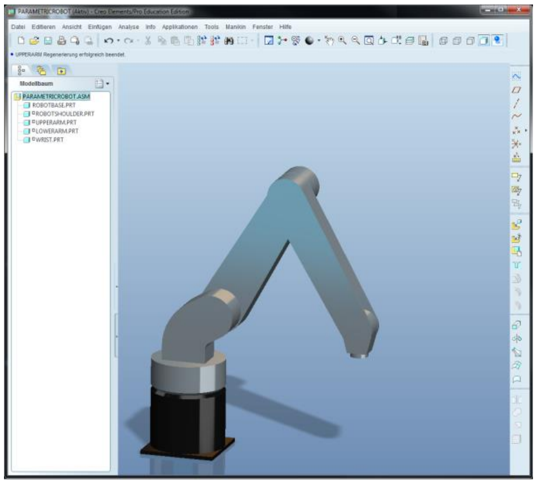
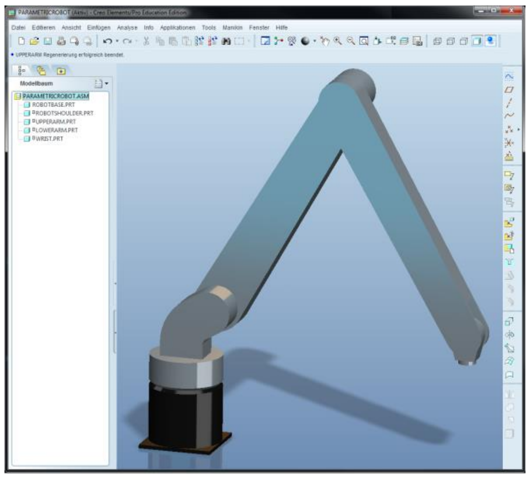
Example of Generic Artifact Handling by our Proposed Approach

First, please do notice that throughout the proposal we used the term "artifact" to convey our general focus and used only the term "code" in the running example of Java in the proposal. We chose Java artifacts purely to explicate our ideas to the general reviewer as it is a well-known programming language and because it is the language used in our paper (see reference [37]) on which the proposal is based.

In the revised proposal we emphasized that the main requisite for the application of our work to a domain such as mechatronics is that the corresponding artifacts can be structurally differentiated. We now illustrate the generality of our proposed approach with an actual example from mechatronics domain. Consider the two designs of robot arms shown below. These snapshots were taken from the tool PTC Creo Elements/Pro, a standard tool in this domain.

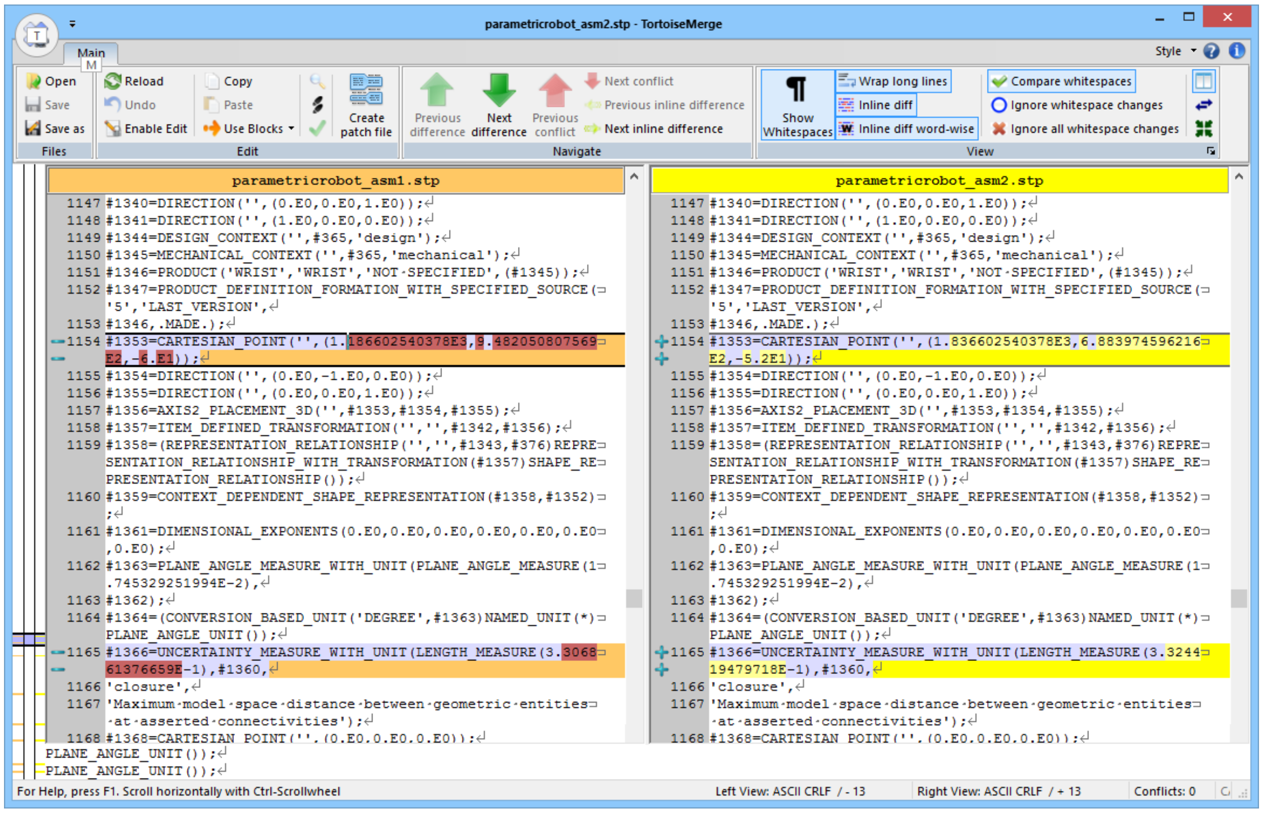

Robot Arm Variant 1


Robot Arm Variant 2

For the sake of argument, let us say that:

- Variant 1 has as identifiable features: `Base` (the black support and its metal part), part `A` and part `B` (the two metal elements that constitute its arm).
- Variant 2 has as identifiable features: `Base` (the black support and its metal part), part `C` and part `D` (the two metal elements that constitute the arm).

This kind of artifact can be serialized into a textual representation that follows an ISO standard (ISO-10303-21) whose structure is defined based on a schema. Even a simple version control tool, as shown in the figure below, can help to detect and visualize basic differences on the textual representation of the two robot arm variants depicted above.



parametricrobot_asm1.stp

```
#1340=DIRECTION('',(0.E0,0.E0,1.E0));
#1341=DIRECTION('',(1.E0,0.E0,0.E0));
#1344=DESIGN_CONTEXT('',#365,'design');
#1345=MECHANICAL_CONTEXT('',#365,'mechanical');
#1346=PRODUCT('WRIST','WRIST','NOT SPECIFIED',(#1345));
#1347=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE(
'5','LAST_VERSION',
#1346,.MADE.);
#1353=CARTESIAN_POINT('',(1.18660254037 8E3,9.482050807569
E2,-6.E1));
#1354=DIRECTION('',(0.E0,-1.E0,0.E0));
#1355=DIRECTION('',(0.E0,0.E0,1.E0));
#1356=AXIS2_PLACEMENT_3D('',#1353,#1354,#1355);
#1357=ITEM_DEFINED_TRANSFORMATION('','',#1342,#1356);
#1358=(REPRESENTATION_RELATIONSHIP('','',#1343,#376)REPRE
SENTATION_RELATIONSHIP_WITH_TRANSFORMATION(#1357)SHAPE_RE
PRESENTATION_RELATIONSHIP());
#1359=CONTEXT_DEPENDENT_SHAPE_REPRESENTATION(#1358,#1352)
;
#1361=DIMENSIONAL_EXPONENTS(0.E0,0.E0,0.E0,0.E0,0.E0,0.E0
,0.E0);
#1363=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(1
.745329251994E-2),
#1362);
#1364=(CONVERSION_BASED_UNIT('DEGREE',#1363)NAMED_UNIT(*)
PLANE_ANGLE_UNIT());
#1366=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(3.3068
61376659E-1),#1360,
'closure',
'Maximum model space distance between geometric entities
 at asserted connectivities');
#1368=CARTESIAN_POINT('',(0.E0,0.E0,0.E0));
```

parametricrobot_asm2.stp

```
#1340=DIRECTION('',(0.E0,0.E0,1.E0));
#1341=DIRECTION('',(1.E0,0.E0,0.E0));
#1344=DESIGN_CONTEXT('',#365,'design');
#1345=MECHANICAL_CONTEXT('',#365,'mechanical');
#1346=PRODUCT('WRIST','WRIST','NOT SPECIFIED',(#1345));
#1347=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE(
'5','LAST_VERSION',
#1346,.MADE.);
#1353=CARTESIAN_POINT('',(1.83660254037 8E3,6.883974596216
E2,-5.2E1));
#1354=DIRECTION('',(0.E0,-1.E0,0.E0));
#1355=DIRECTION('',(0.E0,0.E0,1.E0));
#1356=AXIS2_PLACEMENT_3D('',#1353,#1354,#1355);
#1357=ITEM_DEFINED_TRANSFORMATION('','',#1342,#1356);
#1358=(REPRESENTATION_RELATIONSHIP('','',#1343,#376)REPRE
SENTATION_RELATIONSHIP_WITH_TRANSFORMATION(#1357)SHAPE_RE
PRESENTATION_RELATIONSHIP());
#1359=CONTEXT_DEPENDENT_SHAPE_REPRESENTATION(#1358,#1352)
;
#1361=DIMENSIONAL_EXPONENTS(0.E0,0.E0,0.E0,0.E0,0.E0,0.E0
,0.E0);
#1363=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(1
.745329251994E-2),
#1362);
#1364=(CONVERSION_BASED_UNIT('DEGREE',#1363)NAMED_UNIT(*)
PLANE_ANGLE_UNIT());
#1366=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(3.3244
19479718E-1),#1360,
'closure',
'Maximum model space distance between geometric entities
 at asserted connectivities');
#1368=CARTESIAN_POINT('',(0.E0,0.E0,0.E0));
```

```
PLANE_ANGLE_UNIT());
PLANE_ANGLE_UNIT());
```

For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel | Left View: ASCII CRLF / - 13 | Right View: ASCII CRLF / + 13 | Conflicts: 0

Hence we argue that it is quite feasible to devise a diffing tool that identifies the structural differences between the two variants from their textual representation. By feeding these two variants into our framework, we can readily identify from their textual representation where the common feature Base is realized. With additional variants that provide feature combinations where parts A and B, and parts C and D are not simultaneously provided, our approach can further discern where these features are realized as explicated in Section 1.2 of the proposal. A core challenge would be then to devise as well a domain-specific composition for this type of artifact. For that we plan to rely on the extensive work on software and feature oriented composition as described in the proposal. Again please do notice that other artifacts from this and other domains will be dealt with in *exactly* the same way as just described.