

# **Would Wood 2: Large-scale robotic 3D printing using biobased composites**

## **Result report for Work Package 3: Information Systems**

**Malgorzata Zboinska**

PhD, Architect SAR/MSA, IARP

Associate Professor in Digital Design, Fabrication and New Media Art

Creative Leader of the Robotic Fabrication Lab

Department of Architecture and Civil Engineering, Chalmers University of Technology

# **Work Package 3: Information Systems**

## **I. Developed foundations and frameworks**

## Work package 3 - basic data

- Leader: Malgorzata Zboinska (Chalmers, Architecture and Civil Engineering)
- Main partners: White arkitekter, Scania, KTH Machine Design
- Primary input from:
  - WP2 - Design for Architecture, Structure and Products
  - WP5 - Printer development
  - WP6 - Manufacture and Construction

## Work package - overview

The Information Systems work package provides an overview of the potential workflows and digital toolsets supporting the process of designing and additively manufacturing large-scale elements from cellulose-based composites. The Information Systems in question embrace the most important types of information exchanges between the different product development stages (design, simulation, fabrication, assembly, use, disassembly, reuse/recycling) and the different actors (e.g. architecture - automotive design exchanges, designer - fabricator exchanges).

## Information systems - introductory information

The work within the Information Systems package was focused on outlining the digital workflows, data exchanges and software toolkits that can aid the transition from a digital design model to a large-scale physical model (architectural and automotive) that is additively manufactured. The theoretical frameworks for workflows and toolsets developed in WP3 were then implemented and specified in demonstrator developments done within Work Packages 3 and 5.

Most fundamentally, the transition from the digital to a physical model requires a translation of 3D data - from a 3D surface representation to a polyline toolpath representation. The toolpath model approximates the surface model to enable its manufacturing. It contains geometrical data (points and lines) that describes the movement path of a fabrication machine that gradually builds up the material layer by layer to create the physical piece. In the final stage of data transition and exchange, the toolpath model data is exported as G-code, using syntax understandable for the particular digital machine chosen to manufacture the piece.

In the WouldWood project, a 6-axis robot arm of the ABB brand is chosen as the 3D printing medium. This entails that the G-code generated from the toolpath model needs to undergo postprocessing in a dedicated software. Namely, inverse kinematic calculations need to be done, to align the 3D tool position from the original G-code into robot axis values for each point. In the WouldWood project, this is done using a dedicated kinematic postprocessor Robot Studio for ABB robots.

An alternative approach would be to use custom parametric robot control add-ins (e.g. KUKA|prc or HAL) directly in a 3D parametric modelling environment (e.g. Grasshopper in Rhinoceros 3D). This approach enables the designer to directly redesign the toolpath model in parallel to final robot code generation. In this way, it helps to program and simulate robot motions and adapt them to the design changes in real time. This approach is effective as it eliminates the need to go through multiple export/import steps between CAD, CAM and the robot code postprocessor. Another benefit for design is the visual simulation of the robotic 3D printing process, which enables ocular and to some extent also computational detection of errors within the toolpath as well as collisions of the machine with the printed object. With this approach, non-standard 3D printing operations departing from the layer-by-layer standard, such

as free-form 3D printing in space, can be easily designed, simulated and executed. However, this approach is not used in the WouldWood project, as it is not compatible to implement directly with the ABB robot and its control firmware Robot Studio.

The toolpath model is of central importance not only for manufacturing but also for the entire digital design process as it allows for the creation of information models for the later design stages. In those stages, accurate data about the 3D printed geometry is needed, e.g. to be able to plan of the post-processing procedures for the 3D printed pieces via CNC milling, to enable computational performance analyses and simulations (structural, thermal etc.), life cycle analyses, assembly planning and detailed design of joints between the 3D printed elements (Fig.1).

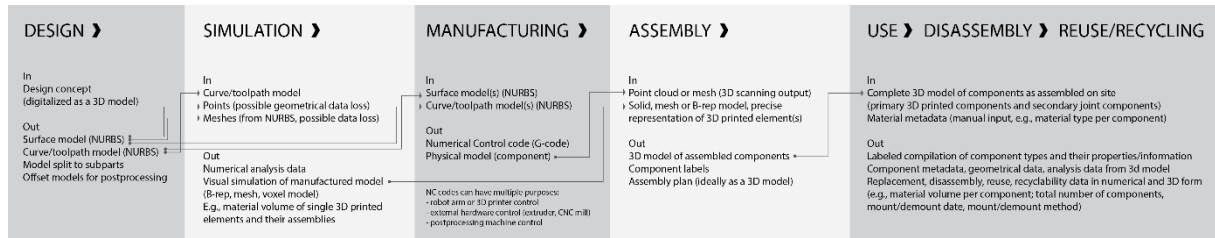


Fig.1. A general framework presenting the conceptual information systems framework for the WouldWood project, outlining examples of 3D data inputs and outputs in the different project development phases and indicating the types and routes of data exchanged between the phases. © Malgorzata Zboinska.

In standard 3D printing, i.e. layer by layer, the optimal file exchange formats between design and manufacturing are IGES and STL as they guarantee minimum geometrical data losses and they are generally accepted as industry standards. In non-standard 3D printing, in which a toolpath model is the central part of the design process (e.g. designed from scratch instead of being generated from the surface model), the data exchanges from design to manufacturing can be done via G-code exchanges (Fig.2). The G-code is generated based on popular 3D printer standards (Makerbot, Ultimaker) and uploaded into the robot programming software (e.g. ABB Robot Studio) and translated into robot-specific code. As already mentioned, another option is to generate the robot program directly from the toolpath using dedicated add-ons (e.g. Kuka|prc or HAL added onto the visual programming interface Grasshopper in a 3D modeller Rhinoceros). This option is the most versatile from the design and manufacturing standpoint as it allows for detailed control of the material deposition process, including customized programming of variable speed deposition, angled extrusion and extrusion in three dimensions without supporting structures.

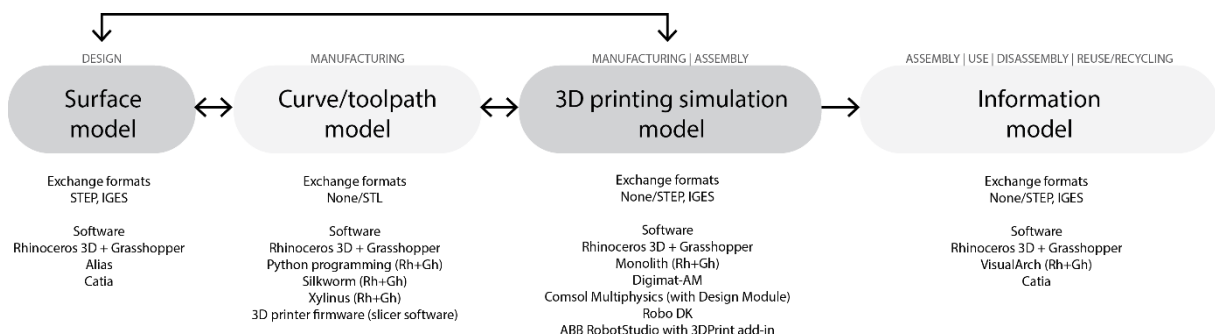


Fig.2. Examples of typical 3D model representations that can be exchanged in the different design and manufacturing stages, with examples of accompanying file formats and software enabling the exchanges. © Malgorzata Zboinska.

# **Work Package 3: Information Systems**

## **II. Developed digital workflow proposals**

## Digital workflow alternatives for 3D printing in wood-based composites

Based on the review of the current state of the art in 3D printing research, in WP3 several alternatives for approaching the 3D printing process were specified and considered for driving the project developments. They are shown in Table 1. Each of these alternatives centers on the idea of manipulating particular parameters in the design-to-manufacturing process, such as deposition, toolpath, G-code, hardware and material properties.

Table 1. Comparison between parameter changes in the respective 3D printing approaches. (\*) indicates approaches tested in the project. © Malgorzata Zboinska.

3D PRINTING APPROACH	CUSTOMIZABLE PARAMETERS				
	Deposition	Toolpath	G-code	Hardware	Material
Standard	x*				
Material-driven	x*	x	x	x*	x*
G-code-driven (includes freeform 3D printing)	x		x*	x*	
Toolpath-driven		x*			

### Standard 3D printing

This mode entails layer-by-layer deposition. Figure 3 presents its typical steps. Here, the 3D surface model of the design part to be manufactured is exported into an STL file format and imported into off-the-shelf slicer software for 3D printers (e.g. Cura, Makerware). Therein, the 3D model is sliced to generate curves that demarcate the final toolpath model. The geometrical data describing this model is then used to output the G-code. If robotic 3D printing is to be used, this G-code is missing the robot axis positioning and rotation data. This data is generated through the already described inverse kinematic calculations done in robot firmware (e.g. ABB RobotStudio). This enables outputting proper G-code that can drive the robot. In this case, material deposition settings are set up in the robot firmware.

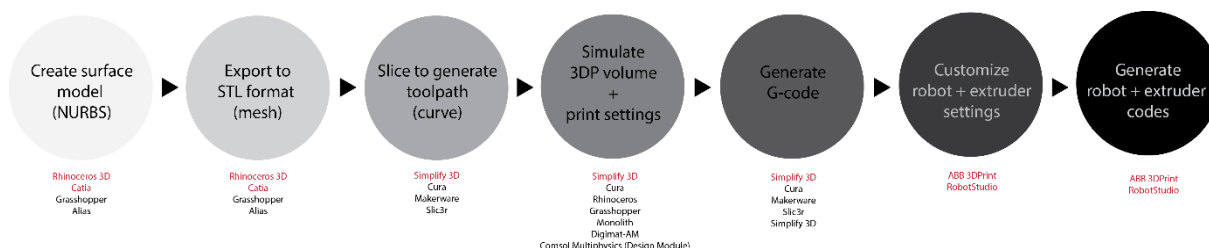


Fig.3. Information system and workflow for standard layer-by-layer robotic 3D printing. © Malgorzata Zboinska.

This approach was used in the project to produce the first 3D printed model mockups for various wood-based material compositions, as well as in the manufacturing of Demonstrator 1 (automotive). The automotive demonstrator was additionally postprocessed through CNC milling to achieve desired surface texturing effects. Its support structures were 3D printed using the freeform spatial printing approach, described in subsection 2.3. further below.

### Material-driven 3D printing

This mode focuses on using non-standard material composition properties as drivers of parameters in the 3D printing process. The printing itself is usually done layer by layer. At the level of information systems, this approach follows the workflows and data exchanges of the standard 3D printing approach, described in subsection 2.1 and shown in Figure 3. The only distinguishing element here is that the 3D printing process parameters, e.g. feed rate and extruder temperature, are being adapted to the needs of a particular material composition being

used for 3D printing. At the same time, non-standard 3D printing approaches can also be implemented here, such as freely printing in 3D space.

In the project, many different material compositions were tested through standard layer-by-layer printing. This required the development of various extruder designs and varying parameters of the 3D printing process. Moreover, a method of printing in air with a resin-coated paper thread was also developed (within Work Package 5), embracing a custom-developed printer head design and a custom-developed G-code featuring the rotation of the 6<sup>th</sup> robot axis based on the normal vector parameters of the printed object (described further below).

### G-code-driven 3D printing

This mode features custom development of G-code from scratch, based on geometrical data from the 3D model of the part to be printed. The purpose of this approach is to achieve greater flexibility when it comes to the esthetic design of the piece to be printed and its way of manufacturing that preferably departs from the layer-by-layer standard and enables setting up custom printing process parameters that vary along the length of the toolpath. The workflow for generating the custom G-code is presented in Figure 4. Once created, this G-code still needs to be postprocessed in the robot firmware, to generate robot code with a complete dataset embracing positions and rotation angles for all robot axes.

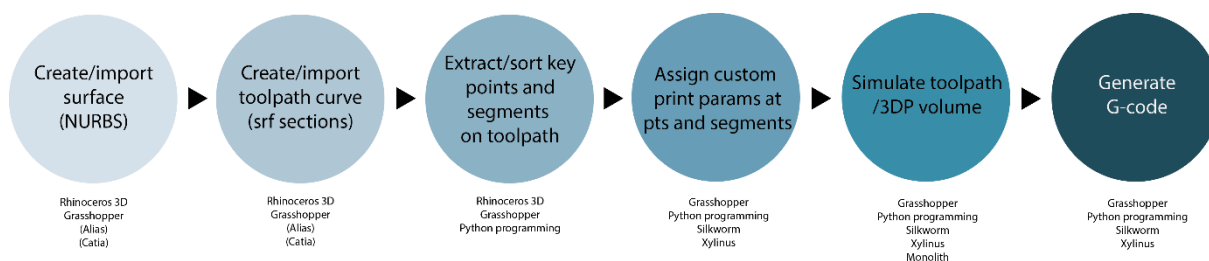


Fig.4. Information system and workflow for G-code-driven 3D printing. © Malgorzata Zboinska.

In the project, the G-code-driven approach was used to steer the rotation of the 6<sup>th</sup> axis of the robot (the one where the extruding tool is mounted). This was not possible to set up in the robot firmware, preventing the implementation of freeform printing in 3D space. Therefore, the custom G-code generation approach was implemented in the parametric modeling environment Grasshopper in 3D modeler Rhinoceros. Therein, two visual programming scripts were created – one enabling the generation of G-code from any toolpath using correct syntax (Ultimaker-based) and another one which enabled the expression of normal vector parameters following the shape of the printed object in the G-code. This enabled the fabrication in Demonstrator 2 (architectural) featuring angled extrusion on angled supports, the development of supports in Demonstrator 1 (automotive) as well as the development of the new 3D printing method using the coated thread in Work Package 5.

Moreover, a combination of a G-code-driven approach with a standard one was also tested. The design workflow featured a division of the piece to be printed into two parts – a base part printed layer-by-layer and a top part printed using a custom G-code setup featuring extrusion with Z-height differentiation. This workflow is presented in Figure 5.

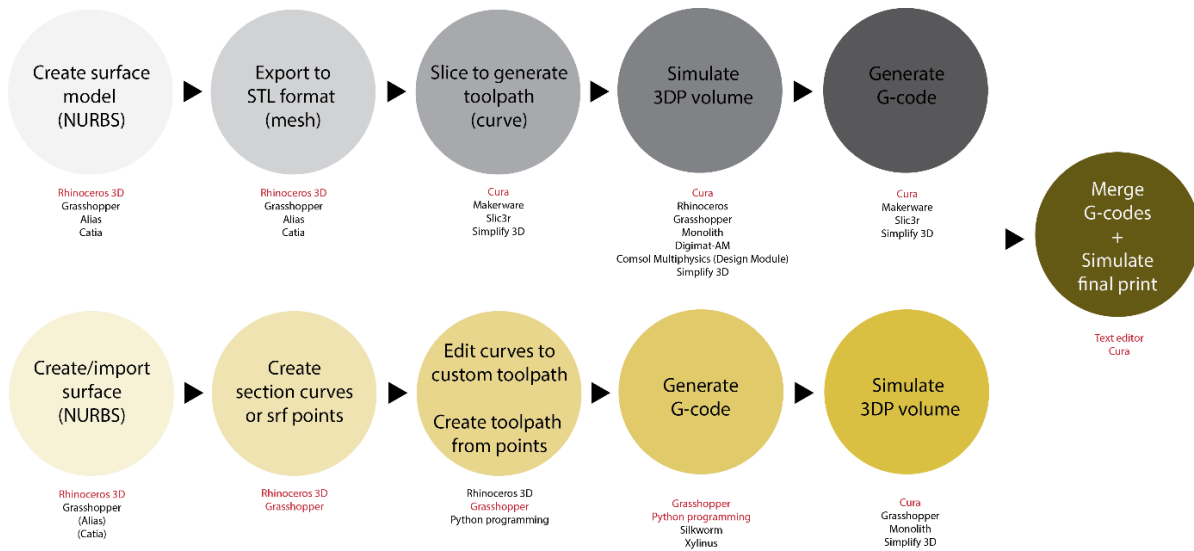


Fig.5. Information system and workflow for a combined 3D printing approach featuring G-code-driven and standard 3D printing. © Malgorzata Zboinska.

### Toolpath-driven 3D printing

This mode features standard layer-by-layer extrusion but featuring a custom toolpath design that affects the way that the material settles upon deposition. The purpose of this approach is on the one hand to enable the design of complex geometries of the printed parts without the need of modeling them as 3D surfaces. Instead, they can be designed and represented directly as toolpath curves. On the other hand, the approach also enables the creation non-standard effects on the surface of the 3D printed piece, achieved by allowing some parts of the material to be unsupported and therefore to sag and settle down as surface artefacts with a decorative function. The workflow for this approach is presented in Figure 6. Once created, the G-code of the custom toolpath still needs to be postprocessed in the robot firmware, to generate robot code with a complete dataset embracing positions and rotation angles for all robot axes.

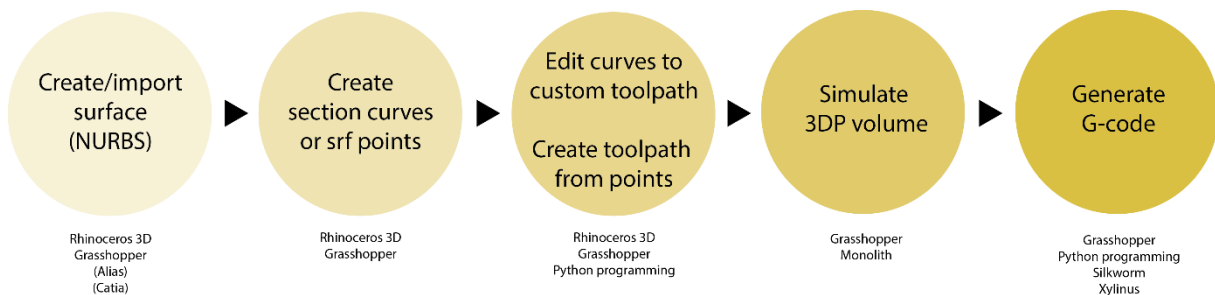


Fig.6. Information system and workflow for toolpath-driven 3D printing. © Malgorzata Zboinska.

In the project, this approach was used to develop Demonstrator 4 (architectural). The workflow featured the generation of toolpaths as slices from a standard 3D model and an application of deviations to some points along the toolpaths to enable material sagging. Then, G-code in Ultimaker syntax was generated and imported into robot firmware, in which the final robot code was created for demonstrator manufacturing.

## **Work Package 3: Information Systems**

### **III. Proposed workflow examples, partially tested in demonstrator developments**

## Information Systems - Demonstrator 1 (automotive)

Workflows and digital tools:

1. Create a 2D pattern to be applied on the top surface of the part (vector-based curves in Adobe Illustrator).
2. Create a 3D model of the part (surface modeling in Alias®; IGES file format used for exchange with Rhinoceros® and Grasshopper®).
3. Import 2D pattern curves into Rhinoceros® and use them as a basis to create parametric 3D curves on the top surface of the 3D model (Rhinoceros®, Grasshopper®).
4. From parametric surface and curve 3D data, generate basic G-code for 3D printing (Grasshopper®).
5. Import basic G-code into the robot's kinematic postprocessor and generate proper robot code (ABB Robot Studio®).
6. Use the final robot code to 3D print the object (ABB robot).
7. Finish the top surface through CNC milling of the part to get a finely treated surface (using the 3D model in step 1 as input for the milling operation).

## Information Systems - Demonstrator 2 (architectural)

Workflows and digital tools:

1. Create a 3D mesh representing the part (generated in stepwise transition from NURBS surface model through voxel-based 3D curve model to 3D mesh) in Rhinoceros®, Grasshopper®).
2. Generate mesh contours as parametric horizontal 2D section curves (Rhinoceros®, Grasshopper®).
3. Deviate points on 2D curves to enable a local unsupported extrusion effect; points of deviation selected based on bitmap image sampling (Rhinoceros®, Grasshopper®).
4. From parametric surface and curve 3D data, generate basic G-code for 3D printing (Grasshopper®).
5. Import basic G-code into the robot's kinematic postprocessor and generate proper robot code (ABB Robot Studio®).
6. Use the final robot code to 3D print the object (ABB robot).

## Information Systems - Demonstrator 3 (general, for spatial 3D printing tests)

Workflows and digital tools:

1. Generate 3D toolpath representing the geometry of the part as a free-form 3D curve that can be 3D printed by extruder rotation to match the curve's normal vector directions (Rhinoceros®, Grasshopper®).
2. From parametric 3D curve data, generate basic G-code for 3D printing (Grasshopper®).
3. Import basic G-code into the robot's kinematic postprocessor and generate proper robot code (ABB Robot Studio®).
4. Use the final robot code to 3D print the object (ABB robot).