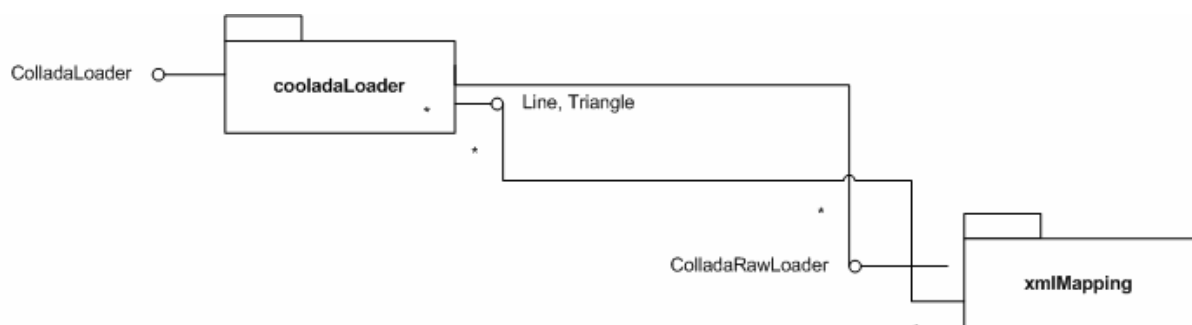


Collada-Loader For Processing Version 1.0

1. Introduction

1.1. System-overview

Google Sketchup is able to export its models into other formats. The freeware version does it into kmz files. Kmz is nothing else but a zipfile that contains an XML file and, if existent, JPEG's for textures. The main goal of the collada-loader is unpack it, parsing the contained XML-File as well as load pictures to PImage objects. Afterwards it maps objects in correct order.



Sketchup exports Triangles and Lines. Thus for user of the Library are 2 Arrays with types of class Triangle and Line available and additional Attributes (e.g. Points) as well. User are able to use the informations to draw into the processing scetch or save to a file. The user can dispose the draw-job to the colladaloader too.

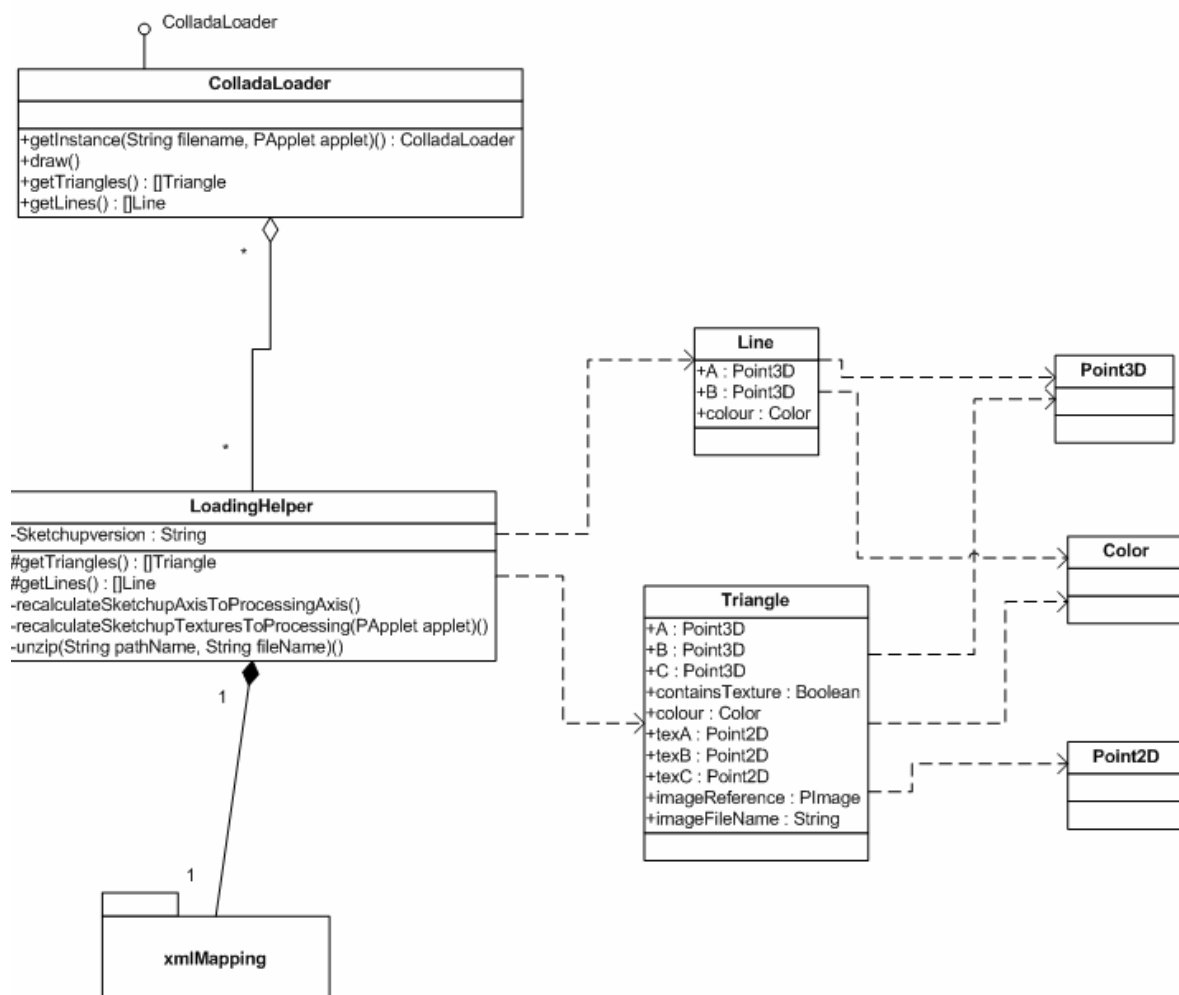
2. Architecture Concept

The choice of Class-Architecture was a bit tricky. The biggest challenge was how to map the xml-file correctly. There exist static dependencies within the xml-tree but also data dependencies between leafs. Thus, the static class hierarchy is based on the xml-Tree. The load sequence, depends on data dependencies, on runtime and is managed by a controller class.

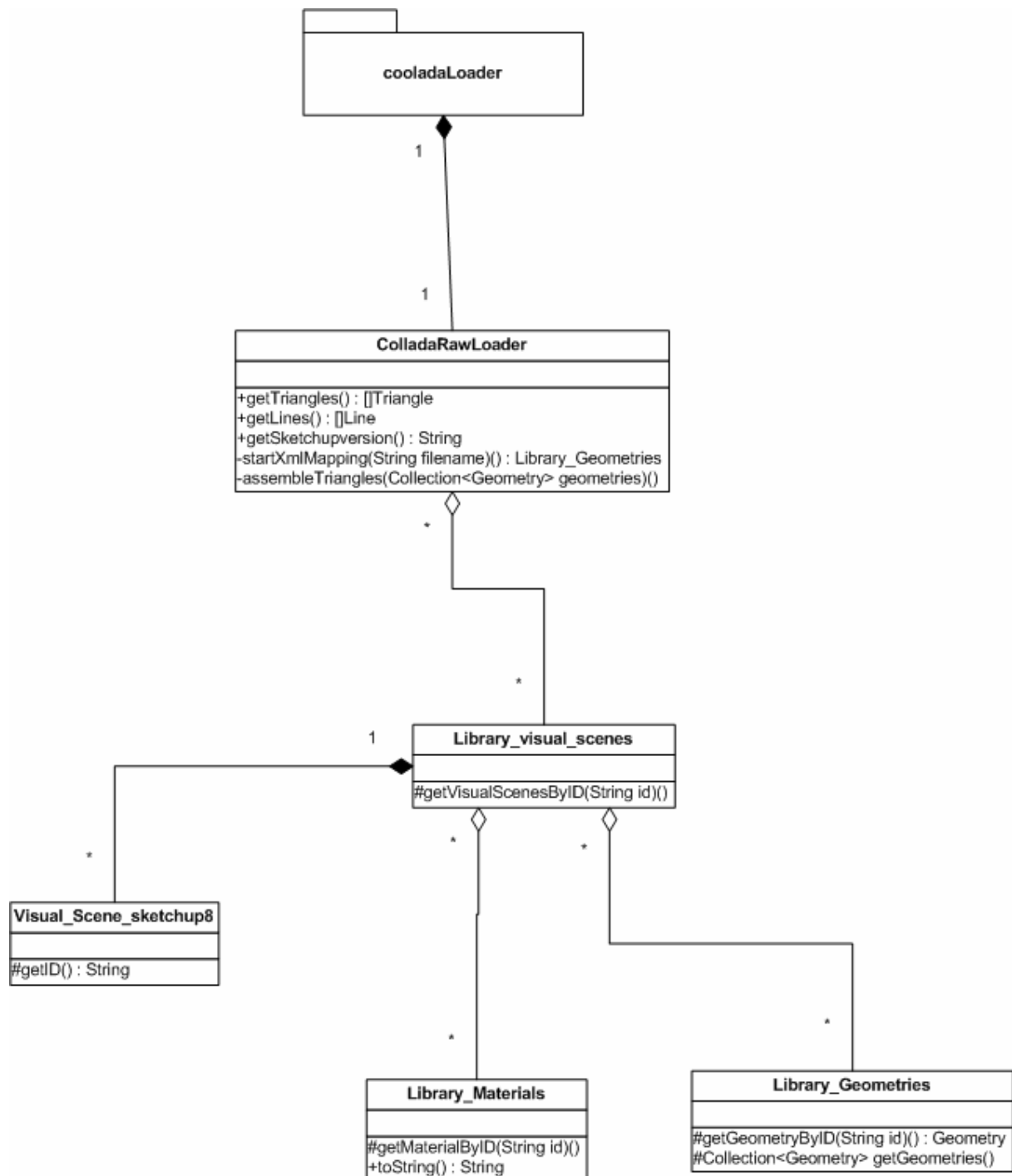
2.1. class-uml

The operating behavior of the classes see next chapter. Some classes are described doubly for more clearly representation

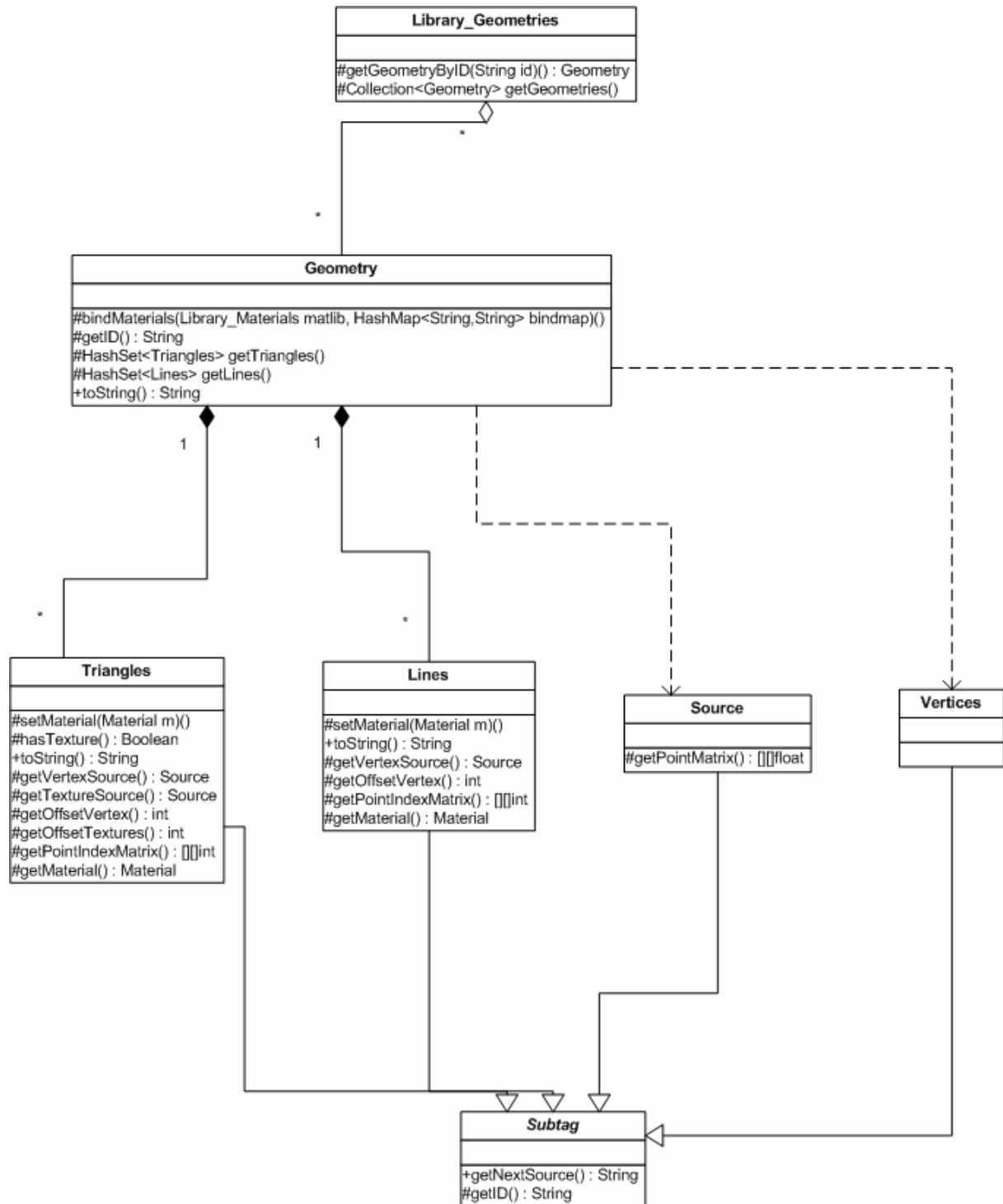
Package CooladaLoader



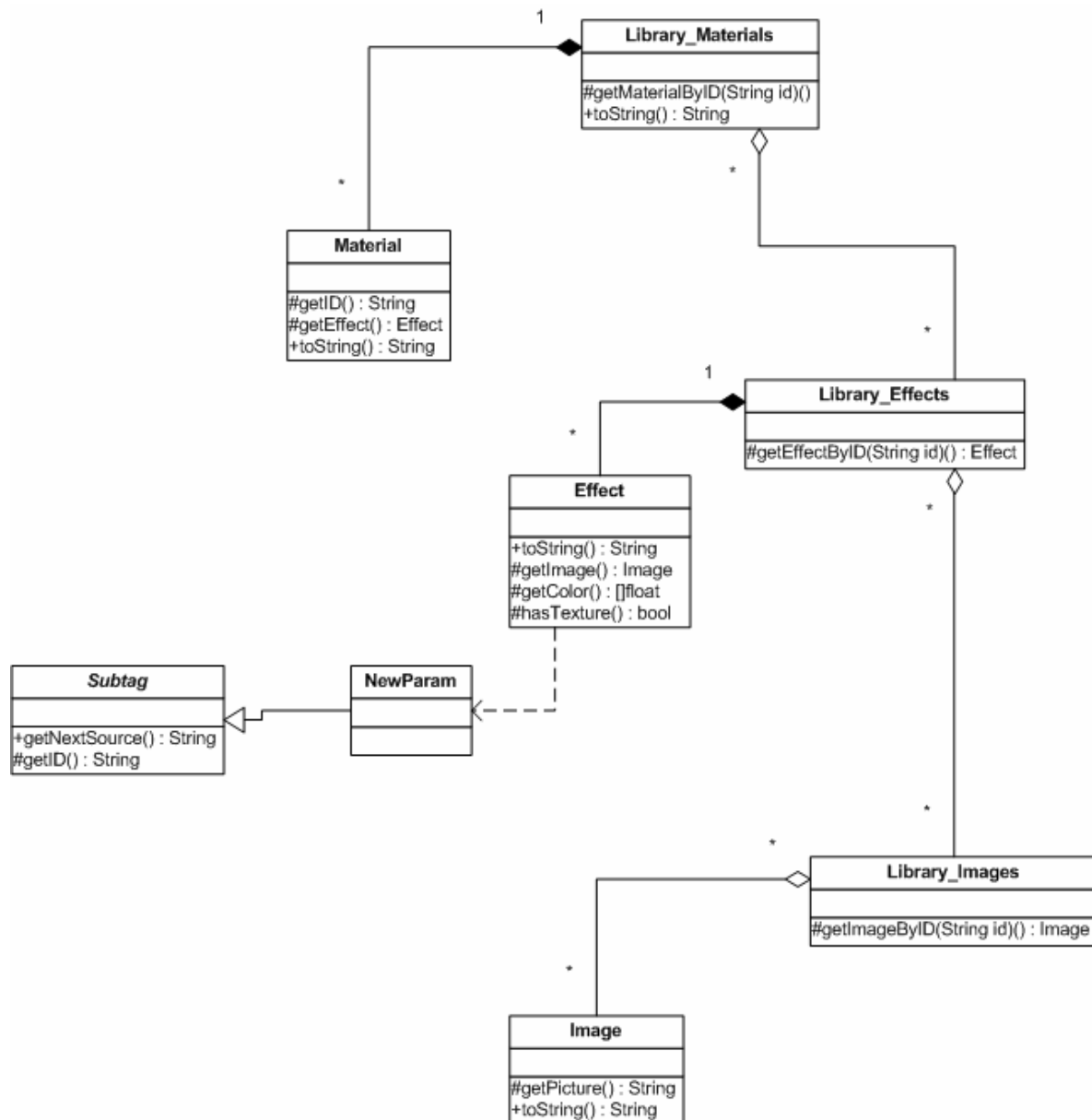
Top-Layer from Package XML-Mapping



Medium layer from Package XML-Mapping



Bottom layer from Package XML-Mapping



2.2. operating view

The operation concept is separated in two packages:

- classes in the Package cooladaLoader do more abstract jobs like unzipping, recalculation of coordinates or loading textures plus User-calls like draw().
- The xml-parsing is done in the package xml-mapping. The class-names are the same as the XML-Tags in the .dae-File. Their constructors analyze the xml-tag autonomous. The ColladaRawLoader is the controller to load the Classes in the right sequence.

(see also Java-doc and comments)

order	Package	class	does
1	coolada	ColladaLoader	Instantiate itself
2	coolada	LoadingHelper	Unzip kmz-File, read doc.kml
3	xmlmapping	ColladaRawLoader	reading .dae-File, loading Library_Images (contains Collection of class Image)
4	xmlmapping	ColladaRawLoader	loading Library_Effects. Has Collection of class Effect. This class has either informations about colors or contains a texture-filename
5	xmlmapping	ColladaRawLoader	loading Library_Materials, contains Collection of class Material. It's doing a Name-Mapping to Effect
6	xmlmapping	ColladaRawLoader	loading Library_Geometries contains Collection of Geometry. The class Geometry may contain Lines oder Triangles. Those contain one or more lines or triangles of the same color or texture again plus a Material-alias for later binding.
7	xmlmapping	ColladaRawLoader	loading Library_Visual_scene_sketchup8. This class binds Materials and Geometries
8	xmlmapping	ColladaRawLoader	Creating the public-Triangle -and Line-wrappers, setting up 3D-Points and texture-points and other attributes to each Triangle/Line
9	coolada	LoadingHelper	Recalculating Sketchup-coordinates to processing readable coordinates (Vertex und textures) loading JPEG-Files and allocation to Triangle-Objects
10	coolada	ColladaLoader	Flushing the Memory. There exist only Triangle and Line-Objekte now avaiable for user usage

3. Interfaces

- cooladaLoader.ColladaLoader (instanciated by user)
- Data-Wrapper-classes (available for user)
 - cooladaLoader.Triangle
 - cooladaLoader.Line
 - cooladaLoader.Point3D
 - cooladaLoader.Point2D
 - cooladaLoader.Color

4. Environment-Requirements

Collada-Loader works:

- dae oder kmz-Files created under Sketchup Version 8
- JavaVM 5 or higher
- Processing Version 1.0.7 or higher

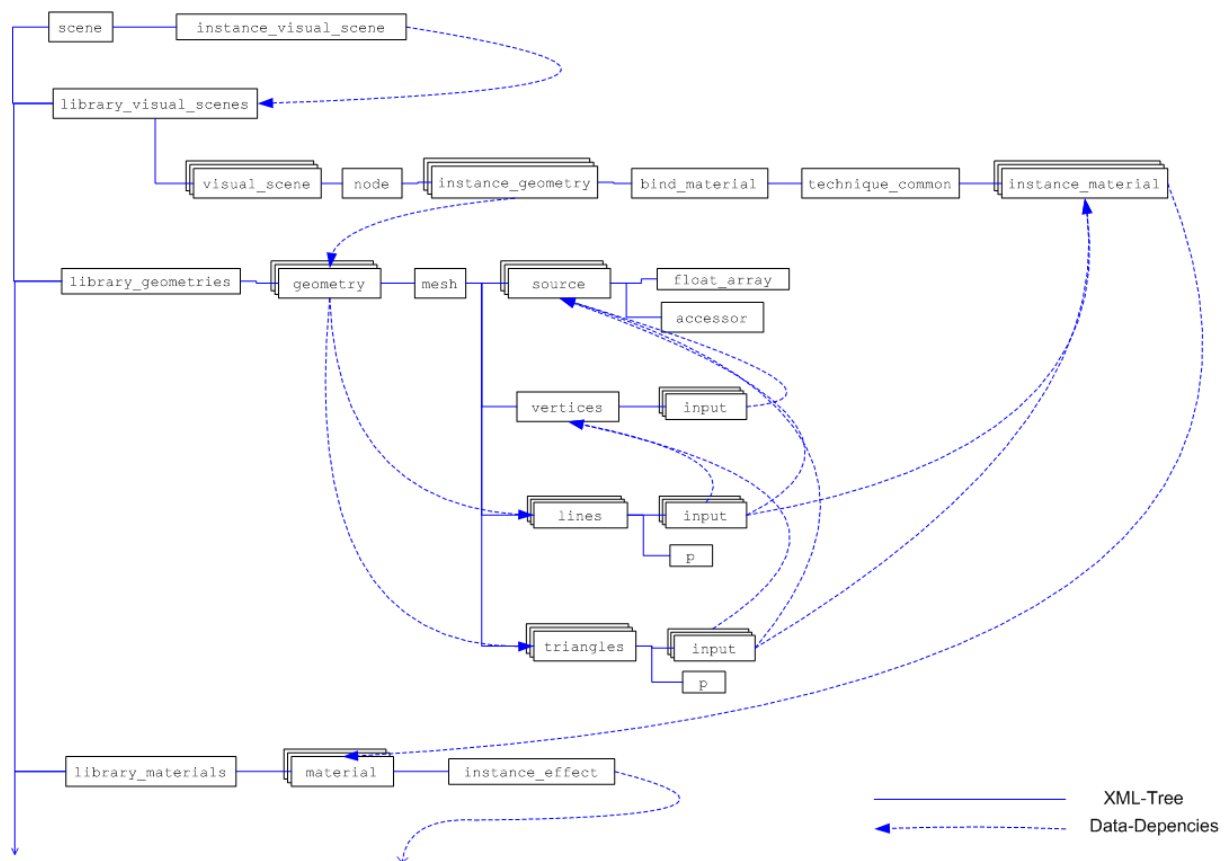
The Collada-Loader does not support:

- groups
- repeated textures

5. Appendix

5.1. XML structure Sketchup 8

The xml-tree and data dependencies is illustrated in next sheet. The Collada-Loader looks up its informations either inside xml-tag attributes or next following Child-Tags. The data dependencies force a correct order in handling tags at runtime. Thus, an <image>-Tag is the first Element and a <visual_scene>-Tag the last one that is loaded by ColladaRawLoader.



In this example the Offset 1 is allocated to textures and the vertices have offset 0. After parsing we get results of:

Vertexes for:

Point A: x=26.2341 y=189.0905 z=0.0

Point B: x=26.2341 y=83.5787 z=0.0

Point C: x=95.1952 y=83.5787 z=0.0

Textures for:

Point A: x=0.9988 y=0.0008

Point B: x=0.0012 y=0.0008

Point C: x=0.0012 y=0.9991

The concept of <lines>-tag construction doesn't have a significant distinction. An <accessor>-Tag says how to read a <float_array>.

5.2. XML-variance of Sketchup 7

This has been analyzed from a file made by scetchup 7. It's not used here in this implementation.

