichtung des zugeordneten Bauelementes
LayerConstructionType*:	Bauart mit: <ul style="list-style-type: none"> - Trapezprofiltafeln - Tafeln mit trapezförmlichem Profil - Kassettenpaneelen - Sandwichelementen - Gasbetonelementen - Fassadentafeln - Allgemeinen Profilen, z.B. für Abstandhalter - Benutzerdefiniert - Keine Angabe
LayerFunctionType:	Funktion innerhalb des Bauelementes: <ul style="list-style-type: none"> - Tragende Schicht - Schallabsorbierende Schicht - Dichtende Schicht - Verkleidung, Verblendung - Wärmedämmschicht - Dampfsperrschicht - Schicht aus Abstandhaltern - Benutzerdefiniert - Keine Angabe

*) Weitere Typen sind unnötig, da lediglich explizit dargestellte Bauteile (*IfcLayoutMember*, *IfcLayoutMemberStyle*) zu Schichten gruppiert werden können.

A.6.5 Klasse *IfcFasteningElementProxy***EXPRESS-Beschreibung:**

```

TYPE IfcFasteningElementTypeEnum = ENUMERATION OF
  (THREAD_ROLLING_SCREW, THREAD_CUTTING_SCREW,
  PARKER_SCREW, SELF_DRILLING_SCREW, WOOD_SCREW,
  STAND_OFF_SCREW, BLIND_RIVET, CARTRIDGE_FIRED_PIN,
  PLUG, SPOT_WELD, CLINCHED_SEAM, SEAM_WELD, GLUE,
  USERDEFINED, NOTDEFINED);
END_TYPE;

```

```

ENTITY IfcFasteningElementProxy
  SUBTYPE OF (IfcProxy);
    FasteningElementType      : IfcFasteningElement-
                               TypeEnum;
    Specification              : OPTIONAL IfcText;
    PositionNumber             : OPTIONAL IfcIdentifier;
  END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
FasteningElementType* :	Art des Befestigungsmittels: <ul style="list-style-type: none"> - gewindefurchende Schraube - gewindeschneidende Schraube - Blechtreibschraube - Bohrschraube - Holzschraube - Sandwichschraube mit 2 Gewinden - Blindniet - Setzbolzen - Dübel - Punktschweißung - Clinchverbindung - Rollnaht - Klebung - Benutzerdefiniert - Keine Angabe
Specification:	Beschreibung der Befestigungsmittels in Textform
PositionNumber:	Positionsnummer

*) Die Definition von Zusatzteilen, wie zum Beispiel Dichtungen oder Unterlegscheiben, werden in den standardisierten Bezeichnungen im Attribut Tag der Oberklasse *IfcTypeProduct* gemäß **Anhang C** zu dieser Arbeit angegeben.

A.7 IFCOPENINGCONSTRUCTIONDOMAIN

A.7.1 Klasse *IfcDoorStyleWithProducts*

EXPRESS-Beschreibung:

```

ENTITY IfcDoorStyleWithProducts
  SUBTYPE OF (IfcDoorStyle) ;
    LiningProducts          : OPTIONAL LIST [1:?] OF
                           IfcProductSpecification;
    LiningProductNumbers   : OPTIONAL LIST [1:?] OF
                           IfcCountMeasure;
    PanelProducts          : OPTIONAL LIST [1:?] OF
                           IfcProductSpecification;
    PanelProductNumbers    : OPTIONAL LIST [1:?] OF
                           IfcCountMeasure;
    EquipmentProducts      : OPTIONAL LIST [1:?] OF
                           IfcProductSpecification;
    EquipmentProductNumbers : OPTIONAL LIST [1:?] OF
                           IfcCountMeasure;
  END_ENTITY ;

```

Attributbeschreibung:

Attribut	Beschreibung
LiningProducts:	Liste mit Referenzen der unterschiedlichen Bauprodukte für den Türrahmen
LiningProduct-Numbers:	Liste, die zur Liste im Attribut <i>LiningProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben.
Die übrigen Attribute haben die analoge Bedeutung für das oder die Türblätter (Panel) und für alles sonstige Zubehör (Equipment). Produkte, die eindeutig am Türblatt befestigt sind (Schließzylinder, Türgriff etc.), sind in der Liste <i>PanelProducts</i> zu führen. Die Zahlen der Liste <i>PanelProductNumbers</i> gelten für <u>alle</u> Türblätter zusammen.	

A.7.2 Klasse *IfcWindowStyleWithProducts*

EXPRESS-Beschreibung:

```

ENTITY IfcWindowStyleWithProducts
  SUBTYPE OF (IfcWindowStyle);
  LiningProducts          : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  LiningProductNumbers   : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  PanelProducts          : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  PanelProductNumbers   : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  EquipmentProducts     : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  EquipmentProductNumbers : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;

END ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
LiningProducts:	Liste mit Referenzen der unterschiedlichen Bauprodukte für den Fensterrahmen
LiningProduct-Numbers:	Liste, die zur Liste im Attribut <i>LiningProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben.
Die übrigen Attribute haben die analoge Bedeutung für das oder die Fensterflügel (Panel) und für alles sonstige Zubehör (Equipment). Produkte, die eindeutig am Fensterflügel befestigt sind (z.B. Fenstergriff), sind in der Liste <i>PanelProducts</i> zu führen. Jalousien oder ähnliches zählen dagegen als „Equipment“. Die Zahlen der Liste <i>PanelProductNumbers</i> gelten für <u>alle</u> Fensterflügel zusammen.	

A.7.3 Klasse *IfcDoorIndustrialStyleWithProducts*

EXPRESS-Beschreibung:

```

ENTITY IfcDoorIndustrialStyleWithProducts
  SUBTYPE OF (IfcDoorIndustrialStyle);
  FrameProducts          : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  FrameProductNumbers   : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  LeafProducts          : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  LeafProductNumbers    : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  DriveProducts         : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  DriveProductNumbers   : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  EquipmentProducts     : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  EquipmentProductNumbers : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;

END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
FrameProducts:	Liste mit Referenzen der unterschiedlichen Bauprodukte für den Torrahmen
FrameProduct-Numbers:	Liste, die zur Liste im Attribut <i>FrameProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben.
Die übrigen Attribute haben die analoge Bedeutung für die Torflügel bzw. -segmente (Leaf), für den Antrieb (Drive) und für alles sonstige Zubehör (Equipment). Produkte, die eindeutig an den Torflügeln befestigt sind (z.B. Griffe), sind in der Liste <i>PanelProducts</i> zu führen. Die Zahlen der Liste <i>PanelProductNumbers</i> gelten für <u>alle</u> Torflügel zusammen.	

Attributbeschreibung:

Attribut	Beschreibung
ShapeType:	Grundrißform: - Kreisförmig - Rechteckig
DomeProducts:	Liste mit Referenzen der unterschiedlichen Bauprodukte für die eigentliche Kuppel
DomeProduct-Numbers:	Liste, die zur Liste im Attribut <i>DomeProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben.
Die übrigen Attribute haben die analoge Bedeutung für den Aufsetzkranz (Upstand), für den Antrieb (Drive) und für alles sonstige Zubehör (Equipment).	

A.7.6 Klasse *IfcDomeProperties***EXPRESS-Beschreibung:**

```

ENTITY IfcDomeProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OffsetX           : IfcLengthMeasure;
  OffsetY           : IfcLengthMeasure;
  OffsetZ           : IfcLengthMeasure;
  DomeThickness    : IfcPositiveLengthMeasure;
  DomeHeight       : IfcLengthMeasure;
  OuterDomeWidth   : IfcPositiveLengthMeasure;
  InnerDomeWidth   : IfcPositiveLengthMeasure;
  Slope1           : IfcPlaneAngleMeasure;
  OuterDomeLength  : OPTIONAL IfcPositiveLength-
                    Measure;
  InnerDomeLength  : OPTIONAL IfcPositiveLength-
                    Measure;
  Slope2           : OPTIONAL IfcPlaneAngleMeasure;
END_ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A3.

A.7.7 Klasse *IfcDomeLightUpstandProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcDomeLightUpstandProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  UpstandHeight      : IfcPositiveLengthMeasure;
  ClearWidth         : IfcPositiveLengthMeasure;
  InnerUpstandWidth  : IfcPositiveLengthMeasure;
  OuterUpstandWidth  : IfcPositiveLengthMeasure;
  ClearLength        : OPTIONAL IfcPositiveLengthMeasure;
  InnerUpstandLength : OPTIONAL IfcPositiveLengthMeasure;
  OuterUpstandLength : OPTIONAL IfcPositiveLengthMeasure;

END ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A3.

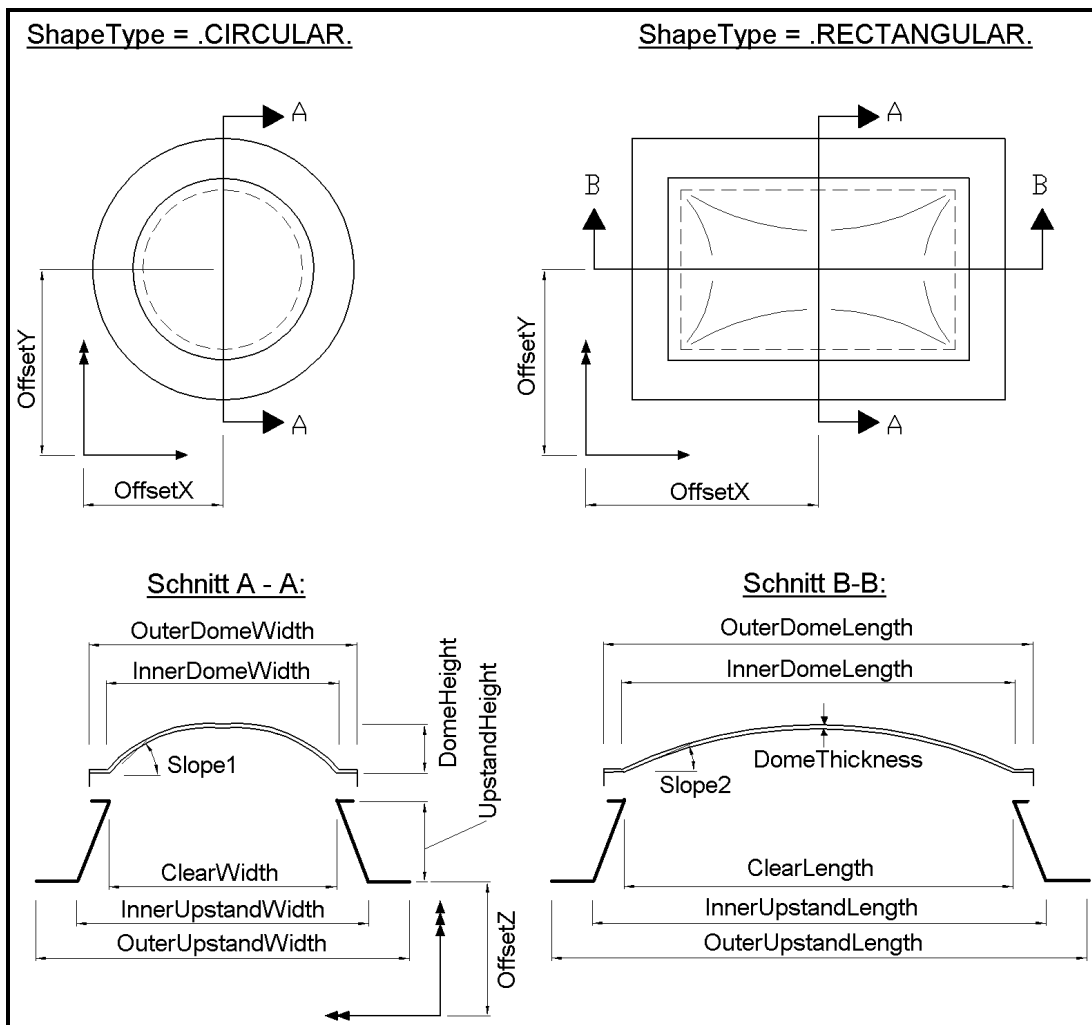


Abbildung A3: Geometrische Parameter von Lichtkuppeln

A.7.8 Klasse *IfcPyramidLightStyle*

EXPRESS-Beschreibung:

```

TYPE PyramidLightShapeEnum = ENUMERATION OF
  (EQUILATERAL_TRIANGULAR, ISOSCELES_TRIANGULAR,
  RECTANGULAR);
END_TYPE;

ENTITY IfcPyramidLightStyle
  SUBTYPE OF (IfcRooflightElementStyle);
  ShapeType                : IfcPyramidLightShapeEnum;
  PyramidProducts          : OPTIONAL LIST [1:?] OF
    IfcProductSpecification;
  PyramidProductNumbers   : OPTIONAL LIST [1:?] OF
    IfcCountMeasure;
  UpstandProducts         : OPTIONAL LIST [1:?] OF
    IfcProductSpecification;
  UpstandProductNumbers   : OPTIONAL LIST [1:?] OF
    IfcCountMeasure;
  DriveProducts           : OPTIONAL LIST [1:?] OF
    IfcProductSpecification;
  DriveProductNumbers     : OPTIONAL LIST [1:?] OF
    IfcCountMeasure;
  EquipmentProducts       : OPTIONAL LIST [1:?] OF
    IfcProductSpecification;
  EquipmentProductNumbers : OPTIONAL LIST [1:?] OF
    IfcCountMeasure;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
ShapeType:	Grundrißform: - Gleichseitiges Dreieck - Rechtwinkliges Dreieck - Rechteck
PyramidProducts:	Liste mit Referenzen der unterschiedlichen Bauprodukte für die eigentliche Pyramide
PyramidProductNumbers:	Liste, die zur Liste im Attribut <i>PyramidProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben.
Die übrigen Attribute haben die analoge Bedeutung für den Aufsetzkranz (Upstand), für den Antrieb (Drive) und für alles sonstige Zubehör (Equipment).	

A.7.9 Klasse *IfcPyramidProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcPyramidProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OffsetX      : IfcLengthMeasure;
  OffsetY      : IfcLengthMeasure;
  OffsetZ      : IfcLengthMeasure;
  PyramidThickness : IfcPositiveLengthMeasure;
  PyramidHeight : IfcLengthMeasure;
  OuterPyramidWidth : IfcPositiveLengthMeasure;
  InnerPyramidWidth : IfcPositiveLengthMeasure;
  OuterPyramidLength : OPTIONAL IfcPositiveLength-
    Measure;
  InnerPyramidLength : OPTIONAL IfcPositiveLength-
    Measure;

END_ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A4.

A.7.10 Klasse *IfcPyramidLightUpstandProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcPyramidLightUpstandProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  UpstandHeight : IfcPositiveLengthMeasure;
  ClearWidth    : IfcPositiveLengthMeasure;
  InnerUpstandWidth : IfcPositiveLengthMeasure;
  OuterUpstandWidth : IfcPositiveLengthMeasure;
  ClearLength   : OPTIONAL IfcPositiveLength-
    Measure;
  InnerUpstandLength : OPTIONAL IfcPositiveLength-
    Measure;
  OuterUpstandLength : OPTIONAL IfcPositiveLength-
    Measure;

END_ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A4.

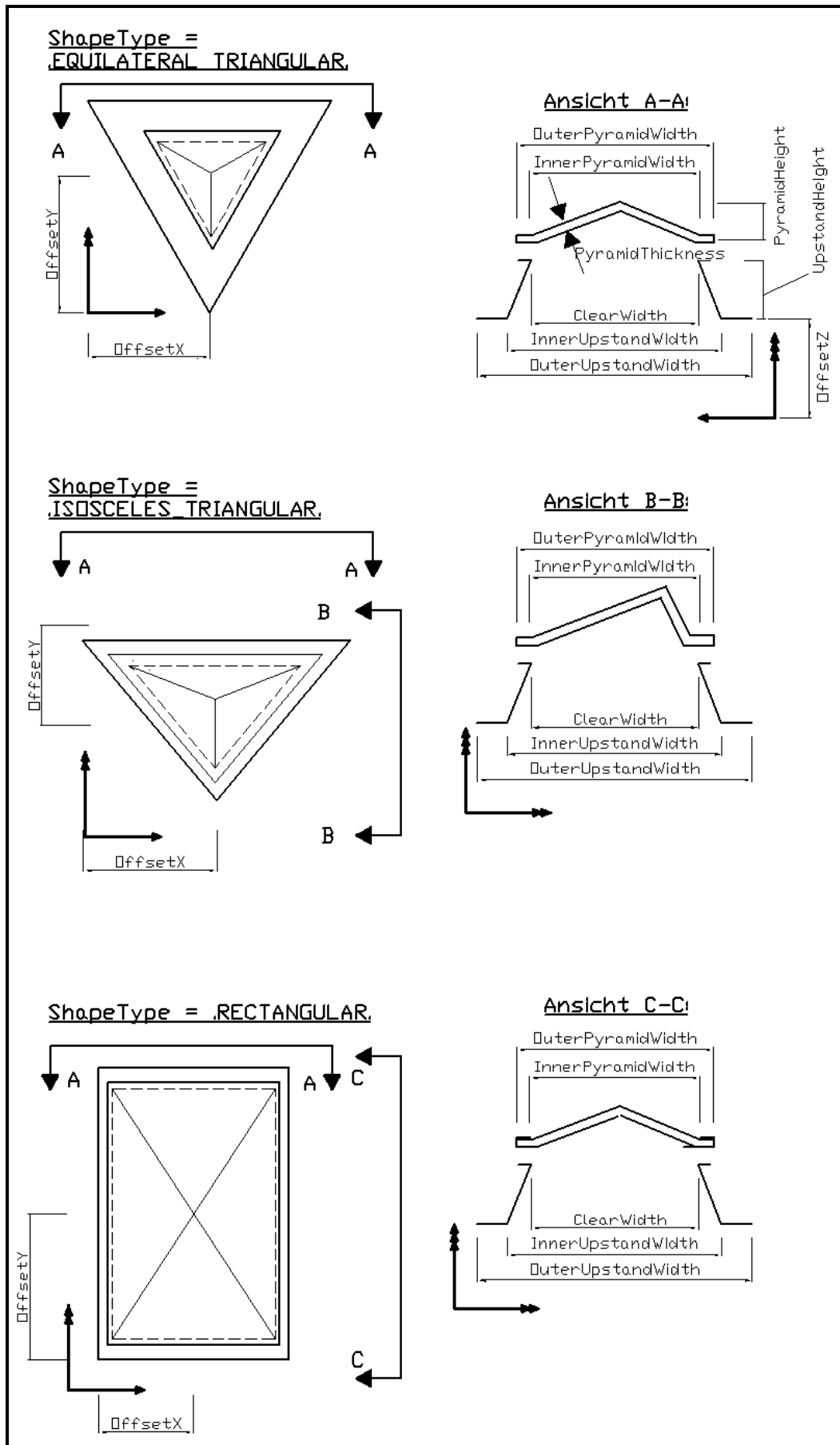


Abbildung A4: Geometrische Parameter von Lichtpyramiden

A.7.11 Klasse *IfcRooflightBarrelVaultStyle*

EXPRESS-Beschreibung:

```

ENTITY IfcRooflightBarrelVaultStyle
  SUBTYPE OF (IfcRooflightElementStyle);
  SegmentNumber           : IfcCountMeasure;
  TensionMemberNumber    : OPTIONAL IfcCountMeasure;
  SegmentProducts        : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  SegmentProductNumbers  : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  FrameProducts          : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  FrameProductNumbers    : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  TransomProducts        : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  TransomProductNumbers  : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  EquipmentProducts      : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  EquipmentProductNumbers : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;

END ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
Segment- Number:	Anzahl der Segmente in Längsrichtung (X-Richtung). Die Form eines gewölbten Lichtsegmentes ist durch die Klasse <i>IfcRooflightVaultSegmentProperties</i> gegeben. Die gewölbten Sprossen sind jeweils zwischen zwei Segmenten („Inner Transom“) und an den Rändern („Outer Transom“) angeordnet. Ihre Form wird durch <i>IfcRooflightVaultTransomProperties</i> gegeben. Die Form der <u>gesamten Zarge</u> wird in <i>IfcRooflightVaultFrameProperties</i> angegeben.
TensionMember- Number:	Anzahl der Zugbänder, die in Querrichtung zur Aufnahme des Bogenschubs angeordnet werden.
Segment- Products:	Liste mit Referenzen der unterschiedlichen Bauprodukte für die Lichtsegmente
SegmentProduct- Numbers:	Liste, die zur Liste im Attribut <i>SegmentProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben. Die Anzahl gilt für <u>alle Segmente</u> zusammen.
Die übrigen Attribute haben die analoge Bedeutung für den Rahmen (Frame), für die Sprossen (Transom) und für alles sonstige Zubehör (Equipment).	

A.7.12 Klasse *IfcRooflightVaultSegmentProperties*

EXPRESS-Beschreibung:

```
ENTITY IfcRooflightVaultSegmentProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  SegmentThickness      : IfcPositiveLengthMeasure;
  SegmentHeight         : IfcPositiveLengthMeasure;
  SegmentWidth          : IfcPositiveLengthMeasure;
  SegmentLength         : IfcPositiveLengthMeasure;
  SegmentSlope          : IfcPlaneAngleMeasure;
END_ENTITY;
```

Attributbeschreibung: Alle Attribute gemäß Abbildung A5.

A.7.13 Klasse *IfcRooflightVaultFrameProperties*

EXPRESS-Beschreibung:

```
ENTITY IfcRooflightVaultFrameProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OffsetX                : IfcLengthMeasure;
  OffsetY                : IfcLengthMeasure;
  OffsetZ                : IfcLengthMeasure;
  FrameHeight            : IfcPositiveLengthMeasure;
  ClearWidth             : IfcPositiveLengthMeasure;
  InnerFrameWidth        : IfcPositiveLengthMeasure;
  OuterFrameWidth        : IfcPositiveLengthMeasure;
  ClearLength            : IfcPositiveLengthMeasure;
  InnerFrameLength       : IfcPositiveLengthMeasure;
  OuterFrameLength       : IfcPositiveLengthMeasure;
END_ENTITY;
```

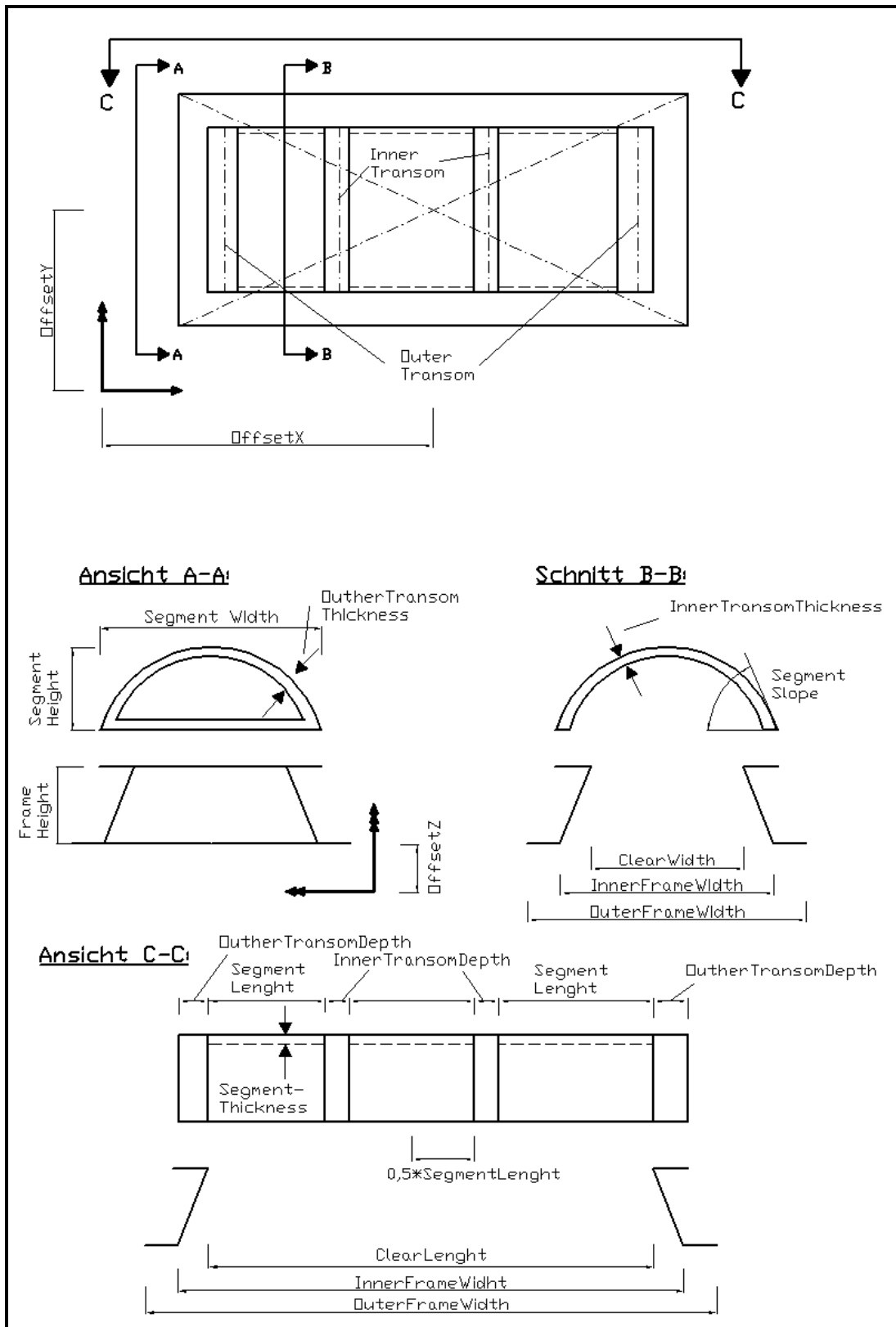
Attributbeschreibung: Alle Attribute gemäß Abbildung A5.

A.7.14 Klasse *IfcRooflightVaultTransomProperties*

EXPRESS-Beschreibung:

```
ENTITY IfcRooflightVaultTransomProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OuterTransomThickness : IfcPositiveLengthMeasure;
  OuterTransomDepth     : IfcPositiveLengthMeasure;
  InnerTransomThickness : OPTIONAL IfcPositive-
                          LengthMeasure;
  InnerTransomDepth     : OPTIONAL IfcPositive-
                          LengthMeasure;
END_ENTITY;
```

Attributbeschreibung: Alle Attribute gemäß Abbildung A5.



**Abbildung A5: Geometrische Parameter von Tonnengewölblichtbändern
(hier: *SegmentNumber* = 3)**

A.7.15 Klasse *IfcRooflightGableRoofStyle*

EXPRESS-Beschreibung:

```

ENTITY IfcRooflightGableRoofStyle
  SUBTYPE OF (IfcRooflightElementStyle);
  SegmentNumber           : IfcCountMeasure;
  TensionMemberNumber    : OPTIONAL IfcCountMeasure;
  SegmentProducts        : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  SegmentProductNumbers  : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  FrameProducts          : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  FrameProductNumbers    : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  TransomProducts       : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  TransomProductNumbers : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;
  EquipmentProducts      : OPTIONAL LIST [1:?] OF
                          IfcProductSpecification;
  EquipmentProductNumbers : OPTIONAL LIST [1:?] OF
                          IfcCountMeasure;

END ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
Segment- Number:	Anzahl der Segmente in Längsrichtung (X-Richtung). Die Form eines sattelförmigen Lichtsegmente ist durch die Klasse <i>IfcRooflightGableSegmentProperties</i> gegeben. Die Sprossen sind jeweils zwischen zwei Segmenten („Inner Transom“) und an den Rändern („Outer Transom“) angeordnet. Ihre Form wird durch <i>IfcRooflightGableTransomProperties</i> gegeben. Die Form der <u>gesamten</u> Zarge wird in <i>IfcRooflightGableFrameProperties</i> angegeben.
TensionMember- Number:	Anzahl der Zugbänder, die in Querrichtung zur Aufnahme des Bogenschubs angeordnet werden.
Segment- Products:	Liste mit Referenzen der unterschiedlichen Bauprodukte für die Lichtsegmente
SegmentProduct- Numbers:	Liste, die zur Liste im Attribut <i>SegmentProducts</i> indexparallel ist. Für jedes dort referenzierte Produkt wird in dieser Liste die benötigte Anzahl angegeben. Die Anzahl gilt <u>für alle Segmente</u> zusammen.
Die übrigen Attribute haben die analoge Bedeutung für den Rahmen (Frame), für die Sprossen (Transom) und für alles sonstige Zubehör (Equipment).	

A.7.16 Klasse *IfcRooflightGableSegmentProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcRooflightGableSegmentProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  SegmentThickness      : IfcPositiveLengthMeasure;
  SegmentHeight        : IfcPositiveLengthMeasure;
  SegmentWidth         : IfcPositiveLengthMeasure;
  SegmentLength        : IfcPositiveLengthMeasure;
END ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A6.

A.7.17 Klasse *IfcRooflightGableFrameProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcRooflightGableFrameProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OffsetX              : IfcLengthMeasure;
  OffsetY              : IfcLengthMeasure;
  OffsetZ              : IfcLengthMeasure;
  FrameHeight         : IfcPositiveLengthMeasure;
  ClearWidth          : IfcPositiveLengthMeasure;
  InnerFrameWidth     : IfcPositiveLengthMeasure;
  OuterFrameWidth     : IfcPositiveLengthMeasure;
  ClearLength         : IfcPositiveLengthMeasure;
  InnerFrameLength    : IfcPositiveLengthMeasure;
  OuterFrameLength    : IfcPositiveLengthMeasure;
END ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A6.

A.7.18 Klasse *IfcRooflightGableTransomProperties*

EXPRESS-Beschreibung:

```

ENTITY IfcRooflightGableTransomProperties
  SUBTYPE OF (IfcPropertySetDefinition);
  OuterTransomThickness : IfcPositiveLengthMeasure;
  OuterTransomDepth    : IfcPositiveLengthMeasure;
  InnerTransomThickness : OPTIONAL IfcPositive-
                          LengthMeasure;
  InnerTransomDepth    : OPTIONAL IfcPositive-
                          LengthMeasure;
END ENTITY;

```

Attributbeschreibung: Alle Attribute gemäß Abbildung A6.

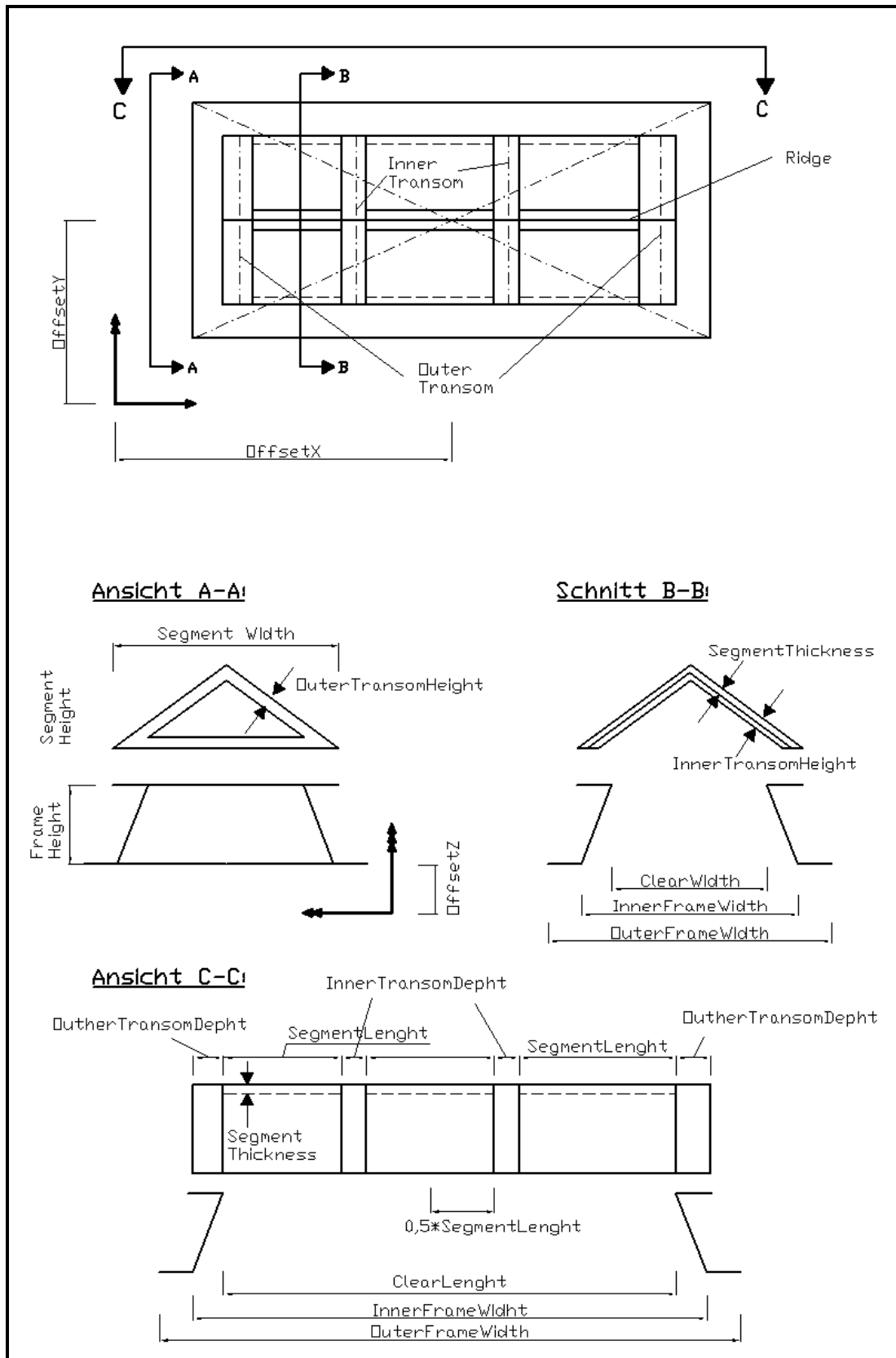


Abbildung A6: Geometrische Parameter von Satteldachlichtbändern
(hier: *SegmentNumber* = 3)

A.8 IFCGLAZINGCONSTRUCTIONDOMAIN

A.8.1 Klasse *IfcGlassFaceIsDecomposed*

EXPRESS-Beschreibung:

```

ENTITY IfcGlassFaceIsDecomposed
  SUBTYPE OF (IfcGlassFace);
END_ENTITY;

```

A.8.2 Klasse *IfcGlazingMember*

EXPRESS-Beschreibung:

```

ENTITY IfcGlazingMember
  SUBTYPE OF (IfcConstructionalElement);
END_ENTITY;

```

A.8.3 Klasse *IfcGlazingMemberStyle*

EXPRESS-Beschreibung:

```

TYPE IfcGlazingSupportTypeEnum = ENUMERATION OF
  (
    POINT_LIKE_SUPPORT, IDEAL_POINT_SUPPORT,
    GLAZING_PROFILE_SUPPORT,
    STRUCTURAL_SEALANT_GLAZING, USERDEFINED,
    NOTDEFINED);
END_TYPE;

TYPE IfcGlazingTypeEnum = ENUMERATION OF
  (
    SINGLE_PANE_FLOAT_GLASS,
    SINGLE_PANE_CAST_GLASS,
    SINGLE_PANE_TEMPERED_SAFETY_GLASS,
    SINGLE_PANE_SAFETY_GLASS_WITH_FILM,
    SINGLE_PANE_PARTLY_TEMPERED_GLASS,
    COMPOUND_GLASS, COMPOUND_SAFETY_GLASS,
    DOUBLE_PANE_INSULATING_GLASS,
    TRIPLE_PANE_INSULATING_GLASS, USERDEFINED,
    NOTDEFINED);
END_TYPE;

ENTITY IfcGlazingMemberStyle
  SUBTYPE OF (IfcTypeProduct);
  SupportType : IfcGlazingSupportTypeEnum;
  GlazingType : IfcGlazingTypeEnum;
END_ENTITY;

```

Attributbeschreibung:

Attribut	Beschreibung
SupportType:	Lagerungsart: <ul style="list-style-type: none"> - Punktförmige Halterung - Ideale Punkthalterung - Mechanische Linienlagerung mit Verglasungsprofilen - Structural Sealant Glazing - Benutzerdefiniert - Keine Angabe
GlazingType:	Glasart: <ul style="list-style-type: none"> - Ein-Scheiben-Floatglas - Ein-Scheiben-Gußglas - Thermisch vorgespanntes Ein-Scheiben-Sicherheitsglas - Ein-Scheiben-Sicherheitsglas mit Folie - Thermisch teilvorgespanntes Ein-Scheiben-Glas - Verbundglas - Verbundsicherheitsglas - Zwei-Scheiben-Isolierglas - Drei-Scheiben-Isolierglas - Benutzerdefiniert - Keine Angabe

A.8.4 Klasse *IfcGlazingSupportMember***EXPRESS-Beschreibung:**

```

ENTITY IfcGlazingSupportMember
  SUBTYPE OF (IfcConstructionalElement);
END_ENTITY;

```

A.8.5 Klasse *IfcGlazingSupportMemberStyleLinear***EXPRESS-Beschreibung:**

```

ENTITY IfcGlazingSupportMemberStyleLinear
  SUBTYPE OF (IfcTypeProduct);
END_ENTITY;

```

A.9 IFCPROFILEEXTENSIONRESOURCE

A.9.1 Klasse *IfcZetaShapeProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcZetaShapeProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position          : IfcAxis2Placement2D;
  CentreOfGravityInY : IfcPositiveLengthMeasure;
  THICKNESS         : IfcPositiveLengthMeasure;
  B1                : IfcPositiveLengthMeasure;
  B2                : IfcPositiveLengthMeasure;
  B3                : IfcPositiveLengthMeasure;
  B4                : IfcPositiveLengthMeasure;
  H1                : IfcPositiveLengthMeasure;
  H2                : IfcPositiveLengthMeasure;
  H3                : IfcPositiveLengthMeasure;
  H4                : IfcPositiveLengthMeasure;
  A1                : IfcPositiveLengthMeasure;
  A2                : IfcPositiveLengthMeasure;
  ALPHA_1           : IfcPlaneAngleMeasure;
  ALPHA_2           : IfcPlaneAngleMeasure;
  CornerRadius      : OPTIONAL IfcPositiveLengthMeasure;
END ENTITY;

```

Die Maße von Zeta-Profilen gemäß Abbildung A7 beziehen sich auf die Profilmittellinie. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in allen Eckbereichen. Ist dieser nicht angegeben, ist er zu Null anzunehmen. Das Attribut *Position* definiert das lokale Querschnittskoordinatensystem, das im angenäherten Schwerpunkt des Profils liegt (Attribut *CentreOfGravityInY*).

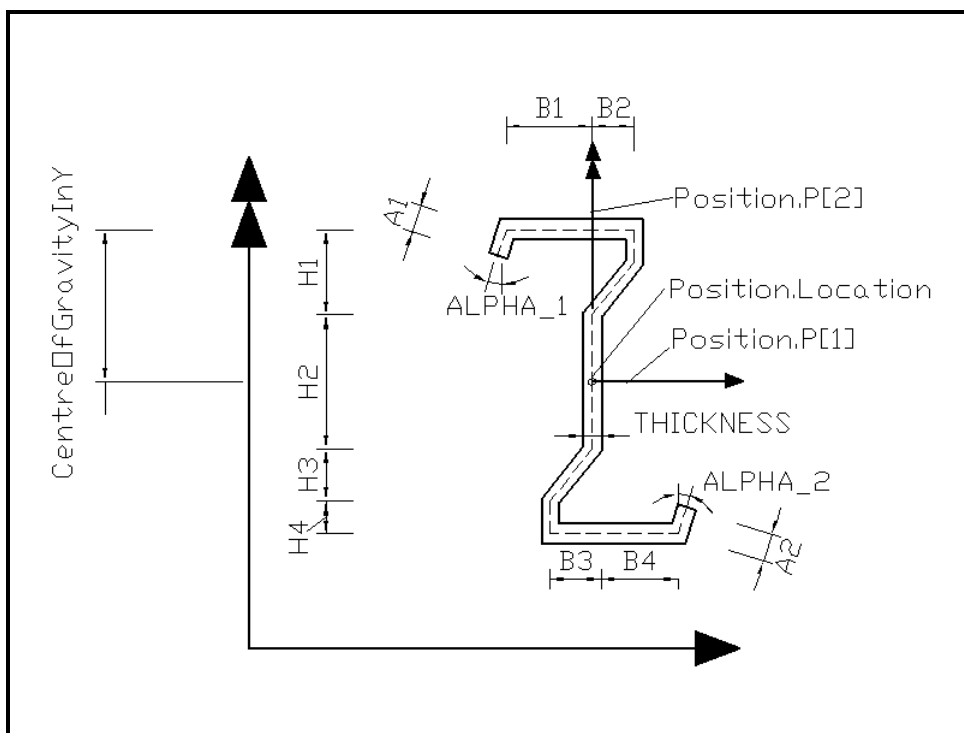


Abbildung A7: Geometrische Parameter von Zeta-Profilen

A.9.2 Klasse *IfcEavesBeamProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcEavesBeamProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position                : IfcAxis2Placement2D;
  CentreOfGravityInY     : IfcPositiveLengthMeasure;
  THICKNESS               : IfcPositiveLengthMeasure;
  B1                      : IfcPositiveLengthMeasure;
  B2                      : IfcPositiveLengthMeasure;
  B3                      : IfcPositiveLengthMeasure;
  H1                      : IfcPositiveLengthMeasure;
  H2                      : IfcPositiveLengthMeasure;
  H3                      : IfcPositiveLengthMeasure;
  H4                      : IfcPositiveLengthMeasure;
  H5                      : IfcPositiveLengthMeasure;
  A1                      : IfcPositiveLengthMeasure;
  A2                      : IfcPositiveLengthMeasure;
  A3                      : IfcPositiveLengthMeasure;
  ALPHA_1                 : IfcPlaneAngleMeasure;
  ALPHA_2                 : IfcPlaneAngleMeasure;
  ALPHA_3                 : IfcPlaneAngleMeasure;
  CornerRadius           : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;

```

Die Maße von Traufpfettenprofilen gemäß Abbildung A8 beziehen sich auf die Profilmittellinie. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in allen Eckbereichen. Ist dieser nicht angegeben, ist er zu Null anzunehmen. Das Attribut *Position* definiert das lokale Querschnittskoordinatensystem, das im angenäherten Schwerpunkt des Profils liegt (Attribut *CentreOfGravityInY*).

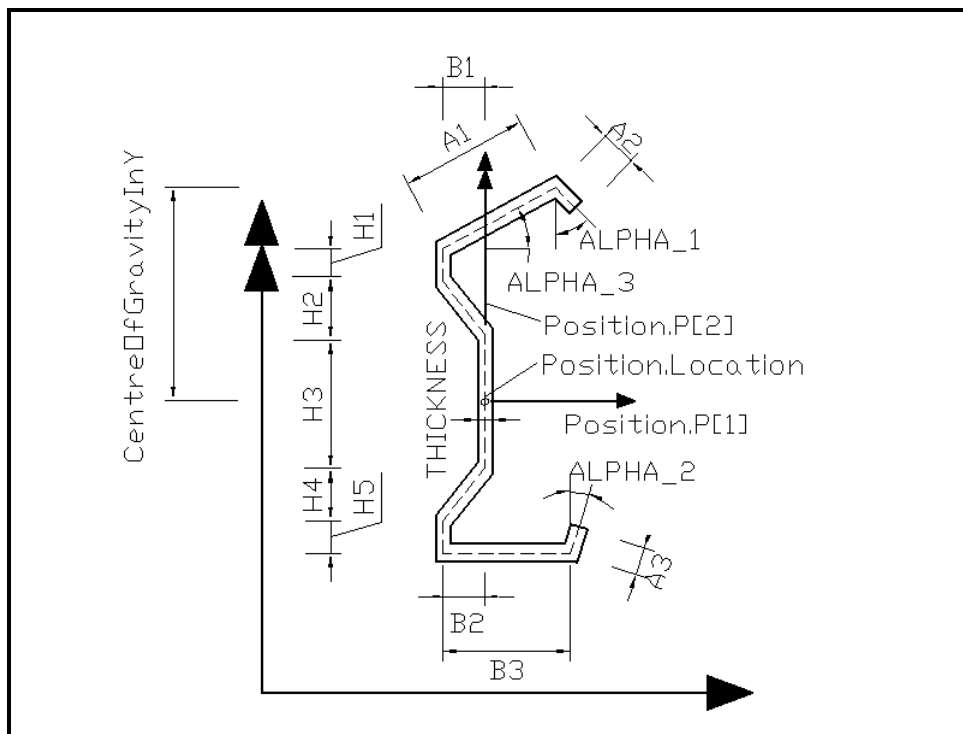


Abbildung A8: Geometrische Parameter von Traufpfettenprofilen

A.9.3 Klasse *IfcTrapezoidalSheetProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcTrapezoidalSheetProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position      : IfcAxis2Placement2D;
  THICKNESS     : IfcPositiveLengthMeasure;
  B1            : IfcPositiveLengthMeasure;
  B2            : IfcPositiveLengthMeasure;
  B3            : IfcPositiveLengthMeasure;
  H1            : IfcPositiveLengthMeasure;
  N            : IfcInteger;
  CornerRadius  : OPTIONAL IfcPositiveLengthMeasure;
  WHERE
    WR1          : N >= 1;
END_ENTITY;

```

Die Maße einer Profilrippe von Trapezprofiltafeln gemäß Abbildung A9 beziehen sich auf die Profilmittellinie. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in allen Eckbereichen. Ist dieser nicht angegeben, ist er zu Null anzunehmen. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilunterkante. Die Verlegebreite einer Profiltafel ergibt sich von dort aus zu N -mal der Rippenbreite $B3$.

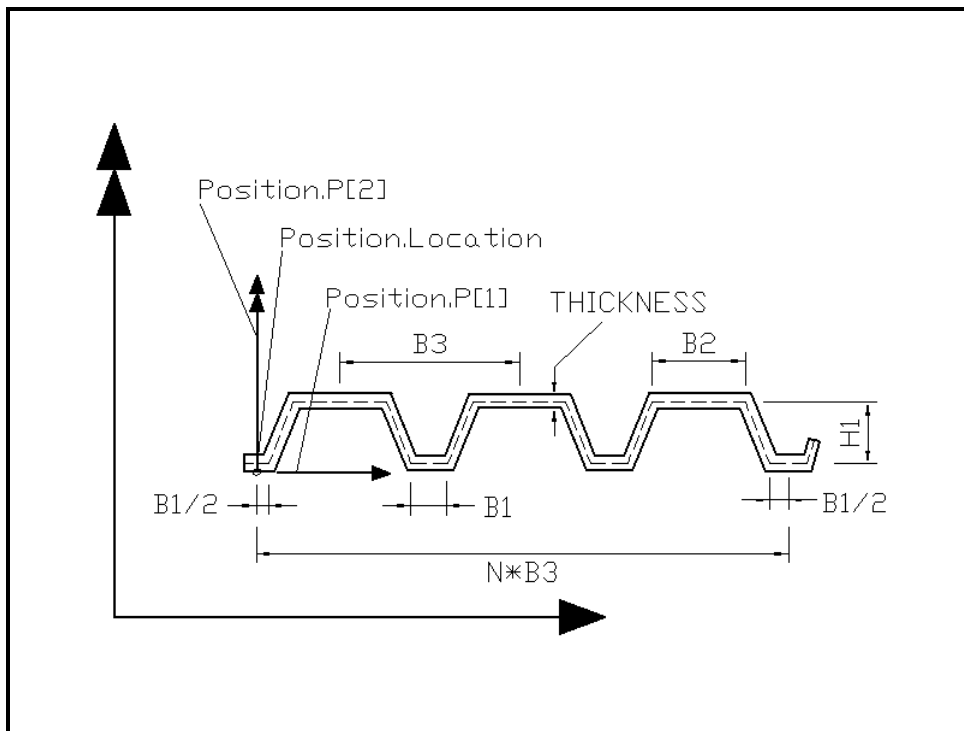


Abbildung A9: Geometrische Parameter von Trapezprofiltafeln

A.9.4 Klasse *IfcCorrugatedProfileDef*

EXPRESS-Beschreibung:

```
ENTITY IfcCorrugatedProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position : IfcAxis2Placement2D;
  THICKNESS : IfcPositiveLengthMeasure;
  L1 : IfcPositiveLengthMeasure;
  H1 : IfcPositiveLengthMeasure;
  N : IfcInteger;
  WHERE
    WR1 : N >= 1;
END_ENTITY;
```

Die Maße einer Profilstelle von Wellprofiltafeln gemäß Abbildung A10 beziehen sich auf die Profilmittellinie. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt im Tiefpunkt der äußersten Welle an der Profilunterkante. Die Verlegebreite einer Profiltafel ergibt sich von dort aus zu N -mal der Wellenlänge $L1$.

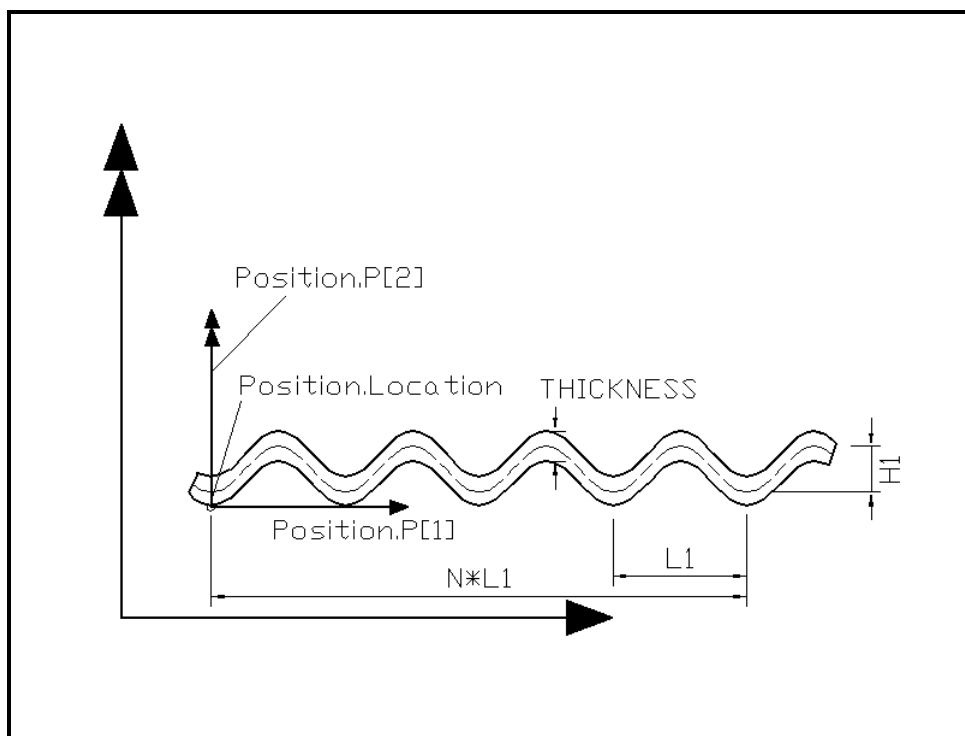


Abbildung A10: Geometrische Parameter von Wellprofiltafeln

A.9.5 Klasse *IfcTrapezoidalCompoundProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcTrapezoidalCompoundProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position      : IfcAxis2Placement2D;
  THICKNESS     : IfcPositiveLengthMeasure;
  B1            : IfcPositiveLengthMeasure;
  B2            : IfcPositiveLengthMeasure;
  B3            : IfcPositiveLengthMeasure;
  H1            : IfcPositiveLengthMeasure;
  N            : IfcInteger;
  CornerRadius  : OPTIONAL IfcPositiveLengthMeasure;
  WHERE
    WR1         : N >= 1;
END_ENTITY;

```

Die Maße einer Profilrippe von Verbunddeckenprofilen gemäß Abbildung A11 beziehen sich auf die Profilmittellinie. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in allen Eckbereichen. Ist dieser nicht angegeben, ist er zu Null anzunehmen. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilunterkante. Die Verlegebreite einer Profiltafel ergibt sich von dort aus zu N -mal der Rippenbreite $B3$.

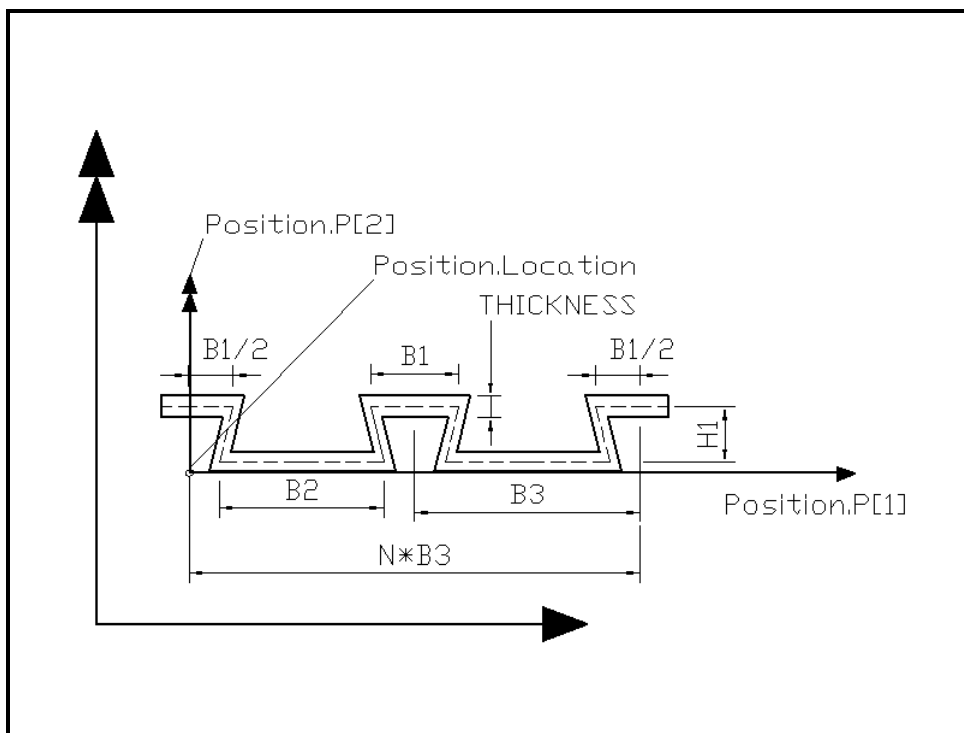


Abbildung A11: Geometrische Parameter von Verbunddeckenprofilen

A.9.6 Klasse *IfcRoofSheetingProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcRoofSheetingProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position      : IfcAxis2Placement2D;
  THICKNESS     : IfcPositiveLengthMeasure;
  B1            : IfcPositiveLengthMeasure;
  B2            : IfcPositiveLengthMeasure;
  B3            : IfcPositiveLengthMeasure;
  H1            : IfcPositiveLengthMeasure;
  N             : IfcInteger;
  InnerCornerRadius: OPTIONAL
                   IfcPositiveLengthMeasure;
  OuterCornerRadius: OPTIONAL
                   IfcPositiveLengthMeasure;

  WHERE
    WR1          : N >= 1;
END_ENTITY;

```

Die Maße einer Profilrippe von Pfannenblechprofilen gemäß Abbildung A12 beziehen sich auf die Profilmittellinie. Der Ausrundungsradius *InnerCornerRadius* gilt für die Mittellinie in hochstehenden Profilrippen; *OuterCornerRadius* für alle übrigen Eckbereiche. Ist eines dieser beiden Attribute nicht angegeben, ist der entsprechende Radius zu Null anzunehmen. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilverkantung. Die Verlegebreite einer Profiltafel ergibt sich von dort aus zu N -mal der Rippenbreite $B3$.

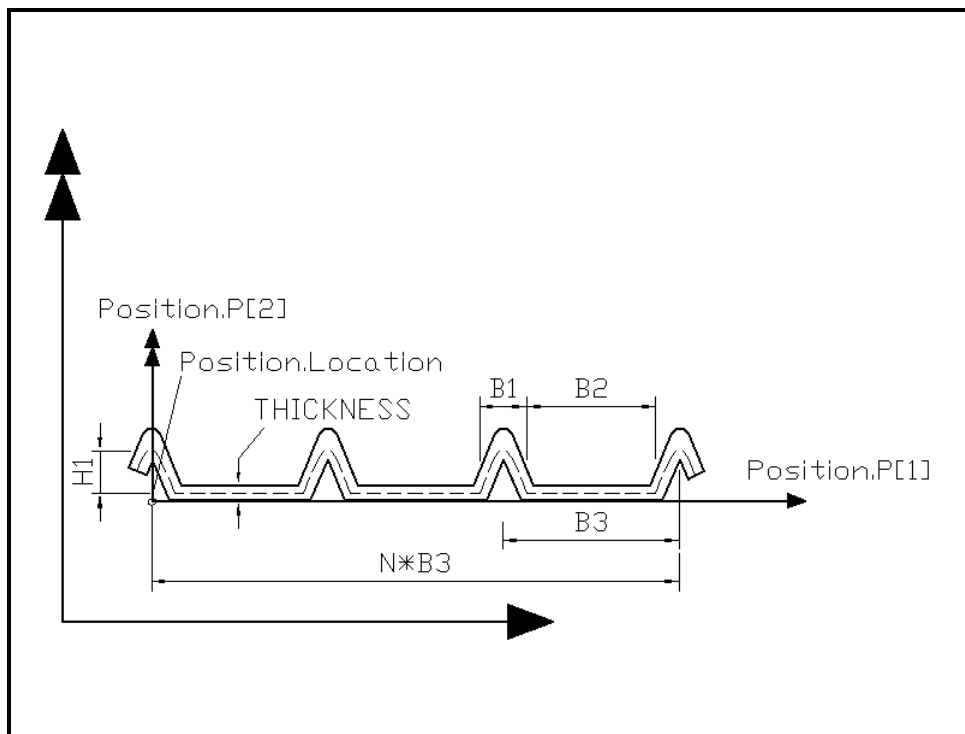


Abbildung A12: Geometrische Parameter von Pfannenblechprofilen

A.9.7 Klasse *IfcStandingSeamProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcStandingSeamProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position      : IfcAxis2Placement2D;
  THICKNESS     : IfcPositiveLengthMeasure;
  B1            : IfcPositiveLengthMeasure;
  B2            : IfcPositiveLengthMeasure;
  H1            : IfcPositiveLengthMeasure;
  H2            : IfcPositiveLengthMeasure;
  H3            : IfcPositiveLengthMeasure;
  CornerRadius  : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;

```

Die Maße von Stehfalzprofilen gemäß Abbildung A13 beziehen sich auf die Profilmittellinie, mit Ausnahme der Verlegebreite B1, die sich auf die Außenkanten der Stege bezieht. Die so definierte Form ist der eingebaute, also gefalzte Zustand. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in allen Eckbereichen außer an der Falz, für die keine Ausrundung angegeben wird. Ist das Attribut *CornerRadius* nicht angegeben, ist der Radius zu Null anzunehmen. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilunterkante.

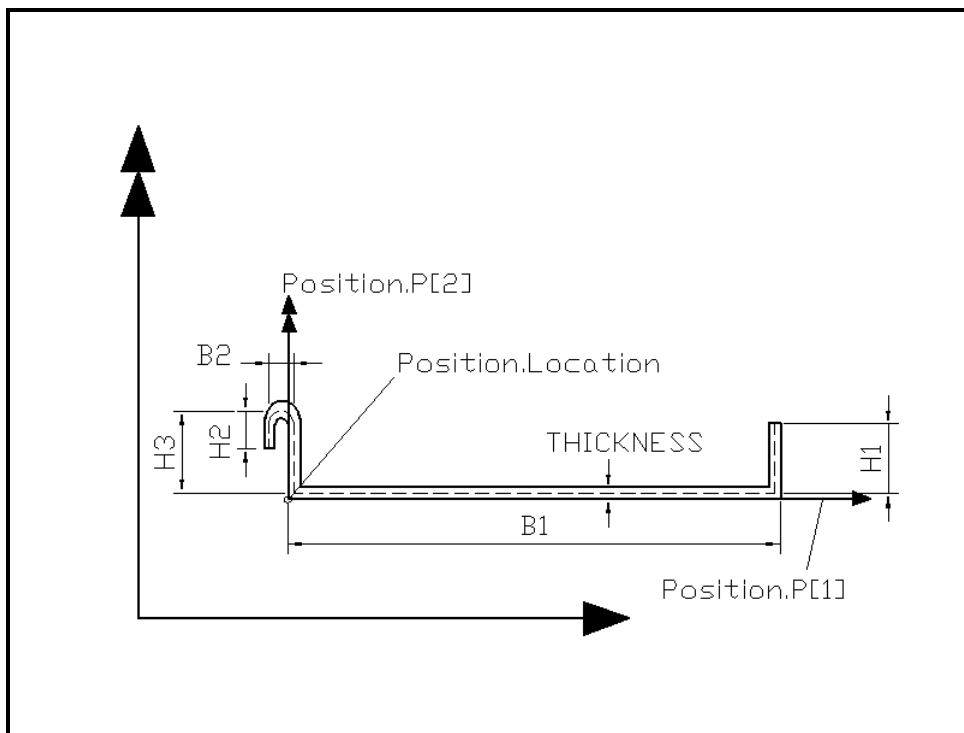


Abbildung A13: Geometrische Parameter von Stehfalzprofilen

A.9.8 Klasse *IfcFacadePanelProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcFacadePanelProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position      : IfcAxis2Placement2D;
  THICKNESS     : IfcPositiveLengthMeasure;
  B1            : IfcPositiveLengthMeasure;
  B2            : IfcPositiveLengthMeasure;
  B3            : IfcPositiveLengthMeasure;
  H1            : IfcPositiveLengthMeasure;
  H2            : IfcPositiveLengthMeasure;
  CornerRadius  : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;

```

Die Maße von Fassadenpaneelprofilen gemäß Abbildung A14 beziehen sich auf die Profilmittellinie. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in allen Eckbereichen. Ist dieser nicht angegeben, ist der Radius zu Null anzunehmen. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilverkantung.

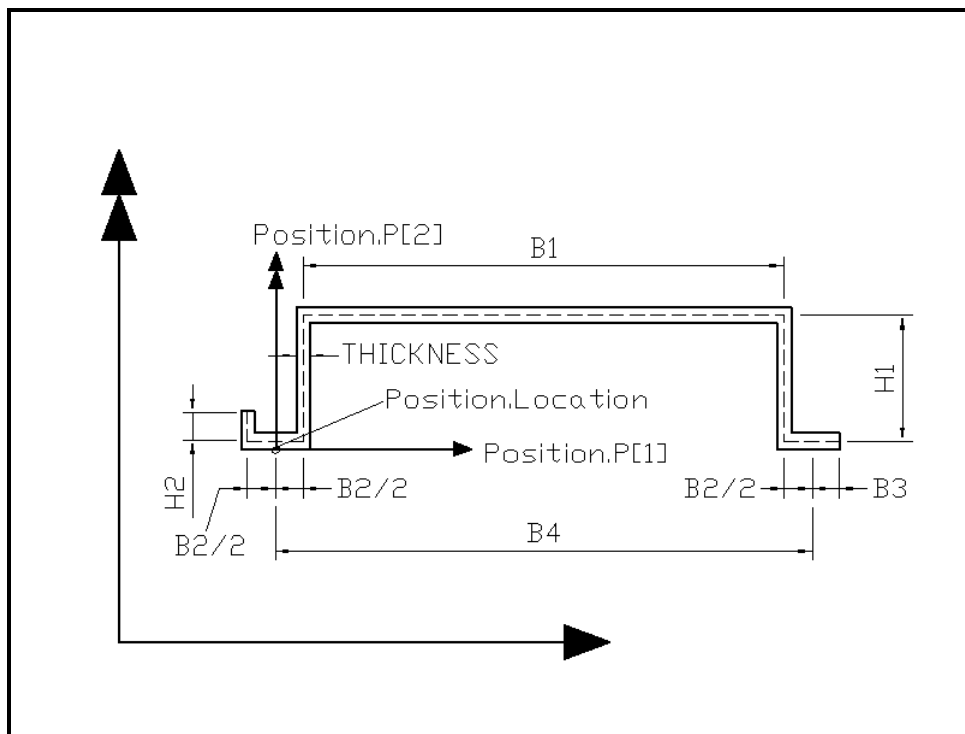


Abbildung A14: Geometrische Parameter von Fassadenpaneelprofilen

A.9.9 Klasse *IfcC-PanelProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcC-PanelProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position      : IfcAxis2Placement2D;
  THICKNESS     : IfcPositiveLengthMeasure;
  B1            : IfcPositiveLengthMeasure;
  B2            : IfcPositiveLengthMeasure;
  H1            : IfcPositiveLengthMeasure;
  A1            : IfcPositiveLengthMeasure;
  A2            : IfcPositiveLengthMeasure;
  ALPHA_1      : IfcPlaneAngleMeasure;
  CornerRadius  : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;

```

Im Gegensatz zu den anderen Profilen für Verlegebauteile beziehen sich die Maße von Kassettenprofilen gemäß Abbildung A15 nicht auf die Profilmittellinie, sondern auf die Kanten des Profils. Man beachte, daß die Attribute *B1* und *H1* am überlappenden Ende die Außenkontur beschreiben, am überlappenden Ende jedoch die Innenkontur. Der Ausrundungsradius *CornerRadius* gilt für die Mittellinie in den beiden Ecken zwischen Stegen und Gurt des Profils, jedoch nicht für die Überlappungsbe-
reiche. Ist das Attribut *CornerRadius* nicht angegeben, ist der Radius zu Null anzunehmen. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Des-
sen Ursprung liegt bündig an den Profilaußenkanten.

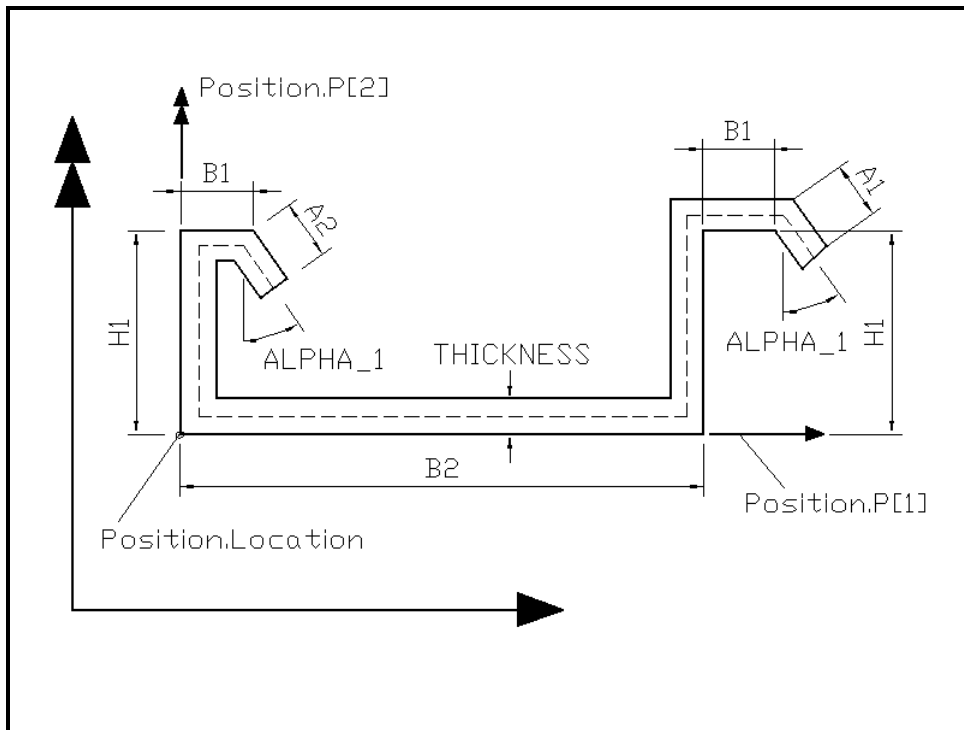


Abbildung A15: Geometrische Parameter von Kassettenprofilen

A.9.10 Klasse *IfcPlaneSandwichProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcPlaneSandwichProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position : IfcAxis2Placement2D;
  B1       : IfcPositiveLengthMeasure;
  B2       : IfcPositiveLengthMeasure;
  H1       : OPTIONAL IfcPositiveLengthMeasure;
  H2       : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;

```

Abbildung A16 zeigt die Bedeutung der geometrischen Attribute von ebenen Sandwich-Profilen. Es wird lediglich der Umriss des Profils mit optionaler, vereinfachter Querstoßausbildung definiert. Eine Unterscheidung in Geometrie der Deckschichten und des Kerns wird nicht gemacht. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt bündig an den Profilaußenkanten.

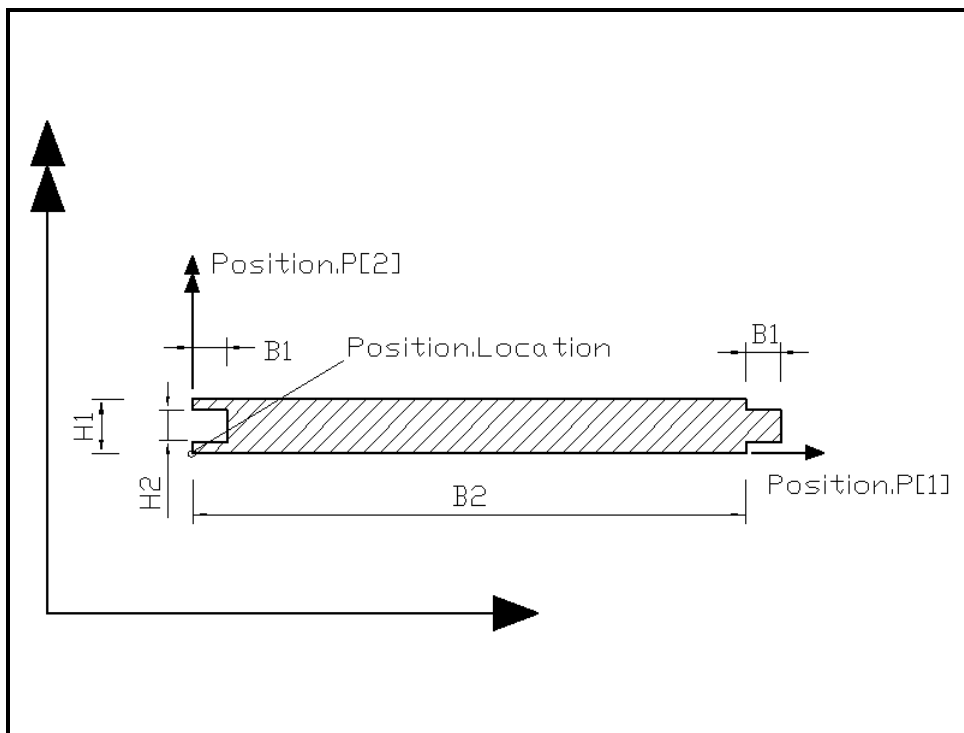


Abbildung A16: Geometrische Parameter von ebenen Sandwich-Profilen

A.9.11 Klasse *IfcProfiledSandwichProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcProfiledSandwichProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position : IfcAxis2Placement2D;
  B1       : IfcPositiveLengthMeasure;
  B2       : IfcPositiveLengthMeasure;
  B3       : IfcPositiveLengthMeasure;
  H1       : OPTIONAL IfcPositiveLengthMeasure;
  H2       : OPTIONAL IfcPositiveLengthMeasure;
  N        : IfcInteger;
  WHERE
    WR1    : N >= 1;
END_ENTITY;

```

Abbildung A17 zeigt die Bedeutung der geometrischen Attribute von einseitig profilierten Sandwich-Profilen, wobei alle Rippen identisch sind. Es wird lediglich der Umriß des Profils definiert. Eine Unterscheidung in Geometrie der Deckschichten und des Kerns wird nicht gemacht. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilunterkante. Die Verlegetbreite einer Sandwichtafel ergibt sich von dort aus zu N -mal der Rippenbreite $B3$.

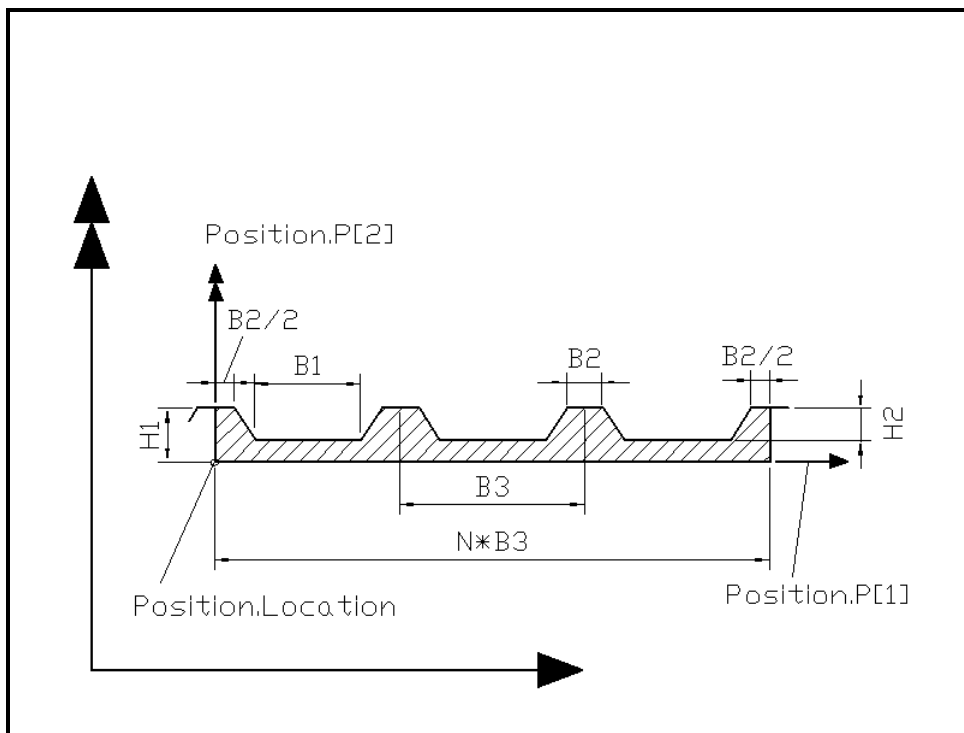


Abbildung A17: Geometrische Parameter von einseitig profilierten Sandwich-Profilen

A.9.12 Klasse *IfcDoubleProfiledSandwichProfileDef*

EXPRESS-Beschreibung:

```

ENTITY IfcDoubleProfiledSandwichProfileDef
  SUBTYPE OF (IfcProfileDef);
  Position : IfcAxis2Placement2D;
  B1       : IfcPositiveLengthMeasure;
  B2       : IfcPositiveLengthMeasure;
  B3       : IfcPositiveLengthMeasure;
  B4       : IfcPositiveLengthMeasure;
  H1       : OPTIONAL IfcPositiveLengthMeasure;
  H2       : OPTIONAL IfcPositiveLengthMeasure;
  H3       : OPTIONAL IfcPositiveLengthMeasure;
  N        : IfcInteger;
  WHERE
    WR1    : N >= 1;
END ENTITY;

```

Abbildung A18 zeigt die Bedeutung der geometrischen Attribute von doppelseitig profilierten Sandwich-Profilen mit optionaler, vereinfachter Querstoßausbildung. Alle Rippen sind identisch und das Profil ist spiegelsymmetrisch zur Horizontalen. Es wird lediglich der Umriss des Profils definiert. Eine Unterscheidung in Geometrie der Deckschichten und des Kerns wird nicht gemacht. Das Attribut *Position* definiert das lokale Einfügekoordinatensystem. Dessen Ursprung liegt in einem gedachten Bezugspunkt an der Profilunterkante. Die Verlegebreite einer Sandwichtafel ergibt sich von dort aus zu N -mal der Rippenbreite $B4$.

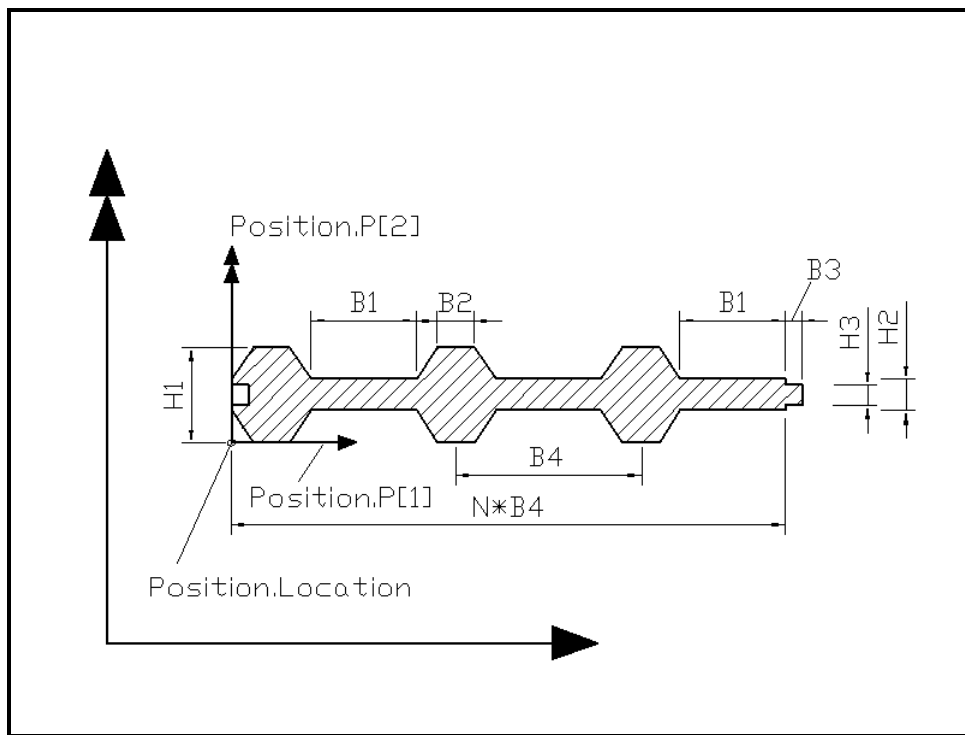


Abbildung A18: Geometrische Parameter von doppelseitig profilierten Sandwich-Profilen

B Anforderungen

B.1 *Pset_RoofAndWallRequirements*

Definition des PropertySet:

PropertySetName	Pset_RoofAndWallRequirements
Definition	Anforderungen an die Bauelemente <i>IfcRoof</i> , <i>IfcWall</i> und <i>IfcSlab</i> .

Definition der Anforderungen:

Name	Property Type	Data Type	Definition
ConstructionType	<i>Property-Enumerated-Value</i>	<i>Pset_RoofAndWall-ConstructionEnum</i> <ul style="list-style-type: none"> • Brick • CastInPlace-Concrete • PreCastConcrete • AeratedConcrete • Trapezoidal • C-Panel • Sandwich • NotRequired 	Geforderte Bauart des Elementes: Mauerwerk Ortbeton Betonfertigteil Gasbeton Tragschicht aus Trapezprofilen Tragschicht aus Kassettenprofilen Sandwich Bauart nicht gefordert
Cladding	<i>Property-EnumeratedValue</i>	<i>Pset_RoofAndWall-CladdingEnum</i> <ul style="list-style-type: none"> • Steel • Aluminum • NotRequired 	Vorgehängte, hinterlüftete Fassade, sofern erforderlich: Stahl Aluminium Keine gefordert
FireRating*	<i>IfcProperty-SingleValue</i>	<i>IfcString</i>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm
Acoustic-Rating*	<i>IfcProperty-SingleValue</i>	<i>IfcReal</i>	Erforderliches Schalldämm-Maß in dB
Load-Bearing*	<i>IfcProperty-SingleValue</i>	<i>IfcBoolean</i>	Tragendes (TRUE) oder nichttragendes Bauelement (FALSE)

Thermal Transmittance-Coefficient *	<i>IfcPropertySingleValue</i>	<i>IfcThermalTransmittanceMeasure</i>	Maximal erlaubter Wärmedurchgangskoeffizient **
Thermal-Resistance	<i>IfcPropertySingleValue</i>	<i>IfcThermalResistanceMeasure</i>	Erforderlicher Wärmedurchlaßwiderstand **
SpecialRequirements	<i>IfcPropertySingleValue</i>	<i>IfcText</i>	Weitere Anforderungen in Textform

*) Diese Anforderungen sind zum Teil in *Pset_RoofCommon* und *Pset_WallCommon* der IFC2x definiert.

***) Die Einheiten werden im globalen Konzept der IFC projektspezifisch festgelegt.

B.2 *Pset_GlassFaceRequirements*

Definition des PropertySet:

PropertySetName	Pset_GlassFaceRequirements
Definition	Anforderungen an die Bauelemente <i>IfcGlassFacade</i> und <i>IfcGlassFace</i> .

Definition der Anforderungen:

Name	Property Type	Data Type	Definition
Glazing-Support-Type	<i>PropertyEnumeratedValue</i>	<i>Pset_GlazingSupportTypeEnum</i> <ul style="list-style-type: none"> • PointLikeSupport • IdealPointSupport • GlazingProfileSupport • StructuralSealantGlazing • NotRequired 	Geforderte Bauart. <u>Unterscheidungskriterium ist die Lagerung:</u> Punktförmige Halterung Ideale Punkthalterung Preßleistenverglasung (Pfosten-Riegel-Bauart) StructuralSealantGlazing Keine bestimmte gefordert
Glazing-Type	<i>PropertyEnumeratedValue</i>	<i>Pset_GlazingTypeEnum</i> <ul style="list-style-type: none"> • SinglePaneFloatGlass • SinglePaneCastGlass • SinglePaneTemperedSafetyGlass 	Geforderte Art der Verglasung: Einfachverglasung aus Floatglas Einfachverglasung aus Gußglas Thermisch vorgespanntes Einscheibensicherheitsglas

		<ul style="list-style-type: none"> • SinglePaneSafetyGlassWithFilm • SinglePanePartlyTemperedGlass • CompoundGlass • CompoundSafetyGlass • DoublePaneInsulatingGlass • TriplePaneInsulatingGlass • UserDefined • NotRequired 	<p>Einscheibensicherheitsglas mit Folie</p> <p>Thermisch teilvorgespanntes Einscheibenglas</p> <p>Verbundglas</p> <p>Verbundsicherheitsglas</p> <p>Zwei-Scheiben-Isolierglas</p> <p>Drei-Scheiben-Isolierglas</p> <p>Benutzerdefiniert</p> <p>Keine bestimmte Gefordert</p>
SafetyRequirement	Property-Enumerated-Value	<i>Pset_GlazingSecurityEnum</i> <ul style="list-style-type: none"> • OverHeadGlazing • None 	<p><u>Sicherheitsanforderungen aufgrund von:</u></p> <p>Überkopfverglasung</p> <p>Keine Sicherheitsanforderung</p>
Glazing-Thickness	IfcProperty-SingleValue	<i>IfcReal</i>	Statisch erforderliche Glasdicke
FireRating	IfcProperty-SingleValue	<i>IfcString</i>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm
Heat-Radiation-Resistance	IfcProperty-SingleValue	<i>IfcString</i>	Erforderlicher Widerstand gegen Hitzestrahlung mit Angabe der gültigen Norm
Acoustic-Rating	IfcProperty-SingleValue	<i>IfcReal</i>	Erforderliches Schalldämmmaß in dB
Thermal-Transmittance-Coefficient	IfcProperty-SingleValue	<i>IfcThermalTransmittanceMeasure</i>	Maximal erlaubter Wärmedurchgangskoeffizient**
Thermal-Resistance	IfcProperty-SingleValue	<i>IfcThermalResistanceMeasure</i>	Erforderlicher Wärmedurchlaßwiderstand**
RainImpermeability	IfcProperty-SingleValue	<i>IfcString</i>	Schlagregendichtigkeit mit Angabe der gültigen Norm
Colour	IfcProperty-SingleValue	<i>IfcString</i>	Geforderte Färbung oder Tönung
SpecialRequirements	IfcProperty-SingleValue	<i>IfcText</i>	Weitere Anforderungen in Textform

***) Die Einheiten werden im globalen Konzept der IFC projektspezifisch festgelegt.

B.3 Pset_DoorIndustrialRequirements

Definition des PropertySet:

PropertySetName	Pset_DoorIndustrialRequirements
Definition	Anforderungen an die Typdefinition <i>IfcDoorIndustrialStyle</i>

Definition der Anforderungen:

Name	Property Type	Data Type	Definition
Special-Purpose	<i>Property-Enumerated-Value</i>	<i>Pset_DoorIndustrial-SpecialPurposeEnum</i> <ul style="list-style-type: none"> • FireProtection • SmokeProtection • RadiationProtection • AcousticInsulation • None 	<u>Besonderes Einsatzgebiet:</u> Brandschutztor Rauchschutztor Strahlenschutztor Schallschutztor Kein besonderes gefordert
SelfClosing	<i>IfcProperty-SingleValue</i>	<i>IfcBoolean</i>	<u>Selbstschließendes Tor:</u> Ja (TRUE) oder Nein (FALSE)
Rapid-Operation	<i>IfcProperty-SingleValue</i>	<i>IfcBoolean</i>	<u>Schnellaufstor:</u> Ja (TRUE) oder Nein (FALSE)
FireRating	<i>IfcProperty-SingleValue</i>	<i>IfcString</i>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm
Acoustic-Rating*	<i>IfcProperty-SingleValue</i>	<i>IfcReal</i>	Erforderliches Schalldämm-Maß in dB
Thermal Transmittance-Coefficient	<i>IfcProperty-SingleValue</i>	<i>IfcThermalTransmittanceMeasure</i>	Maximal erlaubter Wärmedurchgangskoeffizient**
SpecialRequirements	<i>IfcProperty-SingleValue</i>	<i>IfcText</i>	Weitere Anforderungen in Textform

***) Die Einheiten werden im globalen Konzept der IFC projektspezifisch festgelegt.

B.4 *Pset_RooflightElementRequirements*

Definition des PropertySet:

PropertySetName	Pset_RooflightElementRequirements
Definition	Anforderungen an die Typdefinition <i>IfcRooflightElementStyle</i>

Definition der Anforderungen:

Name	Property Type	Data Type	Definition
SmokeAnHeat-Ventilation	<i>IfcProperty-SingleValue</i>	<i>IfcBoolean</i>	Rauch- und Wärmeabzugsfunktion: Ja (TRUE) oder Nein (FALSE)
Effective-OpeningArea:	<i>IfcProperty-SingleValue</i>	<i>IfcReal</i>	Erforderliche aerodynamisch wirksame Öffnungsfläche zum Rauch- und Wärmeabzug
FireRating	<i>IfcProperty-SingleValue</i>	<i>IfcString</i>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm
Thermal-Transmittance-Coefficient	<i>IfcProperty-SingleValue</i>	<i>IfcThermal-TransmittanceMeasure</i>	Maximal erlaubter Wärmedurchgangskoeffizient**
SpecialRequirements	<i>IfcProperty-SingleValue</i>	<i>IfcText</i>	Weitere Anforderungen in Textform

***) Die Einheiten werden im globalen Konzept der IFC projektspezifisch festgelegt.

B.5 *Pset_HeatAndSmokeControlUnitRequirements*

Definition des PropertySet:

PropertySetName	Pset_HeatAndSmokeControlUnitRequirements
Definition	Anforderungen an die Typdefinition <i>IfcHeatAndSmokeControlUnitStyle</i>

Definition der Anforderungen:

Name	Property Type	Data Type	Definition
Effective-OpeningArea	<i>IfcProperty-SingleValue</i>	<i>IfcReal</i>	Erforderliche aerodynamisch wirksame Öffnungsfläche zum Rauch- und Wärmeabzug
SpecialRequirements	<i>IfcProperty-SingleValue</i>	<i>IfcText</i>	Weitere Anforderungen in Textform

B.6 Pset_SlabRequirements

Definition des PropertySet:

PropertySetName	Pset_SlabRequirements
Definition	Anforderungen an das Bauelement <i>IfcSlab</i>

Definition der Anforderungen:

Name	Property Type	Data Type	Definition
ConstructionType	<i>Property-EnumeratedValue</i>	<i>Pset_RoofAndWall-ConstructionEnum</i> <ul style="list-style-type: none"> • Brick • CastInPlace-Concrete • CompoundSteel-Concrete • PreCastConcrete • AeratedConcrete • Trapezoidal • C-Panel • Sandwich • NotRequired 	Geforderte Bauart des Elementes: Mauerwerk Ortbeton Verbund mit Stahl und Beton Betonfertigteil Gasbeton Tragschicht aus Trapezprofilen Tragschicht aus Kassettenprofilen Sandwich Keine bestimmte gefordert
Cladding	<i>Property-EnumeratedValue</i>	<i>Pset_RoofAndWall-CladdingEnum</i> <ul style="list-style-type: none"> • Steel • Aluminum • NotRequired 	<u>Vorgehängte, hinterlüftete Fassade, sofern gefordert:</u> Stahl Aluminium Keine gefordert
FireRating*	<i>IfcProperty-SingleValue</i>	<i>IfcString</i>	Erforderliche Feuerwiderstandsdauer in Minuten oder Widerstandsklasse mit Angabe der gültigen Norm
Acoustic-Rating*	<i>IfcProperty-SingleValue</i>	<i>IfcReal</i>	Erforderliches Schalldämm-Maß in dB
Load-Bearing*	<i>IfcProperty-SingleValue</i>	<i>IfcBoolean</i>	Tragendes (TRUE) oder nichttragendes Bauelement (FALSE)
Thermal-Transmittance-Coefficient*	<i>IfcProperty-SingleValue</i>	<i>IfcThermalTransmittanceMeasure</i>	Maximal erlaubter Wärmedurchgangskoeffizient**

Thermal-Resistance	<i>IfcProperty- SingleValue</i>	<i>IfcThermalResis- tanceMeasure</i>	Erforderlicher Wärmedurchlaßwiderstand**
SpecialRe- quirements	<i>IfcProperty- SingleValue</i>	<i>IfcText</i>	Weitere Anforderungen in Textform

*) Diese Anforderungen sind zum Teil in *Pset_SlabCommon* der IFC2x definiert.

***) Die Einheiten werden im globalen Konzept der IFC projektspezifisch festgelegt.

B.7 *Pset_BeamAndColumnRequirements*

Definition des PropertySet:

PropertySetName	Pset_BeamAndColumnRequirements
Definition	Anforderungen an die Bauelemente <i>IfcBeam</i> und <i>IfcColumn</i>

Definition der Anforderungen:

Name	Property Type	Datentyp	Definition
Construc- tionType	<i>Property- Enumera- tedValue</i>	<i>Pset_RoofAndWall- ConstructionEnum</i>	Geforderte Bauart des Elementes: Ortbeton Verbundstütze aus Stahl und Beton Betonfertigteil Stahl Keine bestimmte gefordert
FireRating	<i>IfcProperty- SingleValue</i>	<i>IfcString</i>	Erforderliche Feuerwiderstands- dauer in Minuten oder Wider- standsklasse mit Angabe der gül- tigen Norm
Fire- Protection	<i>IfcProperty- SingleValue</i>	<i>IfcString</i>	Erforderliche Brandschutzmaß- nahmen
LoadBearing	<i>IfcProperty- SingleValue</i>	<i>IfcBoolean</i>	Tragendes (TRUE) oder nichttra- gendes Bauelement (FALSE)
SpecialRe- quirements	<i>IfcProperty- SingleValue</i>	<i>IfcText</i>	Weitere Anforderungen in Text- form

C Bezeichnungen für Befestigungsmittel

C.1 Vereinbarungen

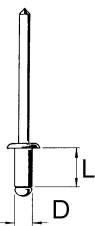
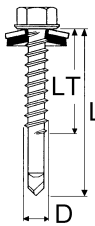
Längenangaben:

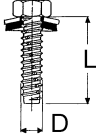
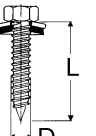
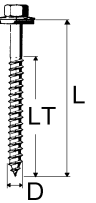
Alle Längen werden in Millimetern angegeben. Als Dezimaltrennzeichen wird ein Punkt verwendet. Ganze Zahlen werden ebenfalls mit einer Nachkommastelle angegeben (z.B. 10.0). Die Zahl der möglichen Nachkommastellen ist nicht begrenzt.

Kürzel für Werkstoffe:

- AL = Aluminium und Aluminiumlegierungen
- S = Stahl
- GS = Galvanized Steel = Verzinkter Stahl
- HS = Hardened Steel = Gehärteter Stahl
- SS = Stainless Steel = Nichtrostender Stahl
- MO = Monel
- ND = Not Defined = Keine Angabe

C.2 Befestigungsmittel für Bleche

Bezeichnung	Erläuterung	Beispiel
Blindniet (Blind Rivet) BR<D>X<L>- <MAT>-<ADD>	 <p>MAT: Hülsenmaterial ADD: - R = Regular = Normaler Blindniet - S = Sealing = Dicht-Blindniet</p>	BR4.8X10.0-AL-R
Bohrschraube (Self Drilling Screw) SDS<D>X<L>X< LT>-<MAT>- <ADD>	 <p>MAT: Material für Schraube und Scheibe ADD: - R = Regular = Normale Schraube - SW = Sealing Washer = Mit Dichtscheibe</p>	SDS5.5X20.0X 10.0-SS-SW

<p>Gewindeschneidende Schraube (Thread Cutting Screw) TCS<D>X<L>- <MAT>-<ADD></p>		<p>MAT: Material für Schraube und Scheibe ADD: - R = Regular = Normale Schraube - SW = Sealing Washer = Mit Dichtscheibe</p>	<p>TCS6.3X16.0-SS-R</p>
<p>Gewindefurchende Schraube (Thread Rolling Screw) TRS<D>X<L>- <MAT>-<ADD></p>		<p>MAT: Material für Schraube und Scheibe ADD: - R = Regular = Normale Schraube - SW = Sealing Washer = Mit Dichtscheibe</p>	<p>TRS6.3X16.0-SS-SW</p>
<p>Holzschraube (Wood Screw) WS<D>X<L>X<LT>- <MAT>-<ADD></p>		<p>MAT: Material für Schraube und Scheibe ADD: - R = Regular = Normale Schraube - SW = Sealing Washer = Mit Dichtscheibe</p>	<p>WS5.5X20.0X10.0-GS-R</p>

LEBENS LAUF

Persönliche Daten:

Vor- und Zuname: Christoph Hörenbaum

Geburtsdatum: 22. Oktober 1968

Geburtsort: Lippstadt

Staatsangehörigkeit: deutsch

Ausbildung:

1974 bis 1988: Grundschule und Evangelisches Gymnasium
in Lippstadt

1988: Abitur

1988 bis 1989: Wehrdienst in Lippstadt

1989 bis 1996: Studium des Bauingenieurwesens
an der Universität Karlsruhe (TH)
Vertiefungsrichtung: Konstruktiver Ingenieurbau
Vertiefungsschwerpunkt: Stahlbau

Juli bis November 1996: Diplomarbeit an der University of Toronto, Kanada

Beruflicher Werdegang:

1991 bis 1995: Studentischer Mitarbeiter am Lehrstuhl für Stahl- und
Leichtmetallbau der Universität Karlsruhe (TH)

Seit Dezember 1996: Wissenschaftlicher Mitarbeiter am Lehrstuhl für Stahl-
und Leichtmetallbau der Universität Karlsruhe (TH)

