# Virtual and rapid prototyping to develop spatial visualization skills of students in engineering

David Wattiaux[1], Olivier Verlinden[2], Edouard Rivière-Lorphèvre[3] Christine Renotte[1]
david.wattiaux@umons.ac.be

[1]Faculty of Engineering, Pedagogical support unit
University of Mons – UMONS Place du Parc, 20, 7000 Mons, Belgium

[2]Faculty of Engineering, Theoretical Mechanics, Dynamics and Vibrations Unit
University of Mons – UMONS Place du Parc, 20, 7000 Mons, Belgium

[3]Faculty of Engineering, Machine Design and Production Engineering Unit
University of Mons – UMONS Place du Parc, 20, 7000 Mons, Belgium

**keywords** : Mechanism modelling, Rapid prototyping, VRML animation, Multibody simulation, Engineering education

## 1 INTRODUCTION

### 1.1 Learning difficulties

Nowadays, with the improvement of both computer hardware and software, the 3D solid modelling CAD has become ubiquitous in industrial applications. The future engineer should be able to read, to interpret and to create an industrial technical drawing of an object or a mechanism (assembly of parts). Several studies, including one conducted by [1], highlight that the spatial visualization ability is an important component of the problem solving, particularly in technical problem solving and design. Therefore, the spatial visualization has become a required skill for engineering students.

At the University of Mons, the first year engineering students mainly develop spatial skills within the course of geometry. The teaching of geometry includes an important part dedicated to descriptive geometry. The descriptive geometry is a graphical communication system allowing the representation of three-dimensional objects in two dimensions by using perspective or parallel projections. In the framework of our training courses, we mainly focus on the orthogonal projections (the projection line is orthogonal to the projection plane) in particular the isometric projection and the Monge's method (Multiview orthographic projection used in standard industrial drawings) which are the most commonly used by engineers. The students start with the paper drawing of Multiview or isometric projections of solids, and especially polyhedra. Afterwards, they use a solid modelling software (in our case `SolidWorks` software) to build 3D solid models and automatically generate the technical drawings with dimensions.

Sight in space is a skill also encountered within the course of Theoretical Mechanics. Theoretical Mechanics is taught within Bachelor of Science in Engineering and it deals

with Statics, Kinematics and Dynamics of mechanical systems. A mechanical system is a set of rigid bodies connected to each other by force elements (springs, dampers, actuators,. . . ) or kinematic joints restraining their relative motion. In kinematics, the students have to be able to predict from a schematic diagram the motion of the bodies in the space.

In both Geometry and Theoretical Mechanics courses, the students often present some weaknesses to visualize in space. This assessment is not peculiar to our engineering faculty. Several pedagogic and scientific researches reveal that the European engineering students face various difficulties in getting acquainted with of spatial visualization [2, 3]. These difficulties are mainly due to the inexistence of a systematic process to analyze complex problems [4]. Commonly, students use the trial-and-error strategy or they simply rely on their physical intuition to solve problems.

### 1.2 Pedagogical project

The University of Mons has a pedagogical support unit whose one of the main objectives is the implementation of special educational supports intended to enhance our education. From discussions between the concerned teachers and the pedagogical support unit, the idea emerged to develop numerical tools, using available technologies such as virtual or rapid prototyping model, in order to improve the spatial sight of our students.

In order to establish a relationship between Geometry and Theoretical Mechanics course, we mainly focus on the study of mechanisms. Indeed, the 3D modelling and the technical drawing of the mechanisms are tackled during the laboratory works of the geometry course. On the other hand, the Kinematics aspects are discussed in the framework of Theoretical Mechanics course.

About three years ago, we have started the implementation of a Mechanical Virtual Laboratory (`MVL`) [5, 6]. Our `MVL` consisted of a dynamic website collecting `VRML` (Virtual Reality Modeling Language) animations of mechanisms whose only some geometrical parameters could be modified by the user. Recently, the `MVL` was enriched with the possibility to introduce Dynamics aspects by allowing the user to change some physical parameters (such as the inertia properties or the stiffness of springs) and to display interesting curves such as the time history of the velocity/acceleration.

Many teachers recognize the educational benefits of physical models for hands-on learning in Mechanics but also in Chemistry (ball-and-stick models of complex molecules). The physical models play an important role in the active learning and they may alleviate some learning disabilities [7]. That is why, alongside the development of our `MVL`, we recently started the execution of 3D-printable physical models of mechanical parts and mechanisms. The use of rapid prototyping technology offers new opportunities to enhance accessibility to physical teaching models and customize them for specific personal learning needs.

This paper presents an overview of the numerical tools that we have used or devel-

oped to build up our Mechanical Virtual Laboratory and our recent initiative to establish a repository of 3D-printable models for education. The ellipsograph mechanism will be the main support for illustration.

## 2 REFERENCE MECHANISM: ELLIPSOGRAPH

### 2.1 Description of the mechanism

The ellipsograph is a mechanical device able to draw ellipses. It is composed of two sliders moving inside two perpendicular rails. The two sliders are connected through a rigid rod using pivots (Figure 1). A pencil attached to the end of the bar allows to trace the ellipse. By adjusting the distance between the centers of the sliders, ellipses of various eccentricities can be generated.

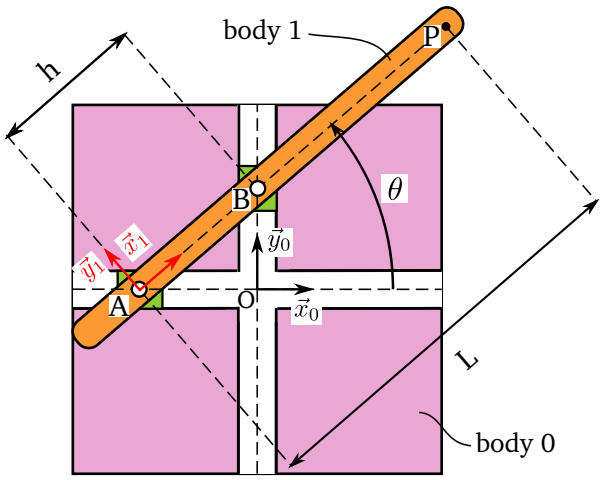



*Figure 1: Diagram of the ellipsograph*

### 2.2 Formalism of the homogeneous transformation matrix

The analysis of a multibody system (mechanism) requires a mathematical tool in order to represent the spatial situation (position and orientation) of the bodies constituting the system. The most commonly used framework is the formalism of the homogeneous transformation matrix. According to this formalism, the spatial situation of the local frame $i$ attached to body $i$ can be described with respect to the global reference frame from a 4 $\times$ 4 matrix partitioned in the following way (cf. Figure 2(1) ) :

$$\mathbf{T}_{0,i} = \begin{pmatrix} \mathbf{R}_{0,i} & \{\vec{\mathbf{e}}_i\}_0 \\ 0\,0\,0 & 1 \end{pmatrix} \tag{1}$$

where $\vec{\mathbf{e}}_i$ is the coordinate vector of the origin of frame $i$ with respect to the origin of the global coordinate system $0$, and $\mathbf{R}_{0,i}$ is the rotation tensor describing the orientation of frame $i$ with respect to frame $0$. The notation $\{\vec{\mathbf{e}}_i\}_0$ represents the 3x1 vector gathering

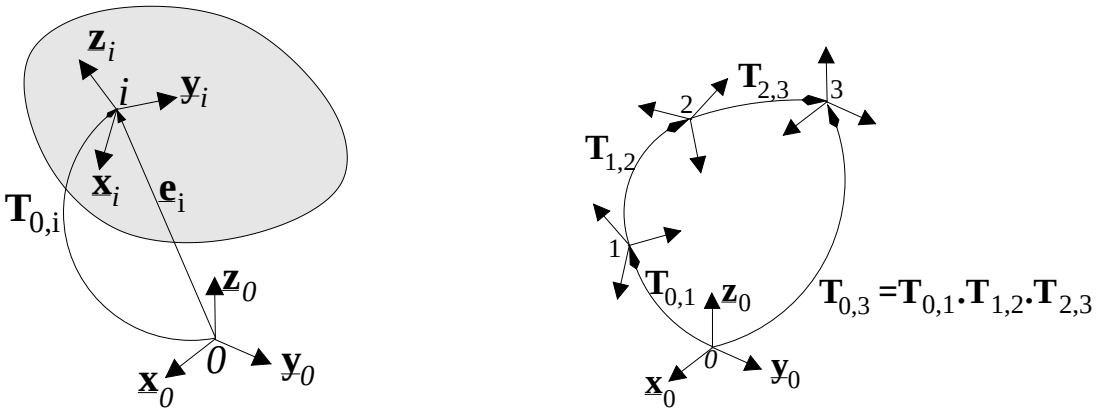
the 3 components of the vector $\vec{\mathbf{e}}_i$ in coordinate system 0.

The main advantage of this formalism is to enjoy the following property:

$$\mathbf{T}_{i,k} = \mathbf{T}_{i,j} \cdot \mathbf{T}_{j,k} \quad \forall i, j, k \ , \tag{2}$$

This means that a complex motion can be defined as a succession of elementary motions (translations, rotations, ...) as illustrated in Figure 2(2) . For any complex mechanical system, the motion of each body can be decomposed into a succession of elementary motions as the rotation about a given axis or the displacement along a straight line. Associated to the property defined by Eq. 2, the homogeneous matrix of each body with respect to the reference frame $0$ can be easily expressed in the following way :

$$\mathbf{T}_{0,i} = \mathbf{T}^{\text{elementary motion 1}} \cdot \mathbf{T}^{\text{elementary motion 2}} \cdot \mathbf{T}^{\text{elementary motion 3}} \ldots \tag{3}$$




(1) local and global frames (2) Composition of motion

*Figure 2: Formalism of homogeneous transformation matrix*

### 2.3 Equations of the ellipse

If $\theta$ represents the angle between the rod and the $x_0$ axis, the coordinates of the point P can easily be expressed in the global reference frame using the formalism of the homogeneous transformation matrix :

$$\begin{pmatrix} \left\{\overrightarrow{\mathrm{OP}}\right\}_0 \\ 1 \end{pmatrix} = \mathbf{T}_{0,1} \begin{pmatrix} \left\{\overrightarrow{\mathrm{AP}}\right\}_1 \\ 1 \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & -h\cos\theta \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} L \\ 0 \\ 0 \\ 1 \end{Bmatrix} \rightarrow \begin{cases} x = (L-h)\cos\theta \\ y = L\sin\theta \\ z = 0 \end{cases} \tag{5}$$

The equations (5) correspond to the parametric equations of an ellipse in the canonical position.

# 3 SIMULATION TOOL : EASYDYN FRAMEWORK

## 3.1 General presentation

About ten years ago, the department of Theoretical Mechanics, Dynamics and Vibration of the University of Mons (UMons – Belgium) started the development of the `EasyDyn` project. `EasyDyn` is an open source numerical tool devoted to the simulation of mechanical and mechatronic systems. The `EasyDyn` framework is an open source numerical tool dedicated to the simulation of mechanical and mechatronic systems. `EasyDyn` permits the numerical construction and integration of the equations of motion from the kinematics, expressed through the Homogeneous Transformation Matrices, and the forces exerted on the mechanical system.

`EasyDyn` was initially built for educational purposes in the framework of the course of *Computer-Aided Analysis of Multibody Systems* taught to students within the Master of Mechanical Engineering. Nowadays, `EasyDyn` is also widely used for research purposes with industrial applications.

Practically, it consists of two main parts: a C++ library and a symbolic tool called `CAGeM` (Computer Aided Generation of Motion). The Components of `EasyDyn` will be described very briefly in this paper. The interested reader can find more information about `EasyDyn` framework in [8] and sample applications in [9]. A complete package can be freely downloaded from our web site (`http://mecara.fpms.ac.be`). It comprises the `EasyDyn` C++ library, the `CAGeM` script, some examples and a series of problems.

## 3.2 Components of EasyDyn

### 3.2.1 The C++ library

The `EasyDyn` C++ library consists of 4 modules : `vec`, `sim`, `mbs` and `visu`.

The `vec` module defines classes related to vector calculus (vectors, rotation tensors, inertia tensors, and homogeneous transformation matrices). Each class encapsulates a set of parameters, operators, and some utility functions which allow to write vector expressions in the C++ code.

The `sim` module provides routines to numerically integrate the second-order differential equations of the motion. So far, only the `Newmark` method is implemented.

The `mbs` module is just a frontend to `sim` which builds the equations of motion of a mechanical system from the Kinematics, expressed in terms of homogeneous transformation matrices, and the applied forces. The module also offers several routines implementing usual force elements like springs, dampers, tyres or compliant contacts.

Finally, the `visu` module allows to define a graphical scene composed of moving shapes. Basically, a shape consists of a set of nodes and a set of edges and surfaces defined between nodes. The initial coordinates of the nodes, indices of end nodes of edges,

and number of vertices and node indices of polygonal surfaces, are saved to a `*.vol` file. The coordinates of nodes for the successive configurations of the scene are saved in a *.van file. The `vol` and `van` files completely define the animation which can be displayed by an independent viewer called `EasyAnim` also freely available from our web site.

In practice, each mechanical system is defined in a specific C++ program in which the kinematics and applied forces are described. The equations of motion are then built and integrated numerically with the help of particular routines of the `EasyDyn` library. The `EasyDyn` application generates the `*.vol` and `*.van` animations files (Figure 3).

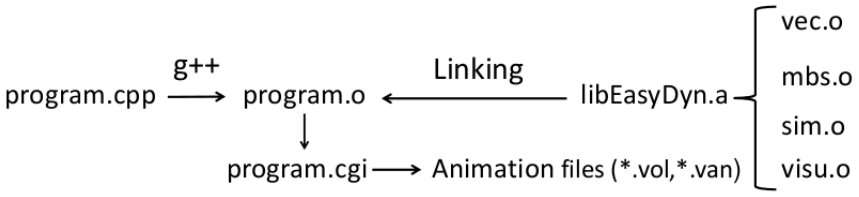



*Figure 3: EasyDyn library*

### 3.2.2 CAGeM : the symbolic tool

`CAGeM` (Computer Aided Generation of Motion) is a symbolic script which generates a core C++ program directly compilable against the `EasyDyn` library from a basic set of informations. Among the latter, the user specifies the number of bodies, the number of degrees of freedom, the inertia characteristics (mass and inertia tensor) but also the definition of the homogeneous transformation matrices associated to each body in terms of configuration parameters. The velocities and the accelerations of all bodies are computed from the homogeneous transformation matrices using the symbolic derivation features of `MuPad/Xcas` [10].

## 4 THE MECHANICAL VIRTUAL LABORATORY

### 4.1 General presentation

The Mechanical Virtual Laboratory (`MVL`) is a dynamic `Web` site offering some interactive animations of mechanisms written in a `VRML` language. Furthermore, specific animated curves, such as the path followed by a point or the time-history of the velocity, can be directly displayed in a `Web` browser.

Figure 4 shows a screenshot of our `MVL` with the example of the ellipsograph mechanism. The user can directly modify from a `HTML` form the distance between the center of the sliders and the length of the connecting rod. The data keyed in the `HTML` form by the user are sent by Common Gateway Interface to the `EasyDyn` program which simulates the mechanism and updates the `VRML` animation. The animated plot of the ellipse is displayed and is synchronized with the `VRML` animation. The animated curves are generated by the

open source Gnuplot software [11].

Our MVL is available from the following URL: http://mrvirtuallab.umons.ac.be/.

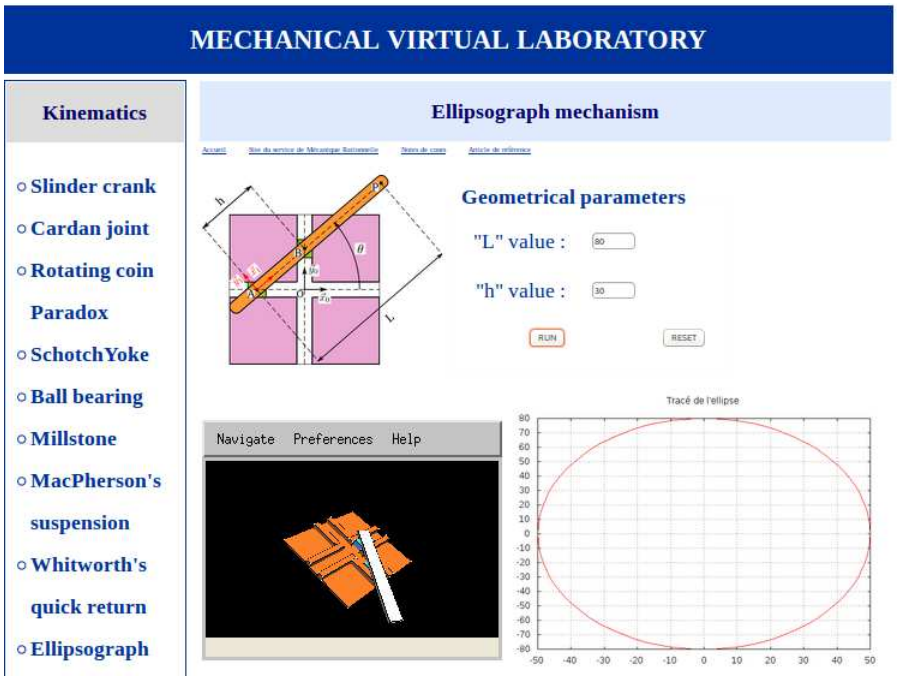



*Figure 4: Interface of the Mechanical Virtual Laboratory*

## 4.2 VRML animation

The structure of the vol and van animation files generated by EasyDyn doesn't allow to display the scene directly in a Web browser. Therefore, it is necessary to convert them to an appropriate format. There are several computing techniques for creating animations on the Web site. In the framework of our MVL project, we have chosen to work with Virtual Reality Modeling Language (VRML) because it is a widely used format for describing 3D virtual world in a network environment, accepted by the International Organization for Standardization (ISO) [12]. Furthermore, with VRML, the user can directly interact with the scene (an object can be rotated or moved in relation to the viewpoint). Therefore, we have written a C++ routine converting the .vol and .van animation files of EasyDyn to a VRML file. The VRML file can be interpreted by several plugins available freely from Internet like CORTONA 3D (for Windows platforms) or freeWRL (for Unix platforms).

The VRML file uses the .wrl file extension and is a plain text file in which the animation of the scene is defined in the same way as in EasyDyn by a set of vertices, edges and polygons. Table 1 shows a simple example of VRML file generating the animation of a scene.

The VRML scene is defined by *nodes* which describe the objects and their properties. In VRML, there are many kinds of *nodes*. For instance, we may cite the geometric node

which defines the geometric objects and the sensor node and the Interpolator node which is used to create the VRML animations. VRML has a variety of interpolator nodes for the creation of keyframe animations. The interpolator node defines a mapping from a key, ranging from 0.0 to 1.0, to a keyValue which can take several types of data depending on the Interpolator. In the case of the CoordinateInterpolator node, the keyValue field corresponds to a list of 3D coordinate values which provides a way to translate the shape along a predefined path. Each interpolator node has an input event named set_changed and an output event named value_changed. Commonly, the Interpolator node is combined with a sensor node that generates events, called fraction_changed event, at each clock tick. These events can be linked to the set_changed event from the interpolator using a ROUTE which carries out the connection between the event generated by the *node* and the *node* receiving the event.

The reader who wishes to go into the matter can find an overview in [13, 14, 15, 16]. Moreover, an interactive tutorial is available on the following web site : *http://www.lighthouse3d.com/vrml/tutorial/*.

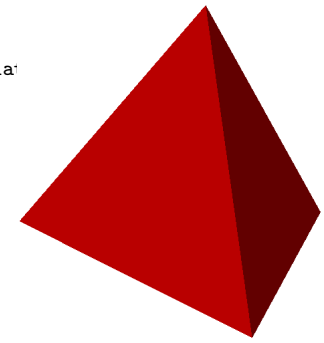
*Table 1: Example of a basic VRML file*

```
#VRML V2.0 utf8
#IndexedFaceSet example: a pyramid
Shape {
appearance Appearance{
material Material {
diffuseColor  1 0 0 #simple red
}
}
geometry IndexedFaceSet {
coord DEF Coordinates Coordinate {
point [
# bottom
-1.0 -1.0 1.0,#vertex 0
1.0 -1.0 1.0,#vertex 1
1.0 -1.0 -1.0,#vertex 2
-1.0 -1.0 -1.0,#vertex 3
# top
0.0 1.0 0.0 #vertex 4
]
}
coordIndex [
#bottom square
0, 3, 2, 1, 0, -1,
#side1
0, 1, 4, -1,
#side2
1, 2, 4, -1,
#side3
2, 3, 4, -1,
#side4
3, 0, 4, -1,
]
}
}
```

```
DEF Clock TimeSensor {
cycleInterval 2.0
loop TRUE
startTime 1.0
stopTime 0.0
}

DEF Interpolator CoordinateInterpola
key [ 0.0, 1.0, 2.0 ]
keyValue [
# 1st coordinate set
-1.0 -1.0 1.0,
1.0 -1.0 1.0,
1.0 -1.0 -1.0,
-1.0 -1.0 -1.0,
0.0 1.0 0.0,
# 2nd coordinate set
-1.0 -1.0 -1.0,
-1.0 -1.0 1.0,
1.0 -1.0 1.0,
1.0 -1.0 -1.0,
0.0 1.0 0.0,
# 3rd coordinate set
1.0 -1.0 -1.0,
-1.0 -1.0 -1.0,
-1.0 -1.0 1.0,
1.0 -1.0 1.0,
0.0 1.0 0.0,
]
}

ROUTE Clock.fraction_changed TO Interpolator.set_fraction
ROUTE Interpolator.value_changed TO Coordinates.set_point
```



### 4.3 Communication server/client : CGI script

The MVL has been written in HTML (mixed occasionally with PHP code). For the development of the Mechanical Virtual Laboratory, we have used Apache server (Lamp package for Unix and Wamp package for Windows).

The input fields of the data sent to the CGI script are specified inside a form tag :

```
<form  NAME = "data" METHOD ="GET" ACTION="/cgi-bin/ellipsograph.cgi"
ONSUMIT="validation() return">
<input type = "text" name = "m" size = "2" value="80" maxlength = "3">
<input type = "text" name = "n" size = "2" value="30" maxlength = "3">
<input type = "submit" value = "Run CGI script" onclick="changeImage()">
</form>
```

The ACTION attribute indicates the URL of the EasyDyn application and the METHOD attribute specifies the method used for the HTTP request. The GET method sends the data of the form to the server, appending them to the URL indicated in the ACTION attribute as follows: /cgi-bin/ellipsograph.cgi?m=data1&n=data2&... . The ONSUMIT attribute allows to run a javascript function when the data of the form is sent to the server. The function validation() checks the content of the HTML form in order to ensure that the data specified by the user have physical sense; otherwise, an alert is sent back. For example, in the case of the ellipsograph mechanism, the distance between the center of the sliders has to be smaller than the length of the connecting rod (see code below).

```
function validation(data)
{
if (parseFloat(data.m.value) < parseFloat(data.n.value))
{
return true;
}
else
{
alert ("error message: the distance between the center of the sliders has
to be smaller than the length of the connecting rod");
return false;
}
}
```

The attribute onclick allows to execute a javascript routine when a button is clicked. In the example above, the routine ChangeImage() is called when the user clicks on the button Run CGI script. This routine updates the animated gif generated by Gnuplot with the values introduced by the user in the HTML form:

```
function ChangeImage()
{
setTimeout(changeImage, 1000);
```

```
document.getElementById("ellipse").src="Gnuplot/ellipse.gif?"+new Date().getTime();
}
```

Given that the updating of the animated gif isn't instantaneous, the execution of the routine ChangeImage() must be done with a delay: the setTimeout() method allows to call a javascript function after a specified number of milliseconds. In our example the delay is of 1000 ms; which corresponds approximately to the time of execution of the CGI script.

The data are then sent to the EasyDyn application via Common Gateway Interface (CGI). The CGI is a set of rules allowing to run programs on a web server and specifying how information is transmitted between the web server and the client browser [17]. Programs using CGI are commonly called CGI scripts and are placed in the cgi-bin directory of the root directory of the website. With an HTTP GET request, the data stored in the end of the URL are retrieved by the CGI script with the QUERY_STRING environment variable.

```
#include <stdio.h>
#include <string.h>
char *data *argument;
data=getenv("QUERY_STRING");
argument=strtok(data,"&");
```

The VRML code generated with CGI EasyDyn script is sent to the standard output stream (sdtout):

```
cout << "Content-type:model/vrml\r\n\r\n";
cout << "#VRML V2.0 utf8\n";
...
// Definition of the scene
cout << "DEF points Shape {\n";
cout << "geometry PointSet{ \n";
cout << "coord DEF noeuds Coordinate {\n";
cout << "point [\n";
for (i=0; i<NbrNodes; i++)
cout << node[i].x << " " << node[i].y << " "  << node[i].z <<",\n";
cout << "] } } }\n";
...
cout <<"DEF TimeSource  TimeSensor {\n";
cout <<"cycleInterval "<< CycleInterval <<"\n";
cout << "startTime 0\n";
cout << "stopTime 0\n";
cout <<"loop TRUE }\n";
cout <<"ROUTE TimeSource.fraction_changed TO animation.set_fraction\n";
cout <<"ROUTE animation.value_changed TO noeuds.set_point\n";
```

Finally, the server sends back to the client browser the VRML code through a Hypertext Transfer Protocol (HTTP) using an appropriate Content-Type HTTP header field. The

Content-Type HTTP header field, so called MIME (Multipurpose Internet Mail Extension), is an Internet standard which indicates to the client browser the media type of the response sent by the server. The client browser uses this Content-Type header to select an appropriate application to read the response of server. The official MIME type for VRML files is defined as [12]: model/vrml. Therefore, an appropriate browser plugin has to properly installed to display the VRML animations.

Besides, during the execution of the CGI script, the Gnuplot software is called. Gnuplot creates the animated curve from a data file[1] generated by the EasyDyn library.

```
//RUN GNUPLOT
FILE *gp;
        gp = popen(GNUPLOT_PATH, "w");
        if(gp == NULL){
            fprintf(stderr, "Oops, I can't find %s.", GNUPLOT_PATH);
             exit(EXIT_FAILURE);
        }
        fprintf(gp, " cd  '/media/www/VirtualLab/Gnuplot/' \n");
        fprintf(gp, " load 'ellipse.plt' \n");
          fflush(gp);
          pclose(gp);
```

The communication between the server and the client browser is depicted in Figure 5.

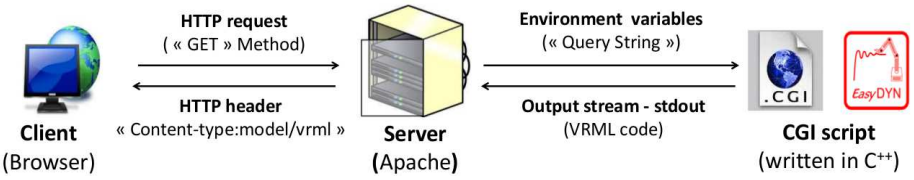



*Figure 5: Communication Client/Server*

## 5 PHYSICAL MODEL – ADDITIVE MANUFACTURING

The Machine Design and Production Engineering Unit of our university hosts a Fabrication Laboratory[2] (FabLab) equipped, among others, with three Ultimaker 3D printers (Figure 6). The Ultimaker 3D printers use an Additive Manufacturing (AM) technology. In particular, the manufacturing technology implemented in the Ultimaker 3D printer is the Fused Deposition Modeling FDM. The (FDM) process uses a thermoplastic filament which is unwound from a coil, heated to its melting point, and then extruded through a nozzle. The object is created by laying down successive layers of filament. The interested reader can find more details in [18].

[1] The data file is a text file containing the coordinates of points as columns.
[2] The interested reader can find more informations on our web site : www.fablabmons.be.

The geometry of the mechanical part to be manufactured is first modelled using the SolidWorks CAD software. With SolidWorks the model design can be exported to a VRML format as well as a STL format. The VRML format allows to easily provide virtual models, downloadable from our Mechanical Virtual Laboratory, of mechanical parts studied in the framework of the course of geometry. The STL (STereoLithography) format is a triangular representation of a 3-dimensional surface geometry and it is the standard file type used by most additive manufacturing systems. Before starting the 3D printing of the mechanical part, the STL file needs to be converted into a format that the printer can understand. The g-code file is the common language used to control automated machine tools. We have used the software Cura to prepare the model for the 3D printing [19]. Cura automatically generates from a STL file the g-code which contains the instructions to run the 3D printer.

Figure 7 shows an example of polyhedral solid modelled with SolidWorks and its physical counterpart.

The 3D printing has significantly expanded in recent years. So that, Internet provides many online libraries of printable teaching models . Most of the time, these models are freely downloadable in STL format. A few of the mechanisms studied within the Theoretical Mechanics course are directly printed from these web sites. However, the STL format doesn't permit to customize the model for specific personal learning needs. For this reason, we recently started to develop our own 3D-printable models. Because the initiative is quite recent, only two mechanisms have been modelled and printed to date : an ellipsograph and a simple epicyclic gear train. Figure 8 shows the example of the ellipsograph mechanism.

## 6 EXISTING EXAMPLES AND FUTURE WORK

Currently, the model of a dozen of mechanical systems are available on our Virtual Mechanical Laboratory. Figure 9 shows a few examples of the mechanisms which have been modelled. For most mechanisms, only the kinematic aspects are implemented. However,

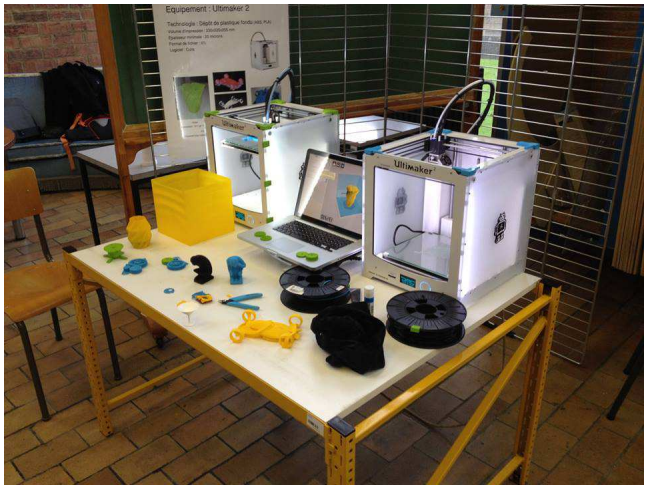
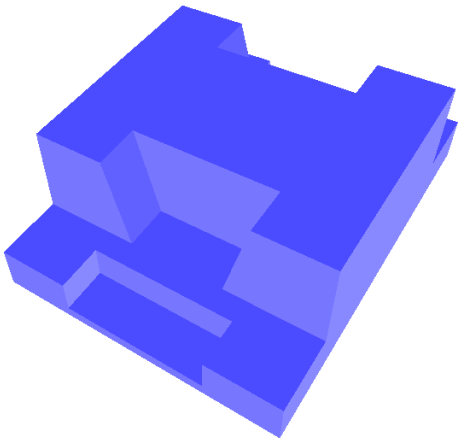
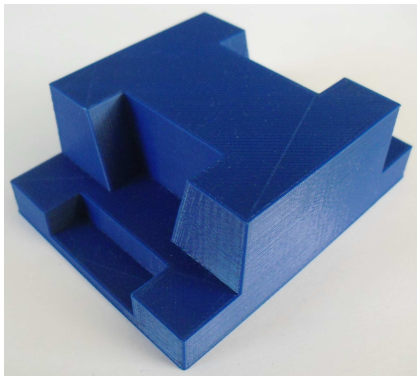



*Figure 6: Ultimaker 3D printers of FabLab (University of Mons)*

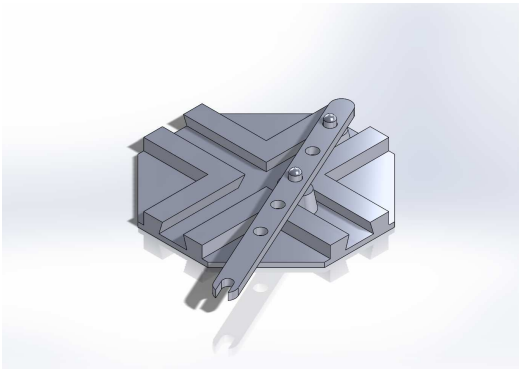


(1) Virtual model (VRML)

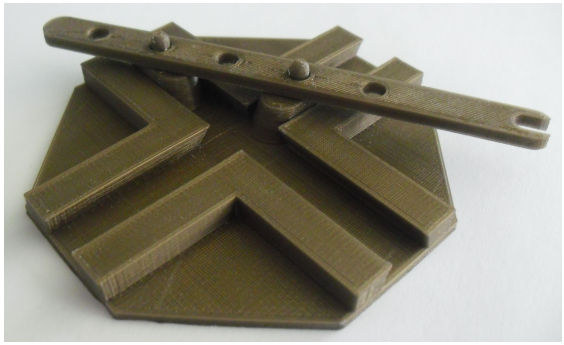


(2) Physical model

*Figure 7: Modelling of polyhedral solid with SolidWorks CAD software*



(1) SolidWorks model



(2) rapid prototyping model

*Figure 8: Ellipsograph of Archimedes*

the dynamic aspects can easily been introduced in the modelling with the `mbs` and `sim` modules of `EasyDyn`. In the case of the harmonic oscillator mechanism, the user can modify the stiffness of the spring, the value of the mass and the viscous damping coefficient, and directly visualize on the `web` page the time-history of the mechanical response (expressed in terms of displacement, velocity or acceleration).

The Mechanical Virtual Laboratory was initiated about two years ago and it is used during the exercise sessions as teaching support. The students have a direct connection with the `MVL` from our existing e-learning platform (the `Moodle` free learning platform is used at the University of Mons).

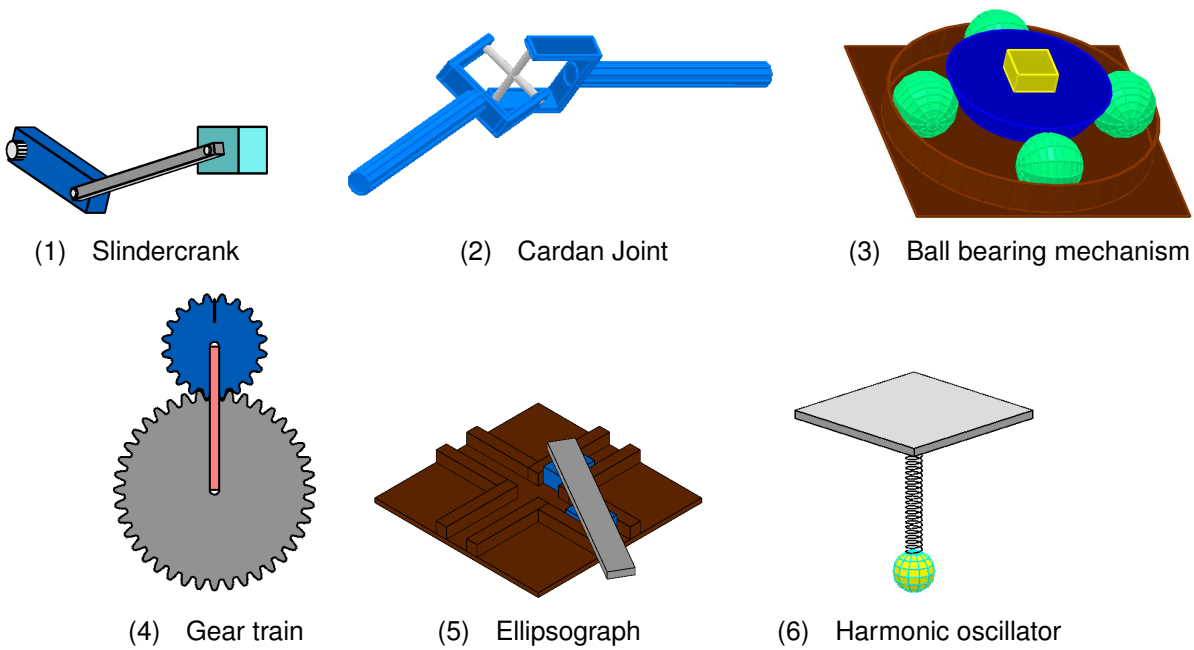
Our Mechanical Virtual Laboratory was actually available to students form the academic year 2013-2014 and it is included in exercise sessions of the course of Theoretical Mechanics, exclusively to illustrate the kinematics of mechanisms. Besides, the students can access the Mechanical Virtual Lab from our e-learning platform (the `Moodle` free learning platform is used at the University of Mons).

Presentlye, we don't know the percentage of students who exploit the Virtual Laboratory at home. Therefore, in addition to expanding our collection of virtual and physical models, we also envisage to realize a survey among the students in order to collect their opinion about our project. , and more importantly to evaluate how the Mechanical Virtual Lab contributed to improve their spatial skills. The PSVT-R test could be used for that purpose. The PSVT-R test is the spatial visualization test which is the most widespread in the scientific literature [20].

## 7 CONCLUSION

Engineers often require the ability to visualize their ideas on paper or computer screens. That is why, the spatial skills are developed from the first year of the engineering study programs; mainly through Geometry and Theoretical Mechanics courses. Often, the first engineering students have weaknesses in spatial visualization. In this context, the pedagogical support unit and the units of the department of mechanical engineering collaborate in the development of a pedagogical project to improve the spatial skill of the students. In order to establish a relationship between the two courses, we were interested in the study of mechanical systems. The CAD modelling and the technical drawing of mechanical parts constituting a mechanism are carried out during the laboratory training of the course of Geometry. The Kinematics are then treated in the course of Theoretical Mechanics.

The paper described the numerical tools that we have used and developed to build up a Mechanical Virtual Laboratory. Our Mechanical Virtual Laboratory is a dynamic web site which offers several interactive animations in VRML format. The mechanical systems



(1) Slindercrank (2) Cardan Joint (3) Ball bearing mechanism

(4) Gear train (5) Ellipsograph (6) Harmonic oscillator

*Figure 9: Examples of scenes built with EasyDyn framework*

are simulated with a home made C++ simulation library called EasyDyn. Practically, each mechanical system is defined in a specific C++ file which generates the VRML code which can be directly visualized in a web browser. For each animation, the user can modify some geometrical and physical parameters and visualize the effects on the mechanism by displaying interesting curves such as velocity or acceleration. The modifiable input parameters of the mechanical system are introduced by the user via a HTML form and sent by Common Gateway Interface (CGI) to the EasyDyn application (server-side processing) which simulates the mechanism and updates the VRML animation file as well as the display of velocity/acceleration curves.

The pedagogical literature highlights the importance of the hands-on models for teaching across all disciplines. The computer-aided modeling tool and the rapid prototyping technology offer the opportunity to enhance accessibility to physical models. We recently decided to combine our Virtual Mechanical Laboratory with printable 3D models dedicated to kinematics teaching. Our long-term aim is the creation of our own repository of 3D printable models for teaching kinematics.

## 8 ACKNOWLEDGMENTS

This work is supported by the FabLab of the Mons town (www.fablabmons.be) for the rapid prototyping of mechanisms presented in this paper.

## References

[1] Koch D., "The effects of solid modeling and visualization on technical problem solving," *Journal of Technology Education*, vol. 22, no. 2, pp. 3–21, 2011. available via https://scholar.lib.vt.edu/.

[2] Garmendia M, Guisasola J., and Sierra E., "First-year engineering students' difficulties in visualization and drawing tasks," *European Journal of Engineering Education*, vol. 32, no. 3, pp. 315–323, 2007.

[3] Nagy-Kondor R., "Spatial ability of engineering students," *Annales Mathematicae et Informaticae*, vol. 34, pp. 113–122, 2007. available via http://ami.ektf.hu/.

[4] Sierra Uria E, Garmendia Mugica M, Muniozguren Colindres J. , and Larrakoetxea Madariaga I., "Methodology for part visualization problem solving (reading, interpretation and creation of multiview technical drawings)," in *Proceedings of the 17th International Conference on Engineering Design (ICED 09)*, (Palo Alto (USA)), pp. 1–12, August 2009.

[5] Wattiaux D. and Verlinden O., "Development of a virtual laboratory in the framework of the education of theoretical mechanics taught to undergraduate engineering students," in *Proceedings (on CD) of the ECCOMAS Thematic Conference Multibody Dynamics*, (Zagreb (Croatie)), pp. 1–11, July 2013.

[6] Wattiaux D., Verlinden O., and Renotte Ch., "Project of mechanical virtual laboratory for the euducation of theoretical mechanics to undergraduate engineering students," in *Colloque de l'enseignement des Technologies et des sciences de l'information et des Sytmes*, (Besanon (France)), pp. 1–6, October 2014.

[7] Lipson H., "Printable 3d models for customized hands-on education," in *Proceedings of Mass Customization and Personalization (MCPC)*, (Cambridge), October 2007. available via http://creativemachines.cornell.edu/publications.

[8] Verlinden O., Kouroussis G., and Conti C., "EasyDyn: a framework based on free symbolic and numerical tools for teaching multibody systems," in *Proceedings (on CD) of the ECCOMAS Thematic Conference Multibody Dynamics*, (Madrid, Spain), June 2005.

[9] Kouroussis G., Verlinden O., Rustin C., and Bombled Q, "EasyDyn: Multibody open-source framework for advanced research purposes," in *Proceedings (on CD) of the ECCOMAS Thematic Conference Multibody Dynamics*, (Brussels, Belgium), July 2011.

[10] Verlinden O., Ben Fékih L., and Kouroussis G., "Symbolic generation of the kinematics of multibody systems in easydyn : from mupad to xcas/giac," *Journal of Theoretical and Applied mechanics letters*, vol. 3, doi:10.1063/2.13013012, no. 1, 2013.

[11] Williams Th. and Kelley C., "Gnuplot : An interactive plotting program," 2011.

[12] Carey R., Bell G., and Marrin Ch., "The virtual reality modeling language(vrml97)." International Specification ISO/IEC 14772-1:1997, December 1997. available via http://www.bitmanagement.de/developer/spec/vrml97specification.pdf.

[13] Tamiosso F. S. , Raposo A. B., Magalhães L. P., and Ricarte I. L. M., "Building interactive animations using vrml and java," in *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing*, (Brazil), 1997.

[14] Taubin G., Horn W., Lazarus F., and Rossignac J., "Geometry coding and vrml," *IEEE Journals and Magazines*, vol. 86, no. 3, 1998.

[15] Brutzman D., "The virtual reality modeling language and java," *Journal of Communications of the ACM*, vol. 46, no. 6, pp. 57–64, 1998.

[16] Lai F. M. and Sourin A., "Function-defined shape node for vrml." The Eurographics Association, 2002. available via http://cis.k.hosei.ac.jp/ F-rep/LaiSourEG02.pdf.

[17] G. S., *CGI Programming on the World Wide Web*. O'Reilly Media, Inc, USA, 1996.

[18] Gibson I., Rosen D. W., and Stucker B., *Additive Manufacturing Technologies : Rapid Prototyping to Direct Digital Manufacturing*. New-York, USA: Springer, 2010.

[19] "Cura : Ultimaker's software for making 3d prints – user manual."

[20] Maeda Y, Yoon S., Kang G., and Imbrie P.K , "The validation of the revised psvt:r for measuring first year engineering students spatial ability," *International Journal of Engineering Education*, vol. 29, no. 3, pp. 763–776, 2013. available via http://www.ijee.ie/contents/c290313.html.