BETTER POLYLINE DEFINITION

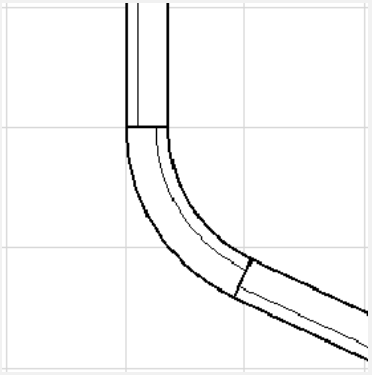
Simpler, more compact, less error-prone
a proposal for IFC4 Add1 (and the reference / design transfer views)

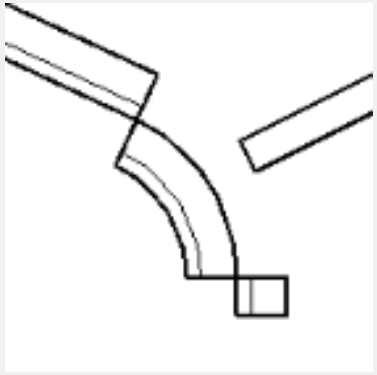
Thomas Liebich – Discussion status July/August 2014

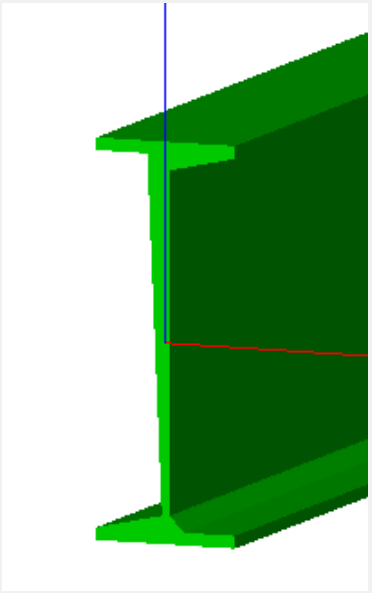
Current status

complex definition for polylines with circular arcs

- counts for > 95% of all polylines
- difficult definition of arcs
  - too many possibilities
  - order, two sense flags
- after >10 years of implementation, still exchange errors
  - “IFC doesn’t work”







Complexity of definition coming from STEP

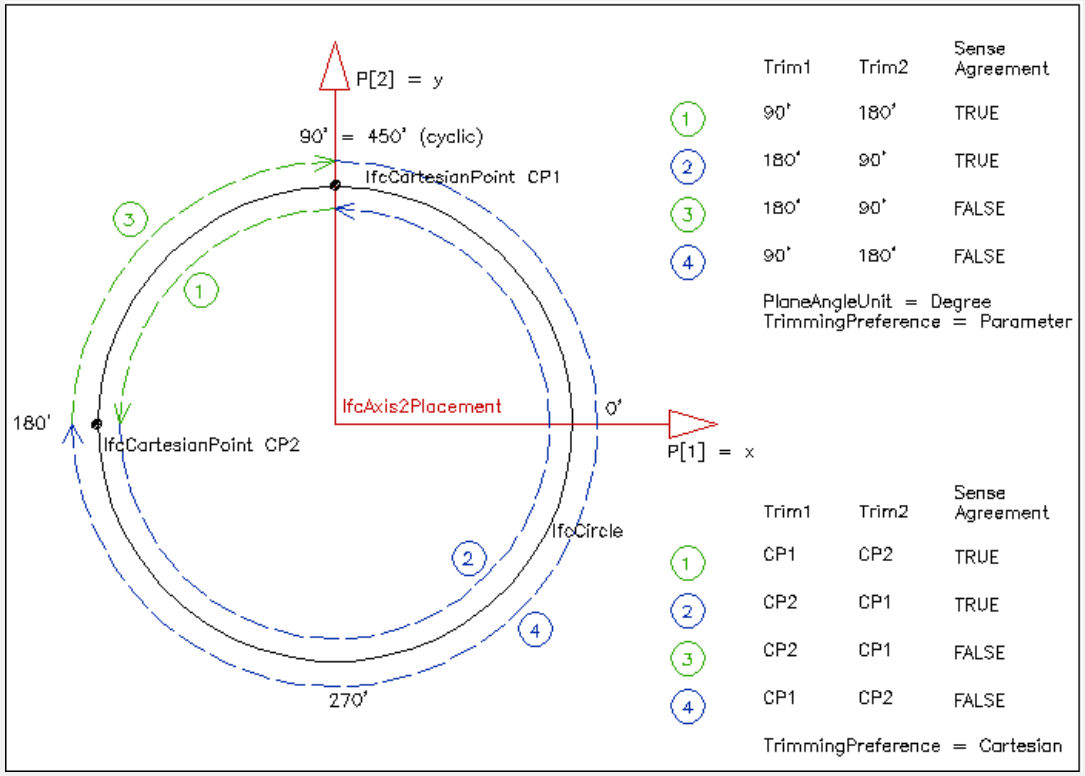
IfcTrimmedCurve = trimmed_curve

- 8 variances

IfcCompositeCurveSegment = composite_curve_segment

- 2 variances

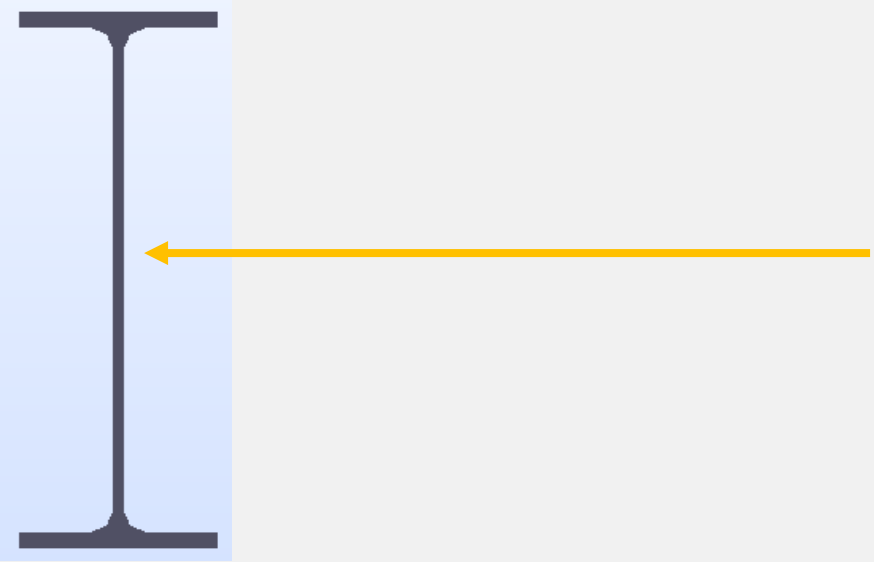
together: 16 variances



Big file sizes

defining this double-I profile

- 61 entities (lines)



```
#63= IFCCOMPOSITECURVE((#64,#68,#75,#78,#81,#84,#87,#90,#97,#100,#107,#110,#113,#116,#119,#122),.U.);
#64= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#67);
#65= IFCCARTESIANPOINT((0.257,0.006));
#66= IFCCARTESIANPOINT((-0.257,0.006));
#67= IFCPOLYLINE((#66,#65));
#68= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#69);
#69= IFCTRIMMEDCURVE(#74,(IFCPARAMETERVALUE(0.0),#65),(IFCPARAMETERVALUE(1.5707963267949),#70),.T.,.PARAMETER.);
#70= IFCCARTESIANPOINT((0.281,0.03));
#71= IFCAXIS2PLACEMENT2D(#72,#73);
#72= IFCCARTESIANPOINT((0.257,0.03000000000000001));
#73= IFCDIRECTION((0.0,-1.0));
#74= IFCCIRCLE(#71,0.024);
#75= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#77);
#76= IFCCARTESIANPOINT((0.281,0.11));
#77= IFCPOLYLINE((#70,#76));
#78= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#80);
#79= IFCCARTESIANPOINT((0.3,0.11));
#80= IFCPOLYLINE((#76,#79));
#81= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#83);
#82= IFCCARTESIANPOINT((0.3,-0.11));
#83= IFCPOLYLINE((#79,#82));
#84= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#86);
#85= IFCCARTESIANPOINT((0.281,-0.11));
#86= IFCPOLYLINE((#82,#85));
#87= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#89);
#88= IFCCARTESIANPOINT((0.281,-0.03));
#89= IFCPOLYLINE((#85,#88));
#90= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#91);
#91= IFCTRIMMEDCURVE(#96,(IFCPARAMETERVALUE(0.0),#88),(IFCPARAMETERVALUE(1.57079632679489),#92),.T.,.PARAMETER.);
#92= IFCCARTESIANPOINT((0.257,-0.006000000000000004));
#93= IFCAXIS2PLACEMENT2D(#94,#95);
#94= IFCCARTESIANPOINT((0.257,-0.03000000000000001));
#95= IFCDIRECTION((1.0,0.0));
#96= IFCCIRCLE(#93,0.024);
#97= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#99);
#98= IFCCARTESIANPOINT((-0.257,-0.006));
#99= IFCPOLYLINE((#92,#98));
#100= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#101);
#101= IFCTRIMMEDCURVE(#106,(IFCPARAMETERVALUE(0.0),#98),(IFCPARAMETERVALUE(1.5707963267949),#102),.T.,.PARAMETER.);
#102= IFCCARTESIANPOINT((-0.281,-0.03));
#103= IFCAXIS2PLACEMENT2D(#104,#105);
#104= IFCCARTESIANPOINT((-0.257,-0.03000000000000001));
#105= IFCDIRECTION((0.0,1.0));
#106= IFCCIRCLE(#103,0.024);
#107= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#109);
#108= IFCCARTESIANPOINT((-0.281,-0.11));
#109= IFCPOLYLINE((#102,#108));
#110= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#112);
#111= IFCCARTESIANPOINT((-0.3,-0.11));
#112= IFCPOLYLINE((#108,#111));
#113= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#115);
#114= IFCCARTESIANPOINT((-0.3,0.11));
#115= IFCPOLYLINE((#111,#114));
#116= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#118);
#117= IFCCARTESIANPOINT((-0.281,0.11));
#118= IFCPOLYLINE((#114,#117));
#119= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#121);
#120= IFCCARTESIANPOINT((-0.281,0.03));
#121= IFCPOLYLINE((#117,#120));
#122= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#123);
#123= IFCTRIMMEDCURVE(#127,(IFCPARAMETERVALUE(0.0),#120),(IFCPARAMETERVALUE(1.57079632679489),#66),.T.,.PARAMETER.);
#124= IFCAXIS2PLACEMENT2D(#125,#126);
#125= IFCCARTESIANPOINT((-0.257,0.03000000000000001));
#126= IFCDIRECTION((-1.0,0.0));
#127= IFCCIRCLE(#124,0.024);
```

new considerations in STEP

STEP-NC

- requirement to define tool paths efficiently and with no interpretation error
- led to a refinement of STEP geometry in AP238

```
*)
ENTITY via_arc_point
  SUBTYPE OF (cartesian_point);
  WHERE
  WR1: SIZEOF(USEDIN(SELF, 'INTEGRATED_CNC_SCHEMA.POLYLINE.POINTS')) > 0;
  WR2: (0 = SIZEOF (QUERY (pl <*
        USEDIN(SELF, 'INTEGRATED_CNC_SCHEMA.POLYLINE.POINTS') |
        ((pl.points[1] = SELF) OR (pl.points[HiIndex(pl.points)] = SELF))
        )));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **via_arc_point** shall appear in the **points** list of at least one **polyline**.

WR2: The **via_arc_point** shall not appear as either the first or last element in the **points** list of any **polyline**.

5.2.3.1.94 via_arc_point

A **via_arc_point** is a **cartesian_point**. When appearing in the **points** list of a **polyline**, the **via_arc_point** defines an arc starting at the previous point in the **polyline**, passing through the **via_arc_point**, and ending at the next point in the **polyline**. The arc defined by the **via_arc_point** shall be less than 2π.

If two via points are considered coincident if any two consecutive points in the via re considered coincident, they shall be considered to decrine a straight line secgrma rather than an arc. context of enclosing representation.

NOTE 1 Since the arc defined by a via is less than 2π, a full circle is described using more than one via point. The use of a via point to describe an arc is also found in the circular_path entity in ISO 10303-105[3].

If a via point and the preceeding or following point in a polyline are considered coincident, the via point shall be considered to define straight line segments rather than an arc.

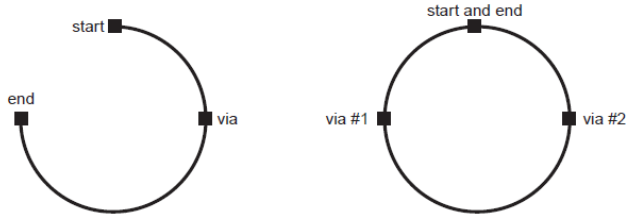
NOTE 2 Two points are considered coincident if they lie within the uncertainty distance given by the global_uncertainty_assigned_context of the enclosing representation.

EXAMPLE In the exchange file fragment below, #10 describes a 270 degree arc on the XY plane, starting at point #20, passing through point #30 and ending at point #40. Instance #50 describes a complete circle on the XY plane, starting and ending at point #60, passing through point #70 and point #80.

```
/* 270 degree arc, centered at (0,0,0) */
#10=POLYLINE('',(#20,#30,#40));
#20=CARTESIAN_POINT('start',(1.,0));
#30=VIA_ARC_POINT('via',(1.,0,0));
#40=CARTESIAN_POINT('end',(-1.,0,0));

/* complete circle, centered at (0,0,0) */
#50=POLYLINE('',(#60,#30,#40,#60));
#60=CARTESIAN_POINT('start and end',(0,1.,0));
#40=VIA_ARC_POINT('via #1',(-1.,0,0));
#30=VIA_ARC_POINT('via #2',(1.,0,0));
```

Figure 53 illustrates the arc and a complete circle described by the exchange file fragment.



Pros and cons of STEP-NC solution

Pros:

- small file sizes
- easy to understand
- minor modification of existing schema (polyline unchanged, Cartesian point subtype added)

Cons:

- specific agreement on full circle adds to potential confusion
- subtyping from Cartesian point based on role of point not elegant
- not easily extensible (e.g. quadric or Bezier curves)
- no easy detection of new functionality for upward compatibility (since same polyline entity is used)

Proposal 1

corresponding lists

```
ENTITY IfcPolyArcLine
 SUBTYPE OF (IfcBoundedCuve);
  Points : IfcCartesianPointList;
  Roles  : LIST [2:?] OF IfcRoleInPath;
END_ENTITY;

TYPE IfcRoleInPath = ENUMERATION OF
 (PNT,
  POC)
END_TYPE;
```

example

```
#65= IFCPOLYARCLINE(#66,(.PNT.,.PNT.,.POC.,.PNT.,.PNT.,
.PNT.,.PNT.,.PNT.,.PNT.,.POC.,.PNT.,.PNT.,.POC.,.PNT.,.PNT.,.
PNT.,.PNT.,.PNT.,.PNT.,.POC.,.PNT.));
#66= IFCCARTESIANPOINTLIST2D((-0.257,0.006),(0.257,0.006),
(0.2739706,0.0130294),(0.281,0.03),(0.281,0.11),(0.3,0.11),
(0.3,-0.11),(0.281,-0.11),(0.281,-0.03),(0.2739706,
-0.0130294),(0.257,-0.006),(-0.257,-0.006),(-0.2739706,
-0.0130294),(-0.281,-0.03),(-0.281,-0.11)),(-0.3,-0.11),
(-0.3,0.11),(-0.281,0.11),(-0.281,0.03),(-0.2739706,
0.0130294),(-0.257,0.006));
```

```
<IfcPolyArcLine Roles="pnt pnt poc pnt pnt pnt pnt pnt pnt
poc pnt pnt poc pnt pnt pnt pnt pnt pnt poc pnt">
  <Points xsi:type="IfcCartsesianPointList2D" CoordList="
-0.257 0.006 0.257 0.006 0.2739706 0.0130294 0.281 0.03 0.281
0.11 0.3 0.11 0.3 -0.11 0.281 -0.11 0.281 -0.03 0.2739706
-0.0130294 0.257 -0.006 -0.257 -0.006 -0.2739706
-0.0130294 -0.281 -0.03 -0.281 -0.11 -0.3 -0.11 -0.3 0.11 -
0.281 0.11 -0.281 0.03 -0.2739706 0.0130294 -0.257 0.006"/>
</IfcPolyArcLine>
```

Proposal 2

list with roles and point indices

```
ENTITY IfcPolyArcLine
 SUBTYPE OF (IfcBoundedCuve);
  Points : IfcCartesianPointList;
  Roles  : LIST [1:?] OF IfcPointsInPath;
END_ENTITY;

TYPE IfcPointsInPath = SELECT
 (IfcRoleInPath,
  IfcIndexList)
END_TYPE;

TYPE IfcRoleInPath = ENUMERATION OF
 (SGMT,
  ARC)
END_TYPE;

TYPE IfcIndexList =
 LIST[1:?] OF INTEGER;
END_TYPE;
```

example

```
#65=IFCPOLYARCLINE(#66,(IFCROLEINPATH(.SGMT.),IFCINDEXLIST(1
,2),IFCROLEINPATH(.ARC.),IFCINDEXLIST(2,3,4),IFCROLEINPATH(.
SGMT.),IFCINDEXLIST(4,5,6,7,8,9),IFCROLEINPATH(.ARC.),IFCIND
EXLIST(9,10,11),IFCROLEINPATH(.SGMT.),IFCINDEXLIST(11,12),IF
CROLEINPATH(.ARC.),IFCINDEXLIST(12,13,14),IFCROLEINPATH(.SGM
T.),IFCINDEXLIST(14,15,16,17,18,19),IFCROLEINPATH(.ARC.),IFC
INDEXLIST(19,20,21));
#66= IFCCARTESIANPOINTLIST2D((-0.257,0.006),(0.257,0.006),
(0.2739706,0.0130294),(0.281,0.03),(0.281,0.11),(0.3,0.11),
(0.3,-0.11),(0.281,-0.11),(0.281,-0.03),(0.2739706,
-0.0130294),(0.257,-0.006),(-0.257,-0.006),(-0.2739706,
-0.0130294),(-0.281,-0.03),(-0.281,-0.11)),(-0.3,-0.11),
(-0.3,0.11),(-0.281,0.11),(-0.281,0.03),(-0.2739706,
0.0130294),(-0.257,0.006));

<IfcPolyArcLine>
 <PointRoles>
  <IfcRoleInPath>sgmt</IfcRoleInPath>
  <IfcIndexList>1 2</IfcIndexList>
+ 14 more lines ++++++++++++++++++++++++++++++++++++++++++
 <Points xsi:type="IfcCartsesianPointList2D" CoordList="
-0.257 0.006 0.257 0.006 0.2739706 0.0130294 0.281 0.03
0.281 0.11 0.3 0.11 0.3 -0.11 0.281 -0.11 0.281 -0.03
0.2739706
-0.0130294 0.257 -0.006 -0.257 -0.006 -0.2739706
-0.0130294 -0.281 -0.03 -0.281 -0.11 -0.3 -0.11 -0.3 0.11 -
0.281 0.11 -0.281 0.03 -0.2739706 0.0130294 -0.257 0.006"/>
</IfcPolyArcLine>
```

Proposal 3

list with explicit segments

```
ENTITY IfcSegmentedCurve
 SUBTYPE OF (IfcBoundedCuve);
  Segments : IfcCurveSegment;
END_ENTITY;

ENTITY IfcCurveSegment
SUPERTYPE OF (ONE OF (
  IfcLineSegment, IfcCircularArcSegment))
 SUBTYPE OF (IfcGeometricRepresentationItem);
END_ENTITY;

ENTITY IfcLineSegment
 SUBTYPE OF (IfcSegmentedCurve)
 Points : LIST [2:?] OF IfcCartesianPoint;
END_ENTITY;

ENTITY IfcCircularArcSegment
 SUBTYPE OF (IfcSegmentedCurve)
 Points : LIST [3:3] OF IfcCartesianPoint;
END_ENTITY;
```

example

```
#65=IFCSEGMENTEDCURVE((#66,#67,#68,#69,#70,#71,#72,#73));
#66=IFCLINESEGMENT((#101,#102));
#67=IFCCIRCULARARCSEGMENT((#102,#103,#104));
#68=IFCLINESEGMENT((#104,#105,#106,#107,#108,#109));
#69=IFCCIRCULARARCSEGMENT((#109,#110,#111));
#70=IFCLINESEGMENT((#111,#112));
#71=IFCCIRCULARARCSEGMENT((#112,#113,#114));
#72=IFCLINESEGMENT((#114,#115,#116,#117,#118,#119));
#73=IFCCIRCULARARCSEGMENT((#119,#120,#121));
#101= IFCCARTESIANPOINT((-0.257,0.006));
#102= IFCCARTESIANPOINT((0.257,0.006));
#103= IFCCARTESIANPOINT((0.2739706,0.0130294));
#104= IFCCARTESIANPOINT((0.281,0.03));
#105= IFCCARTESIANPOINT((0.281,0.11));
#106= IFCCARTESIANPOINT((0.3,0.11));
#107= IFCCARTESIANPOINT((0.3,-0.11));
#108= IFCCARTESIANPOINT((0.281,-0.11));
#109= IFCCARTESIANPOINT((0.281,-0.03));
#110= IFCCARTESIANPOINT((0.2739706,-0.0130294));
#111= IFCCARTESIANPOINT((0.257,-0.006));
#112= IFCCARTESIANPOINT((-0.257,-0.006));
#113= IFCCARTESIANPOINT((-0.2739706-0.0130294));

 + 8 more lines ++++++++++++++++++++++++++++++++++++++++
```

Proposal 4

list with indexed segments

```
ENTITY IfcIndexedPolyCurve
 SUBTYPE OF (IfcBoundedCuve);
  Points :        IfcCartesianPointList;
  Segments :      OPTIONAL LIST [1:?] OF
                  IfcSegmentIndexSelect;
  SelfIntersect : OPTIONAL IfcBoolean;
END_ENTITY;


TYPE IfcSegmentIndexSelect = SELECT
 (IfcLineIndex,
  IfcArcIndex)
END_TYPE;

TYPE IfcLineIndex =
 LIST[2:?] OF IfcPositiveInteger;
END_TYPE;

TYPE IfcArcIndex =
 LIST[3:3] OF IfcPositiveInteger;
END_TYPE;
```

example

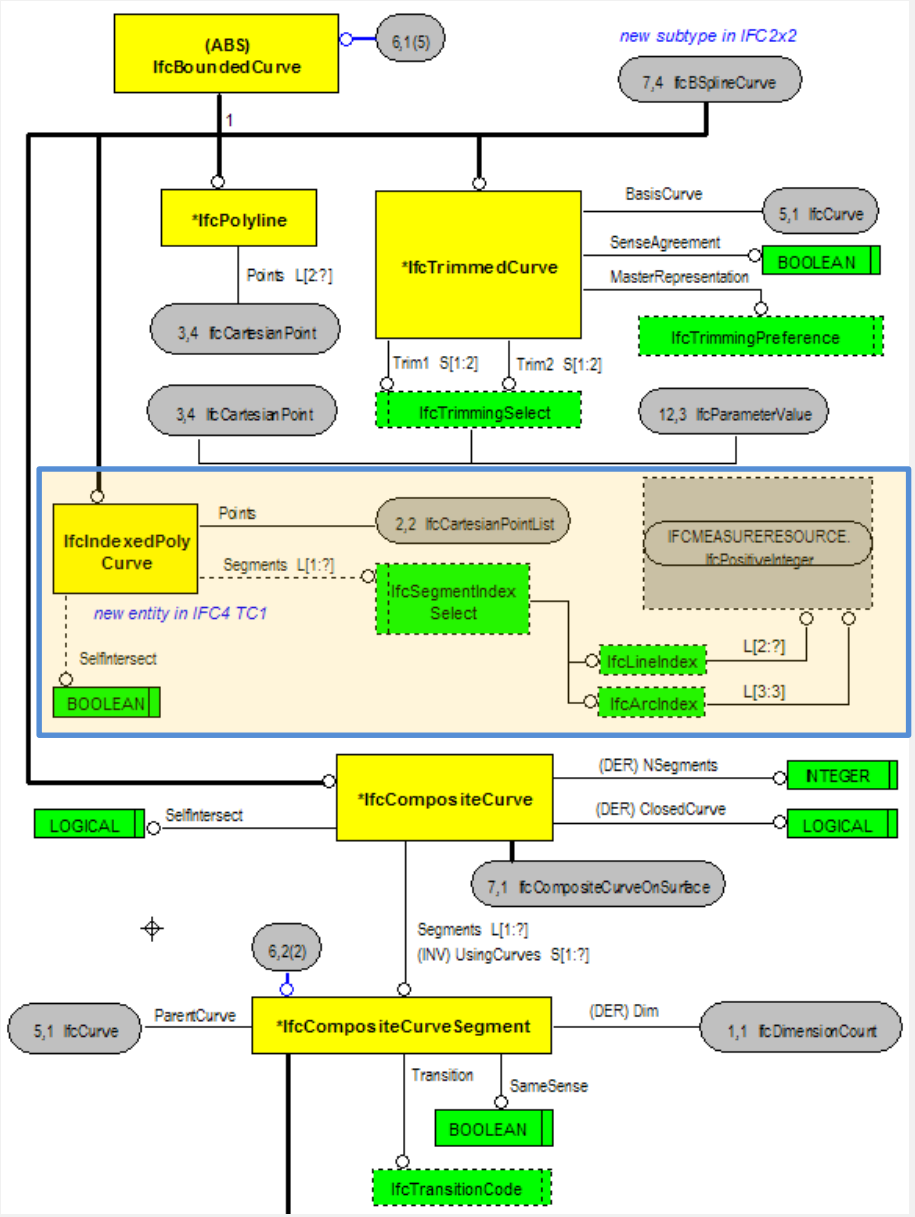
```
#65=IFCINDEXEDPOLYCURVE((#66,(IfcLineIndex((1,2)),IfcArcInde
x((2,3,4)),IfcLineIndex((4,5,6,7,8,9)),IfcArcIndex((9,10,11)
),IfcLineIndex((11,12)),IfcArcIndex((12,13,14)),IfcLineIndex
((14,15,16,17,18,19),IfcArcIndex((19,20,1)), .F.);
#66= IFCCARTESIANPOINTLIST2D((-0.257,0.006),(0.257,0.006),
(0.2739706,0.0130294),(0.281,0.03),(0.281,0.11),(0.3,0.11),
(0.3,-0.11),(0.281,-0.11),(0.281,-0.03),(0.2739706,
-0.0130294),(0.257,-0.006),(-0.257,-0.006),(-0.2739706,
-0.0130294),(-0.281,-0.03),(-0.281,-0.11)),(-0.3,-0.11),
(-0.3,0.11),(-0.281,0.11),(-0.281,0.03),(-0.2739706,
0.0130294),(-0.257,0.006));
```

```
<IfcIndexedPolyCurve>
 <Segments>
  <IfcLineIndex>1 2</IfcLineIndex>
  <IfcArcIndex>2 3 4</IfcArcIndex>
  <IfcLineIndex>4 5 6 7 8 9</IfcLineIndex>
  <IfcArcIndex>9 10 11</IfcArcIndex>
  <IfcLineIndex>11 12</IfcLineIndex>
  <IfcArcIndex>12 13 14</IfcArcIndex>
  <IfcLineIndex>14 15 16 17 18 19</IfcLineIndex>
  <IfcArcIndex>19 20 1</IfcArcIndex>
 </Segments>
 <Points xsi:type="IfcCartsesianPointList2D" CoordList="
-0.257 0.006 0.257 0.006 0.2739706 0.0130294 0.281 0.03
0.281 0.11 0.3 0.11 0.3 -0.11 0.281 -0.11 0.281 -0.03
0.2739706 -0.0130294 0.257 -0.006 -0.257 -0.006 -0.2739706
-0.0130294 -0.281 -0.03 -0.281 -0.11 -0.3 -0.11 -0.3 0.11 -
0.281 0.11 -0.281 0.03 -0.2739706 0.0130294 -0.257 0.006"/>
</IfcIndexedPolyCurve>
```

Proposed IFC extension

IfcIndexedPolyCurve

- new subtype of bounded curve
- along *IfcPolyline* (so it remains unchanged for compatibility)
- uses new *IfcCartesianPointList* (introduced for tessellation) for even smaller files
- list of segments stored with identifier (line | arc) and index into point list
- easily extensible (e.g. to hold control points for quadric or Bezier curves)



Comparison of *IfcCompositeCurve & IfcPolyArcLine*

Using IfcCompositeCurve

```
#63= IFCCOMPOSITECURVE((#64,#68,#75,#78,#81,#84,#87,#90,#97,#100,#107,#110,#113,#116,#119,#122),.U.);
#64= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#67);
#65= IFCCARTESIANPOINT((0.257,0.006));
#66= IFCCARTESIANPOINT((-0.257,0.006));
#67= IFCPOLYLINE((#66,#65));
#68= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#69);
#69= IFCTRIMMEDCURVE(#74,(IFCPARAMETERVALUE(0.0),#65),(IFCPARAMETERVALUE(1.57079632679491),#70),.T.,.PARAMETER.);
#70= IFCCARTESIANPOINT((0.281,0.03));
#71= IFCAXIS2PLACEMENT2D(#72,#73);
#72= IFCCARTESIANPOINT((0.257,0.0300000000000001));
#73= IFCDIRECTION((0.0,-1.0));
#74= IFCCIRCLE(#71,0.024);
#75= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#77);
#76= IFCCARTESIANPOINT((0.281,0.11));
#77= IFCPOLYLINE((#70,#76));
#78= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#80);
#79= IFCCARTESIANPOINT((0.3,0.11));
#80= IFCPOLYLINE((#76,#79));
#81= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#83);
#82= IFCCARTESIANPOINT((0.3,-0.11));
#83= IFCPOLYLINE((#79,#82));
#84= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#86);
#85= IFCCARTESIANPOINT((0.281,-0.11));
#86= IFCPOLYLINE((#82,#85));
#87= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#89);
#88= IFCCARTESIANPOINT((0.281,-0.03));
#89= IFCPOLYLINE((#85,#88));
#90= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#91);
#91= IFCTRIMMEDCURVE(#96,(IFCPARAMETERVALUE(0.0),#88),(IFCPARAMETERVALUE(1.57079632679489),#92),.T.,.PARAMETER.);
#92= IFCCARTESIANPOINT((0.257,-0.00600000000000004));
#93= IFCAXIS2PLACEMENT2D(#94,#95);
#94= IFCCARTESIANPOINT((0.257,-0.0300000000000001));
#95= IFCDIRECTION((1.0,0.0));
#96= IFCCIRCLE(#93,0.024);
#97= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#99);
#98= IFCCARTESIANPOINT((-0.257,-0.006));
#99= IFCPOLYLINE((#92,#98));
#100= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#101);
#101= IFCTRIMMEDCURVE(#106,(IFCPARAMETERVALUE(0.0),#98),(IFCPARAMETERVALUE(1.57079632679491),#102),.T.,.PARAMETER.);
#102= IFCCARTESIANPOINT((-0.281,-0.03));
#103= IFCAXIS2PLACEMENT2D(#104,#105);
#104= IFCCARTESIANPOINT((-0.257,-0.0300000000000001));
#105= IFCDIRECTION((0.0,1.0));
#106= IFCCIRCLE(#103,0.024);
#107= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#109);
#108= IFCCARTESIANPOINT((-0.281,-0.11));
#109= IFCPOLYLINE((#102,#108));
#110= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#112);
#111= IFCCARTESIANPOINT((-0.3,-0.11));
#112= IFCPOLYLINE((#108,#111));
#113= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#115);
#114= IFCCARTESIANPOINT((-0.3,0.11));
#115= IFCPOLYLINE((#111,#114));
#116= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#118);
#117= IFCCARTESIANPOINT((-0.281,0.11));
#118= IFCPOLYLINE((#114,#117));
#119= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#121);
#120= IFCCARTESIANPOINT((-0.281,0.03));
#121= IFCPOLYLINE((#117,#120));
#122= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#123);
#123= IFCTRIMMEDCURVE(#127,(IFCPARAMETERVALUE(0.0),#120),(IFCPARAMETERVALUE(1.57079632679489),#66),.T.,.PARAMETER.);
#124= IFCAXIS2PLACEMENT2D(#125,#126);
#125= IFCCARTESIANPOINT((-0.257,0.0300000000000001));
#126= IFCDIRECTION((-1.0,0.0));
#127= IFCCIRCLE(#124,0.024);
```

Using new proposed IfcIndexedPolyCurve

```
#65=IFCINDEXEDPOLYCURVE((#66,(IfcLineIndex((1,2)),IfcArcIndex((2,3,4)),IfcLineIndex((4,5,6,7,8,9)),IfcArcIndex((9,10,11)),IfcLineIndex((11,12)),IfcArcIndex((12,13,14)),IfcLineIndex((14,15,16,17,18,19),IfcArcIndex((19,20,1)), .F.);
#66= IFCCARTESIANPOINTLIST2D((-0.257,0.006),(0.257,0.006), (0.2739706,0.0130294),(0.281,0.03),(0.281,0.11),(0.3,0.11),
(0.3,-0.11),(0.281,-0.11),(0.281,-0.03),(0.2739706,-0.0130294),(0.257,-0.006),(-0.257,-0.006),(-0.2739706,-0.0130294),(-0.281,-0.03),(-
0.281,-0.11)),(-0.3,-0.11),(-0.3,0.11),(-0.281,0.11),(-0.281,0.03),(-0.2739706, 0.0130294),(-0.257,0.006));
```