**Final report for work performed**

**BPA NNK100C04Z Subtask 50**

**IT-C1 Development**

**12/20/2013**

I started this task by working on a COLLADA exporter for the CATIA CGR (tessellated) data. The previous work operated on solids data contained in CATPart files. This was meant to complement the existing exporter and support the remaining geometry not already handled. This includes ergo men, foreign tessellated geometry imported into CATIA without STEP, and some piping and cabling.

On the 26th of November, I received an email from Wyck Hebert with some new priorities. The new goal was to avoid the COLLADA issues by delivering models to the TUT importer in the Wavefront OBJ format. Wyck has a custom importer he wrote for this format.

I was about halfway through implementing the CGR COLLADA exporter when I switched over to working on these new priorities.

I decided the first problem, TUT’s inability to import COLLADA, would best be handled by converting the exported COLLADA file to OBJ.

The second problem was that TUT, even if it could import COLLADA, it still would treat the model as a single rigid entity ­­– so having the MPC2 telemetry refer to entities inside the model file wouldn’t work anyway. So really there were two problems to solve.

I decided to solve this second problem by breaking up the single model into smaller models representing rigid subsets of the full model. These smaller models would correspond to the object IDs in the MPC2 telemetry file.

I wrote the COLLADA-to-OBJ converter code in such a way that it could be applied to a portion of a COLLADA document.

I called the command-line utility to achieve all this “tut\_export”. It takes as input a COLLADA file and a set of “.sim” XML files (DVG internal) containing captured motion. It produces a directory containing a set of .mpc files (one for each .sim file), a set of .obj files (one for each MPC2 object ID), their corresponding .mtl files, and all texture image files.

Tut\_export first has to cross-reference all of the entities in each of the .sim files to each other and to the COLLADA file. It then figures out which branches of the scene graph are rigid and runs the OBJ converter on those branches of the COLLADA file. The conversion of the .sim file data to MPC2 is straightforward.

CATIA/DELMIA

NEVA Motion Capture Tool

NEVA COLLADA Exporter

COLLADA Model

(.dae file)

Captured Animation

(.sim XML file)

Captured Animation

(.sim XML file)

Captured Animation

(.sim XML file)

MPC2 File

MPC2 File

MPC2 File

OBJ File / MTL File

OBJ File / MTL File

OBJ File / MTL File

Texture Files

Texture Files

Texture Files

Texture Files

Texture Files

Texture Files

tut\_export

I was able to complete tut\_export on schedule, but I had little time for testing. I did test it to the point that I know it runs and produces the correct output on my small test case. However, it will likely run into problems when applied to a more realistic data set. It just needs some more testing.

I have examined the OBJ files it produces, and both the geometry and texture mapping look good. I have tested the code that calculates the transformations for the MPC output and it works correctly with my test case that has rotations at two levels of the scene graph.

I have fixed all of the bugs that I found during testing. The only remaining issue that I know of is that the OBJ files tut\_export produces are not optimized. There are a lot of redundant material definitions and a lot of repeated vertex information. Tut\_export does not introduce redundancies, but it does propagate them if the source data contains them. It would be fairly easy to write a stand-alone tool to scrub through an OBJ file and remove redundant data.