FROM CAD MODEL TO 3D PRINT VIA "STL" FILE FORMAT

Prof.dr.eng. Cătălin **IANCU**[1], **Eng.** Daniela **IANCU**[2], **Dr.eng.** Alin **STĂNCIOIU**[1]
[1]Univ. "C-tin Brâncuşi" Tg-Jiu, [2]S.C. TREFO Rovinari,
email: ciancu@utgjiu.ro

Abstract: *The paper work presents the STL file format, which is now used for transferring information from CAD software to a 3D printer, for obtaining the solid model in Rapid prototyping and Computer Aided Manufacturing. It's presented also the STL format structure, its history, limitations and further development, as well as its new version to arrive and other similar file formats. As a conclusion, STL files used to transfer data from CAD package to 3D printers has a series of limitations and therefore new formats will replace it soon.*

Keywords: rapid prototyping, layer, STL format.

1. 3D Printing in Rapid Prototyping

3-D Printing is a step in Rapid Prototyping and Manufacturing. Its systems, materials and parts service reduce the time and cost of designing products and facilitate direct and indirect manufacturing by creating actual parts directly from digital input, making possible to deliver the broadest available range of precision plastic and metal parts and finished assemblies. These solutions are used for design communication and prototyping as well as for production of functional end-use parts.

3D printing (also called 3-D Modeling) takes digital input from three-dimensional data and creates solid, three-dimensional parts through an additive, layer-by-layer process. Easy to use, affordably priced and compact for the office, 3-D Modeling is used extensively by designers, engineers and hobbyists for concept development and product design to accelerate the design process and reduce the time to market (TTM).

In recent years 3D printing and 3D printers have become financially accessible to small and medium sized business, thereby taking prototyping out of the heavy industry and into the office environment. It is now also possible to simultaneously deposit different types of materials. While *rapid prototyping* dominates current uses, 3D printers offer tremendous potential for production applications as well. The technology also finds use in the jewellery, footwear, industrial design, architecture, automotive, aerospace, dental and medical industries.

Here are some examples of 3D printed parts, from large used goods to elaborated construction models:

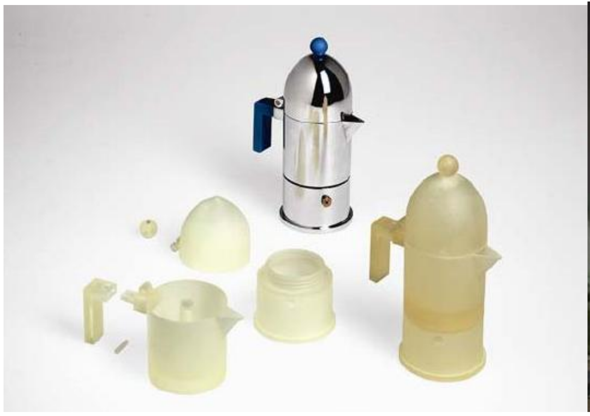


Fig.1 Cuppolla coffeemaker prototype

Fig.2 Scale model 1:80 of Dubai apartment building

2. STL file format

STL is a file format native to the stereolithography CAD software created by **3D Systems Inc.** This file format is supported by many other software packages; it is widely used for rapid prototyping and computer-aided manufacturing. STL native files describe only the surface geometry of a three dimensional object without any representation of color, texture or other common CAD model attributes.

The STL format specifies both ASCII and Binary representations. Binary files are more common, since they are more compact.

An STL file describes a raw unstructured triangulated surface by the unit normal and vertices (ordered by the right-hand rule) of the triangles using a three-dimensional Cartesian coordinate system.

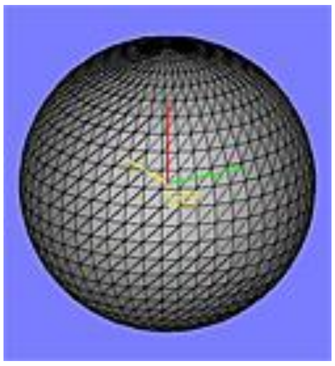


Fig.3 Surfaces are represented by triangles

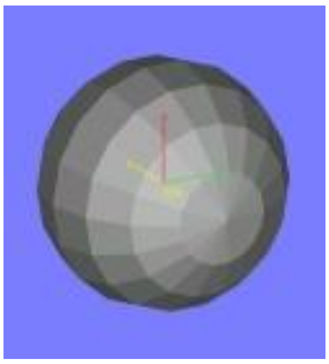


Fig.4 Faceted stl file translated with coarse tolerance

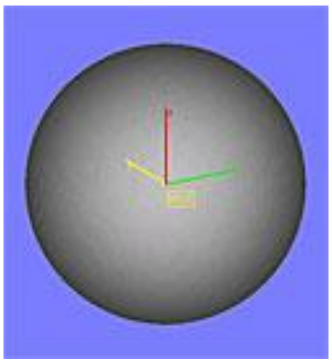


Fig.5 File translated with fine tolerance

2.1. ASCII STL

An ASCII STL file begins with the line:

```
solid name
```

Where *name* is an optional string (note that if the name is omitted there must still be a space after solid). The file continues with any number of triangles, each represented as follows:

```
facet normal ni nj nk
  outer loop
    vertex v1x v1y v1z
    vertex v2x v2y v2z
    vertex v3x v3y v3z'
  endloop
endfacet
```

where n_i-n_k and $v1_x$-$v3_z$ are floating point numbers in sign-mantissa 'e'-sign-exponent format. The file concludes with:

```
endsolid name
```

The structure of the format suggests that other possibilities exist (e.g., Facets with more than one 'loop' or loops with other than three vertices) but in practice, all facets are simple triangles.

White space (spaces, tabs, newlines) may be used anywhere in the file except within numbers or words. The spaces between 'facet' and 'normal' and between 'outer' and 'loop' are required.

2.2. Binary STL

Because ASCII STL files can become very large, a binary version of STL exists. A binary STL file has an 80 character header (which is generally ignored - but which should never begin with 'solid' because that will lead most software to assume that this is an ASCII STL file). Following the header is a 4 byte unsigned integer indicating the number of triangular facets in the file. Following that is data describing each triangle in turn.

The file simply ends after the last triangle.

Each triangle is described by twelve 32-bit-floating point numbers: three for the normal and then three for the X/Y/Z coordinate of each vertex - just as with the ASCII version of STL. After the twelve floats there is a two byte unsigned 'short' integer that is the 'attribute byte count' - in the standard format, this should be zero because most software does not understand anything else.

```
UINT8[80]          -   Header
UINT32             -   Number of triangles

for each triangle
  REAL32[3]        -    Normal vector
  REAL32[3]        -    Vertex 1
  REAL32[3]        -    Vertex 2
  REAL32[3]        -    Vertex 3
  UINT16           -    Attribute byte count
end
```

2.3. Colour in Binary STL

There are at least two variations on the binary STL format for adding colour information:

a. VisCAM/SolidView

The VisCAM and SolidView software packages use the two 'attribute byte count' bytes at the end of every triangle to store a 15 bit RGB colour:

- bit 0 to 4 are the intensity level for blue (0 to 31)
- bits 5 to 9 are the intensity level for green (0 to 31)
- bits 10 to 14 are the intensity level for red (0 to 31)
    - bit 15 is 1 if the colour is valid
    - bit 15 is 0 if the colour is not valid (as with normal STL files)

b. Magics

The Materialise Magics software does things a little differently. It uses the 80 byte header at the top of the file to represent the overall colour of the entire part. If colour is used, then somewhere in the header should be the ASCII string "COLOR=" followed by four bytes representing Red, Green, Blue and Alpha channel (transparency) in the range 0-255. This is the colour of the entire object unless overridden at each facet.

Magics recognize also a material description, more detailed surface characteristic. Just after "COLOR=RGBA" specification should be another ASCII string "MATERIAL=" followed by three colours (3 x 4 bytes): first is a colour of diffuse reflection, second is a colour of specular highlight, and third is an ambient light. Material settings are preferred over colour.

The per-facet colour is represented in the two 'attribute byte count' bytes as follows:

- bit 0 to 4 are the intensity level for red (0 to 31)
- bits 5 to 9 are the intensity level for green (0 to 31)
- bits 10 to 14 are the intensity level for blue (0 to 31)
    - bit 15 is 0 if this facet has its own unique colour
    - bit 15 is 1 if the per-object colour is to be used

2.4. The Facet Normal

In both ASCII and binary versions of STL, the **facet normal** should be a unit vector pointing outwards from the solid object. In most software this may be set to (0,0,0) and the software will automatically calculate a normal based on the order of the triangle vertices using the 'right hand rule'. Some STL loaders (e.g. the STL plugin for Art of Illusion) check that the normal in the file agrees with the normal they calculate using the right hand rule and warn you when it does not. Other software may ignore the facet normal entirely and use only the right hand rule. So in order to be entirely portable one should provide both the facet normal and order the vertices appropriately - even though it is seemingly redundant to do so. Some other software (e.g. SolidWorks) use the normal for shading effects, so the "normals" listed in the file are not the true facets' normals.

Vertex-to-vertex rule. Each triangle must share two vertices with each of its adjacent triangles. In other words, a vertex of one triangle cannot lie on the side of another. This is illustrated in Figure 7.

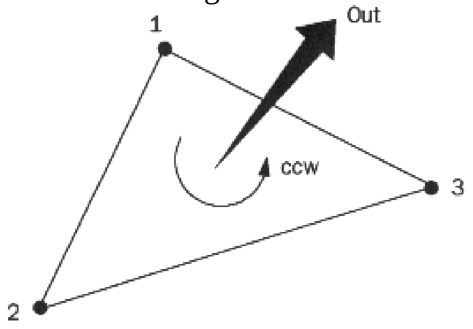


Fig 6. Orientation of a facet is determined by the direction of the unit normal and the order in which the vertices are listed.

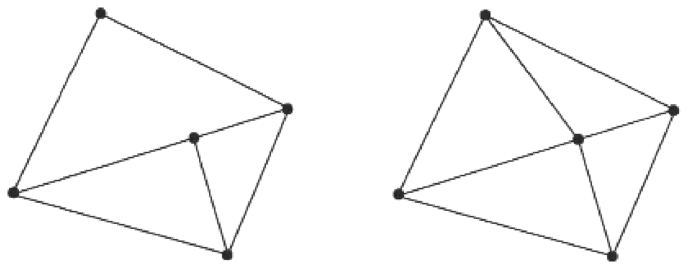


Figure 7. The vertex-to-vertex rule. The left figure shows a violation of the rule. A correct configuration is shown on the right.

3. General characteristics and limitations of .STL file format

Stereolithography machines are basically 3D printers that can build any volume shape as a series of slices. Ultimately these machines require a series of closed 2D contours that are filled in with solidified material as the layers are fused together.

The natural file format for such a machine would be a series of closed polygons corresponding to different Z-values. However, since it's possible to vary the layer thicknesses for a faster though less precise build, it seemed easier to define the model to be built as a closed polyhedron that could be sliced at the necessary horizontal levels.

The STL file format appears capable of defining a polyhedron with any polygonal facet, but in practice it's only ever used for triangles, which means that much of the syntax of the file is superfluous. It is also the case that the value of the normal shouldn't be necessary, since that is a direct calculation from the coordinates of the triangle with the orientation being controlled by the right hand rule. STL files are supposed to be closed and connected like a combinatorial surface, where every triangular edge is part of exactly two triangles, and not self-intersecting. Since the syntax does not enforce this property, it can be ignored for applications where the closeness doesn't matter.

The closeness only matters insofar as the software which slices the triangles requires it to ensure that the resulting 2D polygons are closed. Sometimes such software can be written to clean up small discrepancies by moving endpoints of edges that are close together so that they coincide. The results are not predictable, but it is often sufficient to get the job done.

Obviously, there is much scope for "improvement" of this file format, which in its present form is nothing more than a listing of groups of 9 (or 12 if you care about the normals) floating point numbers embedded in some unnecessary syntax. Since each vertex is on average going to be used in six different triangles, considerable savings in memory could be obtained by listing all the points in a table at the beginning of the file, and concluding with a list of triangle definitions composed of triplets of integers that referenced this table.

However, for the purpose of generating a single contour slice using a very lightweight piece of software on a computer with little memory, this format is perfect since it can be processed in one pass regardless of file size.

4. History of .STL file format

The STL format was developed by the Albert Consulting Group (Albert-Battaglin Consulting Group today) for **3D Systems Inc.** in 1987 for moving 3D CAD models to its stereolithography apparatus (SLA) machines. STL files use linked triangle facets to describe the surface geometry of a three dimensional object. The STL format allows for both ASCII and (smaller) binary representations. STL is an acronym of Standard Tessellation Language.

Over the years, a number of companies and individuals have proposed changing the standard, but to no avail. Instead, system manufacturers have developed their own proprietary standard to optimize the capability of their equipment. **Z Corporation** uses ZPR, a file format that accommodates properties such as color and texture. **Object Geometries Inc.** uses its ObjDF format for its multi-part multi-material Connex 3D printers.

Of course, a whole software and service industry has grown up around repairing and otherwise manipulating STL files. Software packages include the **Materialise** Magics RP suite, **SYCODE**'s OpenRP, Virtual Grid's VRMesh Design, **TransMagic**'s STL PRO, and **Robert McNeel & Associates**' Rhino, among others.

This industry, combined with STL-based software investments made by hardware manufacturers, has created a constituency that is invested in the status quo. The biggest challenge is reaching a consensus and adoption by industry.

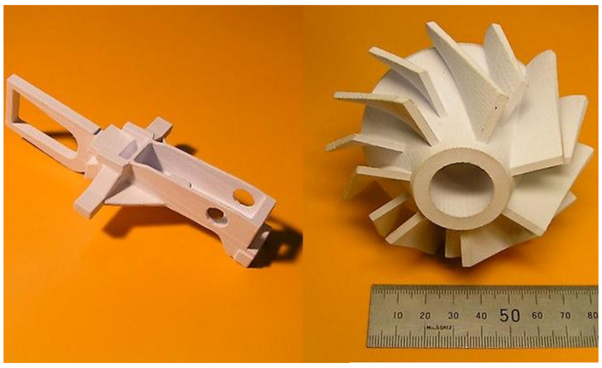
5. Use of .STL in other fields

Many *Computer-aided design* (CAD) systems are able to output the STL file format among their other formats because it's quick and easy to implement, if you ignore the connection criteria of the triangles. Many *Computer-aided manufacturing* (CAM) systems require triangulated models as the basis of their calculation.

Since an STL file output, of a sort, is almost always available from the CAD system, it's often used as a quick method for importing the necessary triangulated geometry into the CAM system. It can also be used for interchanging data between CAD/CAM systems and computational environments such as Mathematica. Once it works, there is very little motivation to change, even though it is far from the most memory and computationally efficient method for transferring this data. Many integrated CAD and CAM systems transfer their geometric data using this accidental file format, because it's impossible to go wrong.

There are many other file formats capable of encoding triangles available, such as VRML, DXF, but they have the disadvantage that it's possible to put things other than triangles into it, and thus produce something ambiguous or unusable.

Here are some examples of 3D Prints (via .STL) from industrial design to architecture, automotive, dental and medical industries:



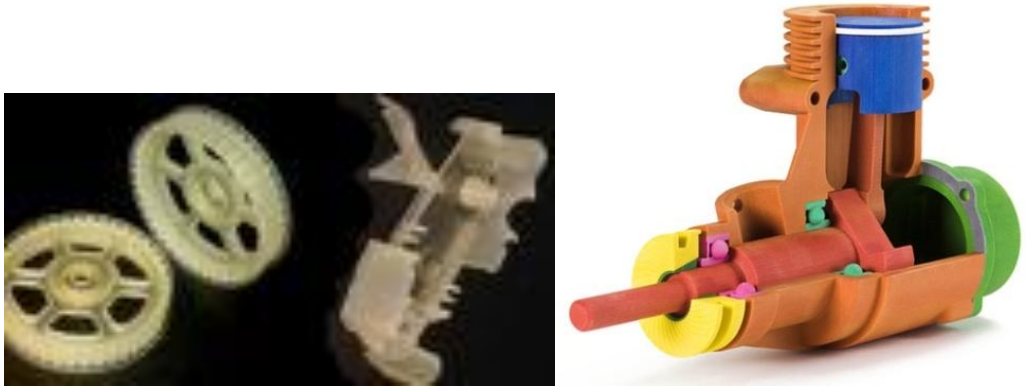


Fig 8. Examples of use of .STL in industrial 3D Prints

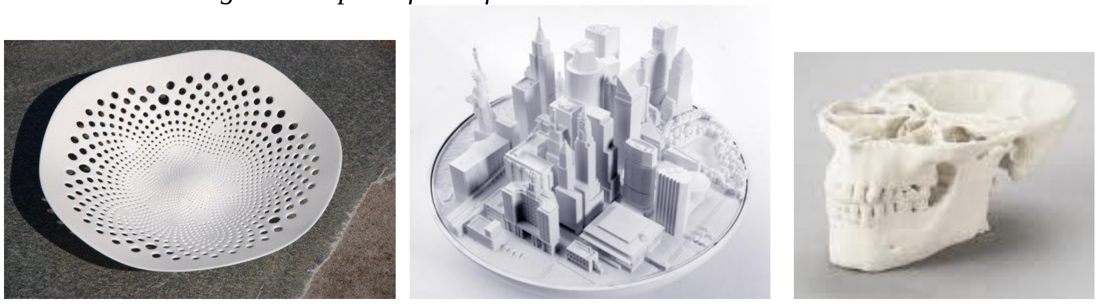


Fig 9. Examples of use of .STL in architecture and medical 3D Prints

6. STL 2.0 may replace old, limited file format

Since soon after it was developed, people in the rapid prototyping field have talked about replacing the industry-standard STL file format. Yet it has endured for 22 years. Still, the pressure for change builds. And now there is a mechanism that may allow it to happen.

The **ASTM (**American Society for Testing and Materials) **Committee on Additive Manufacturing Standards** (F42) had its first meeting in July 2009, and agreed to form a task group to recommend a new file format standard, called STL 2.0.

"The time is right. It's time for a change," says Hod Lipson, chair of the ASTM task group on file formats. In Lipson's day job as associate professor at Cornell University he has seen first-hand the limitations of STL. With a couple graduate students, Lipson developed the remarkable low-cost open-source Fab@Home fabbing machine. He also runs the school's Computational Synthesis Laboratory. As Lipson tells it, STL endured so long for one reason: it's not over-complicated. "The beauty of it is it's very very simple," he says. "You can code it up very quickly and you can read or write to it very simply. For years, simplicity outweighed any other drawbacks, but the balance is beginning to change."

Now, with the advancing capability of additive fabrication systems, simple isn't sufficient anymore, and the problem is growing. There are new technological capabilities that the format doesn't address.

There are also "obvious" new needs (multiple materials, color, microstructure requirements) as well as "more subtle" ones:

- Better accuracy in representing curved surfaces
- Surface texturing requirements
- Buildability verification (leak patching)
- Inclusion of metadata (e.g. authorship and copyright information)

The STL file format doesn't just lack these capabilities, it also has a few inherent problems: file size is excessive, file security is limited, and it can't detect or fix errors (especially unintended holes) in the part to be built.

To help generate consensus, the Work Committee is soliciting input via an online survey and discussion group, planning to keep the survey up for a couple months.

Following a quick review of the results so far, it has been noticed a top concern for the new format: It should be a non-proprietary open format. "Making it open format will make it future-proof. It will be something that can grow and can be backwards and forwards compatible". "Keeping it simple is the key."

The results of the survey were presented in November 9-10, 2009 at the Committee on Additive Manufacturing Standards meeting at ASTM International headquarters in West Conshohocken, PA. From there, was made a draft out by March, 2010, followed by two months of discussion, and a finalization by mid-2010.

7. What is OpenRP and RP file format

The STL (Stereolithography) file format is, and for the foreseeable future, will be the standard mode of data exchange in the Rapid Prototyping industry. It is a known fact that the STL file format does not employ efficient methods of data storage and lacks security. Over the years, attempts have been made to replace the STL file format with a better, efficient and secure alternative. Most of these attempts have been made by major RP software developers who, in one way or the other, have tried to derive financial benefit in the bargain.

OpenRP is not another attempt to replace the STL file format. OpenRP is a non-profit initiative by **SYCODE Inc.** which aims at offering the RP industry an alternative mode of data exchange while maintaining perfect compatibility with the STL file format and without any loss of data whatsoever. OpenRP offers the RP industry a new RP file format and free software to read and write RP files. All software offered by SYCODE is and will always remain free. None of the applications, plug-ins or libraries will ever expire. You can freely use the software for as long as you like.

The RP file format is an efficient and secure file format created by the OpenRP initiative which is perfectly compatible with the STL file format.

Data is compressed using state of the art compression algorithms thereby reducing the file size by 97% (when compared to an ASCII STL file) or 90% (when compared to a Binary STL file), which is much more than the compression achieved by WinZip or similar compression programs. The RP file format has two levels of security: file and user level. At the file level, the RP file is encrypted using the most advanced encryption algorithms. At the user level, security is provided by means of an optional user-defined password. There is absolutely no loss of data due to compression and encryption.

The RP market itself will validate the RP file format or the new .STL 2.0 format.

8. Conclusions

In recent years 3D printing and 3D printers have become financially accessible to small and medium sized business, thereby taking prototyping out of the heavy industry and into the office environment. The technology of *rapid prototyping* also finds use from industrial design to dental and medical industries.

STL files used to transfer data from CAD package to 3D printers have a series of limitations and therefore new formats will replace it soon.

References

[1]. Grenda, E., *The Most Important Commercial Rapid Prototyping Technologies at a Glance*, Prentice-Hall Inc., N.Y., 2006.

[2]. Chee Kai Chua; Kah Fai Leong, Chu Sing Lim, *Rapid Prototyping*, World Scientific, pp. 124., 2008

[3]. Wright, Paul K., *21st Century manufacturing*. New Jersey, Prentice-Hall Inc., 2001.

[4]. ******* StereoLithography Interface Specification, 3D Systems, Inc., October 1989

Webography

[5]. http://www.3dsystems.com/products/3dprinting/overview.asp

[6]. http://www.bastech.com/sla/techTips/STLfiles.asp

[7]. http://www.fabbers.com/stl.asp