Universidade do Minho
Escola de Engenharia
Departamento de Engenharia Mecatrónica

José Miguel Alves Pires

Development of a multimaterial laser sintering/melting equipment

July 2020

Universidade do Minho
Escola de Engenharia
Departamento de Engenharia Mecatrónica

José Miguel Alves Pires

Development of a multimaterial laser sintering/melting equipment

Master dissertation
Master Degree in Mechatronics Engineering

Dissertation supervised by
Professor Doutor Óscar Samuel Novais de Carvalho
Professor Doutor Paulo Francisco Silva Cardoso

July 2020

To my parents,
for always pushing me forward.

”Wir müssen wissen,
Wir werden wissen”

David Hilbert, 1930

Acknowledgements

First and foremost, I would like to thank...

Abstract

Functional design is the desirable and most sustainable method to design products: by adding the material only where is strictly required to perform its function, the resources usage is optimized, especially materials and energy. However, functional design may dictate the usage of several materials or a combination of them to fulfill its goal, which is hindered by the current manufacturing methodologies. An example of a class of products where functional design is key is biomedical implants, like the hip implant.

Keywords — functional design, multimaterial, laser based additive manufacturing

Resumo

O design funcional é o método mais desejável e sustentável de projetar produtos: adicionando material somente é estritamente necessário para a sua função, o uso de recursos é otimizado, especialmente materiais e

Palavras-chave — design funcional, multimaterial, manufatura aditiva baseada em laser

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

Notation	Description	Page List
FGM	Functionally Graded Material	1, 3, 8
GUI	Graphical User Interface	33, 43, 46, 47, 77
HTML	Hypertext Markup Language	17
I/O	Input/Output	70
IC	Integrated Circuit	59
LBAM	Laser-based Additive Manufacturing	3–6, 19, 20, 52, 70, 89, 103, 105, 108, 109
LOM	Laminated Object Manufacturing	5, 8
LVM	Linux Virtual Machine	33
MMAM	Multi-Material Additive Manufacturing	3, 7
MMFGM	Multi-Material Functionally Graded Material	2, 20
MMS	Multi-Material Sintering	54, 58, 72, 78, 79, 88, 97, 101–105, 125–127
MMSLS	Multi-Material SLS	19, 28, 49, 52, 54, 72, 77, 95, 107
NC	Numerical Control	23
OMT	Object-Modeling Technique	16
OOSE	Object Oriented Software Engineering	16
PBF	Powder Bed Fusion	6

Notation	Description	Page List
PCB	Printed Circuit Board	63, 87, 109
PCI	Peripheral Component Interconnect	72
PID	Proportional Integrative Derivative	60, 62, 63, 67, 100
POV	Persistence Of Vision	32
PWM	Pulse-Width Modulation	59, 69
RTOS	Real-Time Operating System	84
SLM	Selective Laser Melting	3, 6, 8, 10, 19, 28, 106–108
SLS	Selective Laser Sintering	3, 6, 8, 10, 28, 90, 91, 98, 106–108
SSR	Solid-State Relay	59, 67, 69
STL	Standard Tessellation Language	5, 8, 22, 32
SVG	Scalable Vector Graphics	17, 18, 30, 31, 33, 35, 86, 107
TIN	Triangular Irregular Network	33
UI	User Interface	95, 103, 109
UML	Unified Modeling Language	16, 21, 74, 84
USB	Universal Serial Bus	32
VDI	Verein Deutscher Ingenieure	10
XML	eXtensible Markup Language	17, 18, 30

1. Introduction

One of the ultimate goals of engineering is the ability to provide sustained development, by efficiently using the available resources — materials, energy, knowledge, time. This is a “Sisyphus’s stone”[1], a perpetual quest for optimisation of efforts and resources. One such example in the engineering field is the ability to produce components with the minimum amount of material needed for its function, i.e., functional design of components without restrictions to geometry or type and number of different materials; instead the focus should be on the desired properties of such components, customising them for the specific field of application.

1.1. Context

The functional design of components is a complex topic, with a myriad of questions to be answered: what is the function of the component?; what design criteria must be met to fulfil its function?; how will the component be produced, and what data does it require?; how will the component’s performance be measured?, among others. The answers are often not clear or simple as they dictate the use of several materials and several manufacturing technologies, increasing severely the complexity of producing such components: how to effectively combine two or more materials into a single component in a synergistic way?

1.2. Motivation

The possibility of controlling composition or structure and thus obtain components with desired local properties, as regarding mechanical, tribological, thermal properties, and others are of great interest, as material is only added where it functionally needed, minimising waste and enhancing the overall properties of the component being built.

[1]King of Ephyra, condemned to push a stone uphill for eternity — immortalised in “The Myth of Sisyphus” by Albert Camus

1.3. Main objectives

The goal of the present work is to close the gap between design and fabrication of multi-material components from metallic/ceramic materials using Selective Laser Sintering (SLS)/Selective Laser Melting (SLM) technology. To this end several main objectives have been outlined:

1. Develop a design methodology for multi-material fabrication of metals/ceramics;
2. Instantiate a practical workflow and respective toolchain from the design methodology;
3. Develop and build a proof-of-concept equipment capable of producing such components;
4. Test the production of multi-material components using the proposed workflow/toolchain and the equipment built.

1.4. Thesis organisation

This thesis is organised as follows: In Chapter 2, the state of the art of the additive manufacturing technology, Laser-based Additive Manufacturing (LBAM), and Multi-Material Additive Manufacturing (MMAM) is presented, with special focus on the last, namely on Functionally Graded Material (FGM) structures. Lastly, a brief overview of the available methodologies in these fields are presented.

In Chapter 3 the theoretical foundations are presented, namely the project development methodologies and associated tools, and the SLS/SLM process in detail.

In Chapter 4, the multi-material design and production problem and its challenges using SLS/SLM technology are presented. The methodology devised for multi-material production through the LBAM technology is presented to tackle the high complexity of the process and the lack of a supporting methodology, taking into account the key agents of the process and leveraging the process information.

In Chapter 5 is presented all the development phase of the project. A specific workflow was instantiated from the methodology, attending to the specific requirements and constraints of the project. Based on this workflow, a toolchain was assembled, designing the required software components. Finally, based on the requirements and constraints of the process itself, the mechanical and electronic infrastructures were designed, and on top of the last, the control software was designed.

In Chapter 6, the workflow and equipment were put to the test to verify their suitability to the process and their performance for multi-material component production. Additionally, production manufacturing tests were also performed. Tests were used to validate the workflow and equipment, pointing out also straightforward ways to adapt and implement custom paths for the multi-material LBAM process.

The Chapter 7 gives a summary of this thesis as well as prospect for future work.

Lastly, the appendices (see Section 7) contain detailed information...

2. State of the art

In this chapter a review on the state of the art is presented. Additive manufacturing is discussed briefly as a preliminary topic. Then, Laser-based additive manufacturing processes are discussed as viable solutions for metallic and composite manufacturing.

3. Theoretical foundations

In this chapter some background is provided for the main subjects. The fundamental technical concepts are presented as they proved its usefulness along the project, namely the project development methodologies and associated tools, and the SLS/SLM process in detail.

3.1. Project methodologies

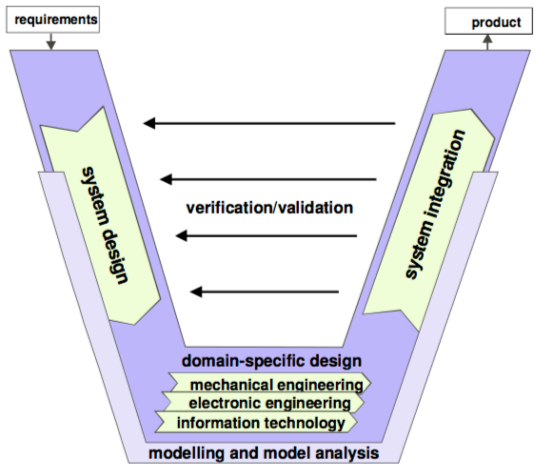
The methodologies used for the project development are briefly described next.

3.1.1. Development methodology of mechatronics — VDI 2206

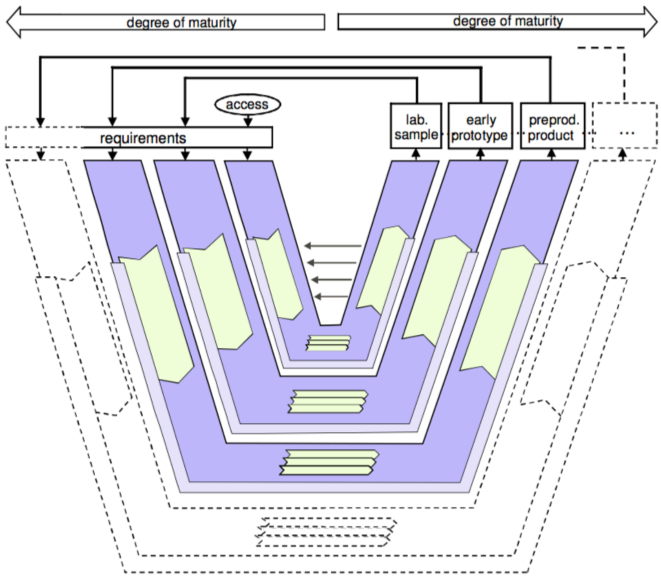
Mechatronics, was defined by Harashima et. al [1] as: "the synergistic integration of mechanical engineering with electronic and intelligent computer control in the design of industrial products and processes". This definition has more than twenty years, and nowadays it could be extended to other domains beyond the industrial sector, but a central idea remains: the synergy between these fields of knowledge, able

3.1.1.1. Process modules for recurrent working steps

The main phases in the v-model are not yet detailed: this needs to be done by the designer. However, some design procedures occur regularly during the design and were identified by field practitioners in terms of predefined process modules, representing procedures and methods for different design tasks, organised in a data base (fig. 3.2[2]). The VDI 2206 guideline provides detailed process modules for the V-model macro cycle. One such example is given in fig. 3.2 for the system design, which can be described phase-milestone-diagrams, checklists, process-flow diagrams, etc..



(a) One macro cycle



(b) Iterating over a number of macro-cycles with increasing process maturity

Figure 3.1.: Mechatronics design V-model, VDI 2206 2003 [2]

3.1.2. Waterfall

For the domain-specific design of software the waterfall methodology is used. The waterfall model (fig. 3.3) represents the first effort to conveniently tackle the increasing complexity in the software development process, being credited to Royce, in 1970, the first formal description of the model, even though he did not coin the term [3]. It envisions the optimal method as a linear sequence of phases, starting from requirement elicitation to system testing and product shipment [4] with the process flowing from the top to the bottom, like a cascading waterfall.

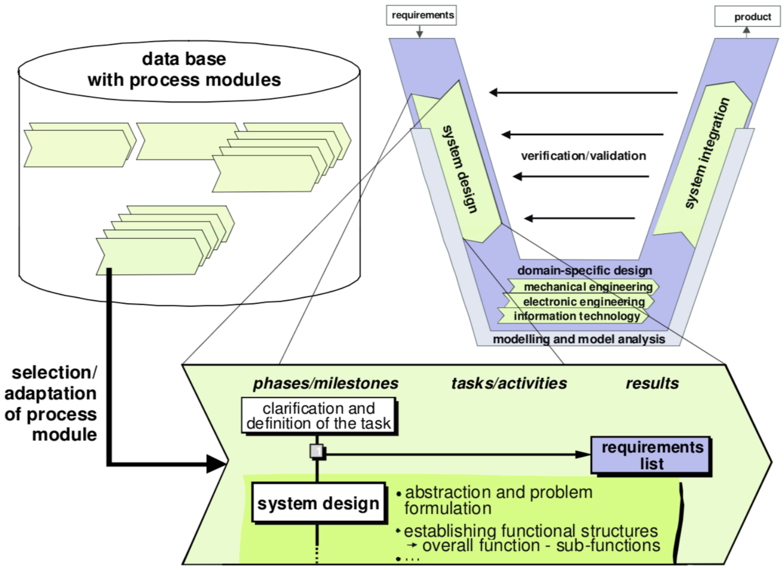


Figure 3.2.: Configuration of process modules for individual operation steps

In general, the phase sequence is as follows: analysis, design, implementation, verification and maintenance.

1. Firstly, the project requirements are elicited, identifying the key requirements and constraints the system being developed must meet from the end-user perspective, captured in natural language in a product requirements document.

2. In the analysis phase, the developer should convert the application level knowledge, enlisted as requirements, to the solution domain knowledge resulting in analysis models, schema and business rules.

3. In the design phase, a thorough specification is written allowing the transition to the implementation phase, yielding the decomposition in subsystems and the software architecture of the system.

4. In the implementation stage, the system is developed, following the specification, resulting in the source code.

5. Next, after system assembly and integration, a verification phase occurs and system tests are performed, with the systematic discovery and debugging of defects.

6. Lastly, the system becomes a product and, after deployment, the maintenance phase start, during the product life time.

While this cycle occurs, several transitions between multiple phases might happen, since an incomplete specification or new knowledge about the system, might result in the need to rethink the document.

The advantages of the waterfall model are: it is simple and easy to understand and use and the phases do not overlap; they are completed sequentially. However, it presents some drawbacks namely: difficulty to tackle change and high complexity and the high amounts of risk and uncertainty. However, in the present work, due to its simplicity, the waterfall model proves its usefulness and will be used along the project.

As a reference in the sequence of phases and the expected outcomes from each one, it will be used the chain of development activities and their products depicted in fig. 3.4 (withdrawn from [5]).

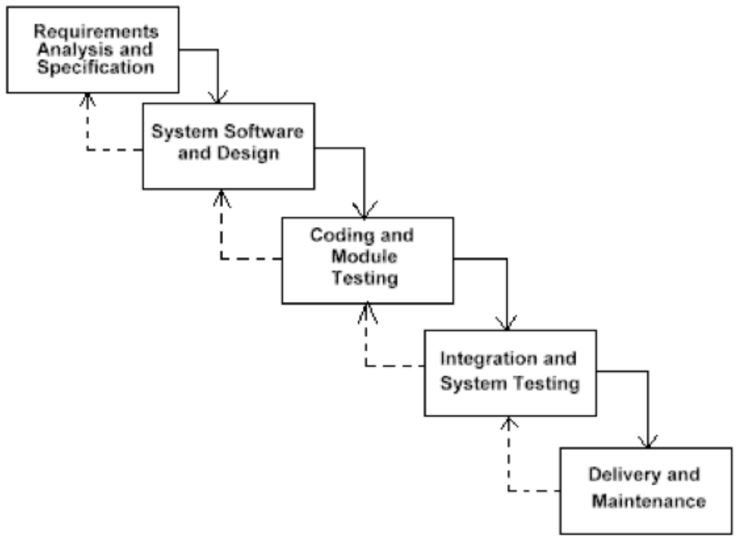


Figure 3.3.: Waterfall model diagram

3.1.3. Unified Modeling Language (UML)

To aid the software development process, a notation is required, to articulate complex ideas succinctly and precisely. The notation chosen was the Unified Modeling Language (UML), as it provides a spectrum of notations for representing different aspects of a system and has been accepted as a standard notation in the software industry [5].

The goal of UML is to provide a standard notation that can be used by all object- oriented methods and to select and integrate the best elements of precursor software notations, namely Object-Modeling Technique (OMT), Booch, and Object Oriented Software Engineering (OOSE) [5]. It provides constructs for a broad range of systems and activities (e.g., distributed systems, analysis, system design, deployment). System development focuses on three different models of the system (fig. 3.4) [5]:

1. **The functional model**: represented in UML with use case diagrams, describes the functionality of the system from the user's point of view.

2. **The object model**: represented in UML with class diagrams, describes the structure of the system in terms of objects, attributes, associations, and operations.

3. **The dynamic model**: represented in UML with interaction diagrams, state-machine diagrams, and activity diagrams, describes the internal behaviour of the system.

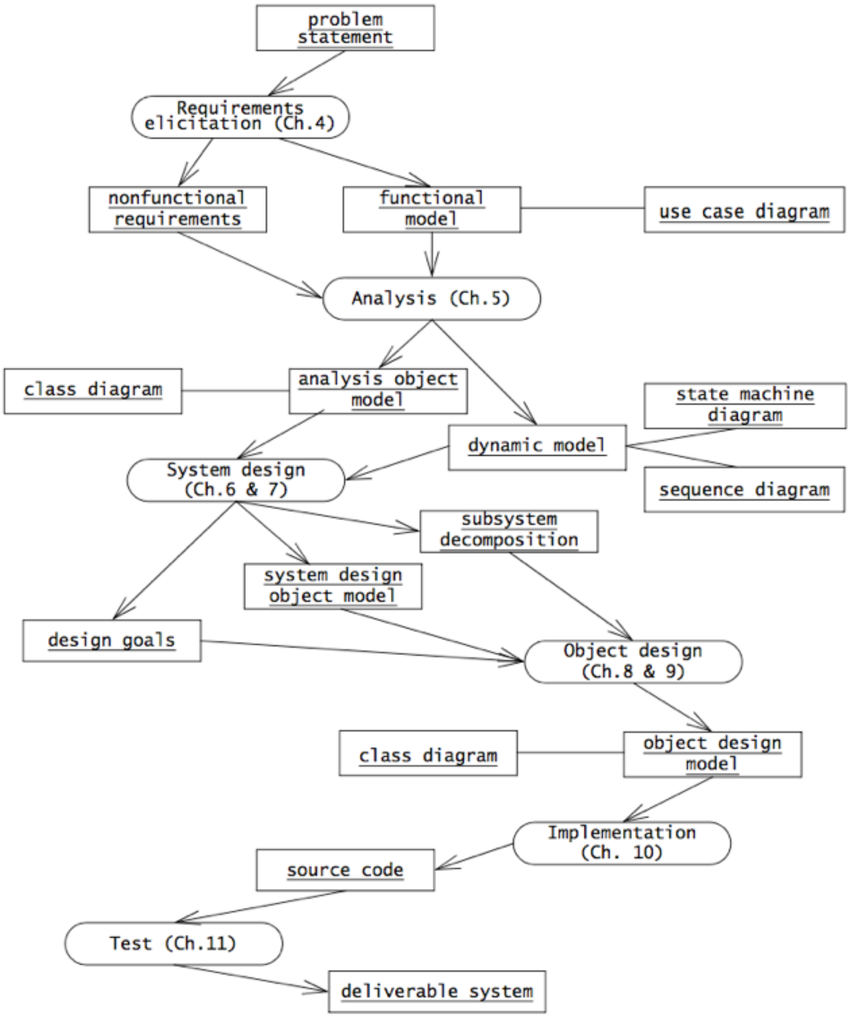


Figure 3.4.: An overview of the object-oriented software engineering development and their products. This diagram depicts only logical dependencies among work products (withdrawn from [5])

4. The problem and its challenges

The multi-material part fabrication is a complex topic and most current commercially available systems have been designed for mono-material part fabrication [6] and are unprepared for multi-material processing due to the lack of flexibility and processing capability.

5. Development

In this chapter, the knowledge acquired through the relevant models contained

The implementation of this thread is presented in listing 5.1.

```
UINT MMSLSDlg::ThreadSerialRx(LPVOID param)
{
    /* Wait for 1st connection to serial port: OnConnect */
    ::WaitForSingleObject( EvSerial.m_hObject , INFINITE);
    tstring szData;
    CDemoEzdDlg *dlg = (CDemoEzdDlg *)param;

    while(1)
    {
        ::WaitForSingleObject( dlg->m_theCommPort.GetWaitForEvent() , INFINITE);
        dlg->m_theCommPort.ReadAvailable (szData);

        if(szData == MSG_CONNECTED) /* Connected */
        {
            dlg->AppendToConsole(_T("Connected!"));
            dlg->AppendToConsole(_T("Initializing...")); /* Resetting the machine */
/* Enable Disconnect and disable connect */
            dlg->m_conn_btn.EnableWindow(false);
            dlg->m_disconn_btn.EnableWindow(true);
            dlg->m_mmsls_conn_status.SetWindowText(_T("Status: Connected"));
            EvPingMach.SetEvent(); /* Set event to stop PingMach thread */
        }
        if(szData == MSG_HOME) /* Home: Reset done */
        {
            dlg->AppendToConsole(_T("Done!"));
            /* Enable Calibration and disable Processing */
            dlg->EnableCalib(true);
            dlg->EnableProcessing(false);
        }
        if(szData == MSG_MANUAL_CALIB) /* Manual Calibration done: Sent by OnCalibDone */
        {
```

```
                dlg->AppendToConsole(_T("Calibration complete!"));
                dlg->AppendToConsole(_T("Ready!"));
                /* Disable Calibration and Enable Processing */
                dlg->EnableCalib(false);
                dlg->EnableProcessing(true);
            }
            if(szData == MSG_RUN) /* Recoating done:  */
                EvRecoatEnd.SetEvent(); /* Signal event to Run */
            if(szData == MSG_IDLE)
            /* Update Laser state */
                dlg->m_mmsls_proc_status.SetWindowText(_T("Status: IDLE"));
            if(szData == EMERG_BYTE) /* Emergency stop activated  */
                EvStop.SetEvent(); /* Signal event to Run */
        }

        return 0;
}
```

Listing 5.1: Thread Serial Rx handler

6. Tests

In this chapter, the LBAM methodology devised was tested and the results

Table 6.1.: MMS machine final specifications

Dimensions (l x w x h)[mm]	560 x 450 x 280
Power supply	Laser: 400 VAC, 10 A
	Machine: 24 V, 15 A
Build dimensions [mm]	50 x 50 x 50
Nr. of materials	2
Temperature	Tested up to 250°C (higher temperatures can be used)
Laser	Type: CO_2
	Power: 30 W
	Spot size: 50 μm
Resolution [μm]	Full-step (all axes, except bed): 5 $\pm$ 0.25
	1/16-step (bed): 0.32 $\pm$ 0.016
Estimated cost [EURO]	Laser: 7500
	Machine: 1500
	Total: 9000

6.1. Summary

In this chapter tests were performed on the workflow, equipment and

7. Conclusion

In this chapter the conclusions and prospect for future work are presented.

Bibliography

[1] Fumio Harashima, Masayoshi Tomizuka, and Toshio Fukuda. Mechatronics-" what is it, why, and how?" an editorial. IEEE/ASME Transactions on Mechatronics, 1(1):1–4, 1996.

[2] Jürgen Gausemeier and Stefan Moehringer. New guideline vdi 2206-a flexible procedure model for the design of mechatronic systems. In DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm, 2003.

[3] Ian Sommerville. Software process models. ACM computing surveys (CSUR), 28(1):269–271, 1996.

[4] Michael A Cusumano and Stanley A Smith. Beyond the waterfall: Software development at microsoft. 1995.

[5] Bernd Bruegge and Allen H Dutoit. Object-Oriented Software Engineering Using UML, Patterns and Java-(Required), volume 2004. Prentice Hall, 2004.

[6] Terry Wohlers. Wohlers report 2011: Additive manufacturing and 3d printing, state of the Industry, wohlers associates, ft. Collins, CO, 2011.

Appendices

A. Use cases (Detailed description)

In this section, the main use cases are described extensively. As many uses cases are similar, only changing the parameter to which they refer to, the Lighting subsystem was used as an example for the Manage<Parameter> use case, where <Parameter> is Temperature, DoorBell, DateAndTime and SystemNotifications.

Although several notifications from relevant events are to be notified by the Master machine to the User, for brevity purposes only the most relevant one is showcased here, namely NotifyManufEnd.

Table A.1.: Use case LoadGeometryFile

Use case name	**LoadGeometryFile**
Participating actors	Initiated by the User
Flow of events	1. The User selects the load geometry file option. 2. The User selects the geometry file to load from a list. 3. If the file is successfully loaded, the filename is displayed, and the file previewed (include PreviewGeometryFile use case). 4. Otherwise, an error message is displayed to the user.
Entry conditions	The User has started the MMSLS machine control software and has previously generated a valid geometry file.
Exit conditions	The file is successfully loaded, previewed and with the filename displayed or an error message is displayed to the User.
Quality requirements	Feedback must be given to the user within 2 seconds; if files are “heavy”, display an ongoing processing.

Table A.2.: Use case PreviewGeometryFile

Use case name	**PreviewGeometryFile**
Participating actors	Initiated by the User
Flow of events	1. The User loads the geometry file. 2. The geometry file is displayed on the canvas.
Entry conditions	A valid geometry file is loaded into the application.
Exit conditions	The geometry file is displayed on canvas.
Quality requirements	The preview should be resizable to accommodate the various component’s dimensions.

Table A.3.: Use case AssignColorsToLaserParams

Use case name	**AssignColorsToLaserParams**
Participating actors	Initiated by the User
Flow of events	1. The User selects a layer. 2. The User associates a material’s color to its laser’s processing color counterpart. 3. The User can edit the laser’s processing color attributes.
Entry conditions	The file is successfully loaded into the application and the preview is editable.
Exit conditions	The materials’ colors has been assigned to valid laser’s processing color with the respective processing parameters.
Quality requirements	Allow multiple line selection.

Table A.4.: Use case LoadManufFile

Use case name	**LoadManufFile**
Participating actors	Initiated by the User
Flow of events	1. The User selects the load manufacturing file option. 2. The User selects the manufacturing file to load from a list. 3. If the file is successfully loaded, the filename is displayed. 4. Otherwise, an error message is displayed to The User.
Entry conditions	The User has started the MMSLS machine control software and has previously generated a valid manufacturing file.
Exit conditions	The file is successfully loaded and with the filename displayed or an error message is displayed to the User.
Quality requirements	Feedback must be given to the user within 2 seconds; if files are 'heavy', display an ongoing processing.

Table A.5.: Use case ConnectToMach

Use case name	**ConnectToMach**
Participating actors	Initiated by the User
Flow of events	1. The User selects the appropriate connection to the machine from a list. 2. The User selects the Connect to Machine option. 3. If the connection is successful, the connection status is updated to Connected and machine operation options are enabled. 4. Otherwise, an error message is displayed to The User.
Entry conditions	The User has started the MMSLS machine control software and there is physical connection between the Master system and the MMS machine.
Exit conditions	The connection is established between Master system and the MMS machine or an error message is displayed to the user.

Table A.6.: Use case DisconnectToMach

Use case name	**DisconnectToMach**
Participating actors	Initiated by the User
Flow of events	1. The User selects the Disconnect to Machine option. 2. If the disconnection is successful, the connection status is updated to Disconnected and machine operation options are disabled. 3. Otherwise, an error message is displayed to The User.
Entry conditions	A successful connection between the Master system and the MMS machine is established.
Exit conditions	The connection is closed and the machine operations options are disabled or an error messaged is displayed to the user

Table A.7.: Use case ManualResetMach

Use case name	**ManualResetMach**
Participating actors	Initiated by the User
Flow of events	1. The User selects the axis to reset, the direction and the reset distance. 2. When the reset is satisfactory, The User acknowledges this fact by selecting the option Reset end. 3. The option to Start Manufacturing is now enabled.
Entry conditions	A successful connection between the Master system and the MMS machine is established.
Exit conditions	The reset is satisfactory (User has selected option Reset End) and the Start Manufacturing is enabled.
Quality requirements	Provide feedback to the User of the manual reset operations (ascent/descent of the axis)

Table A.8.: Use case StartManuf

Use case name	**StartManuf**
Participating actors	Initiated by the user
Flow of events	1. The User selects the Start manufacturing option. 2. If successful, the machine initiates the manufacturing process the machine status is updated to Run.
Entry conditions	1. A successful connection between the Master system and the MMS machine is established. 2. Reset is finished. 3. Valid geometry and manufacturing file have been loaded.
Exit conditions	• Success: The MMS machine status is updated to Run, the Start manufacturing option is disabled, and the options Pause manufacturing and Stop manufacturing are enabled. • Fail: An error message is displayed to the user.
Quality requirements	Update the relevant processing information to the User (include use case UpdateInfo).

Table A.9.: Use case PauseManuf

Use case name	**PauseManuf**
Participating actors	Initiated by the user
Flow of events	1. The User selects the Pause manufacturing option. 2. If successful, the manufacturing process is paused.
Entry conditions	The manufacturing has started (startManuf was triggered), but has not yet finished.
Exit conditions	• Success: The manufacturing process is paused, and the MMS machine status is updated to Idle. The Pause manufacturing option is disabled and the Start manufacturing option is re-enabled. • Fail: An error message is displayed to the user.

Table A.10.: Use case StopManuf

Use case name	**StopManuf**
Participating actors	Initiated by the User
Flow of events	1. The User selects the Stop manufacturing option. 2. If successful, the manufacturing process is stopped.
Entry conditions	The manufacturing has started (startManuf was triggered), but has not yet finished.
Exit conditions	• Success: The manufacturing process is stopped, and the MMS machine status is updated to Stopped. The Stop manufacturing option is disabled and the Start manufacturing option is re-enabled. • Fail: An error message is displayed to the user.

Table A.11.: Use case NotifyManufEnd

Use case name	**NotifyManufEnd**
Participating actors	Initiated by MMS-Mach; User participates
Flow of events	1. When manufacturing is completed, the Master system notifies The User about this fact.
Entry conditions	The manufacturing has started (startManuf was triggered), and is not paused or stopped.
Exit conditions	The manufacturing process stops and the User is notified about this fact.

Table A.12.: Use case VisualizeManuf

Use case name	**VisualizeManuf**
Participating actors	Initiated by MMS-Mach or Laser; User participates
Flow of events	1. Relevant manufacturing information is sent by the MMS-Mach or the Laser to the Master system — on a time-basis or triggered by a relevant event like the manufacturing completion of one layer — that is updated to the User.
Entry conditions	The manufacturing has started (startManuf was triggered)
Exit conditions	The manufacturing is stopped or has ended.
Quality requirements	• Time-basis The relevant manufacturing information must be updated with 1 second refresh rate. • Event The information associated with the triggering event must be updated immediately.

B. Sequence Diagrams

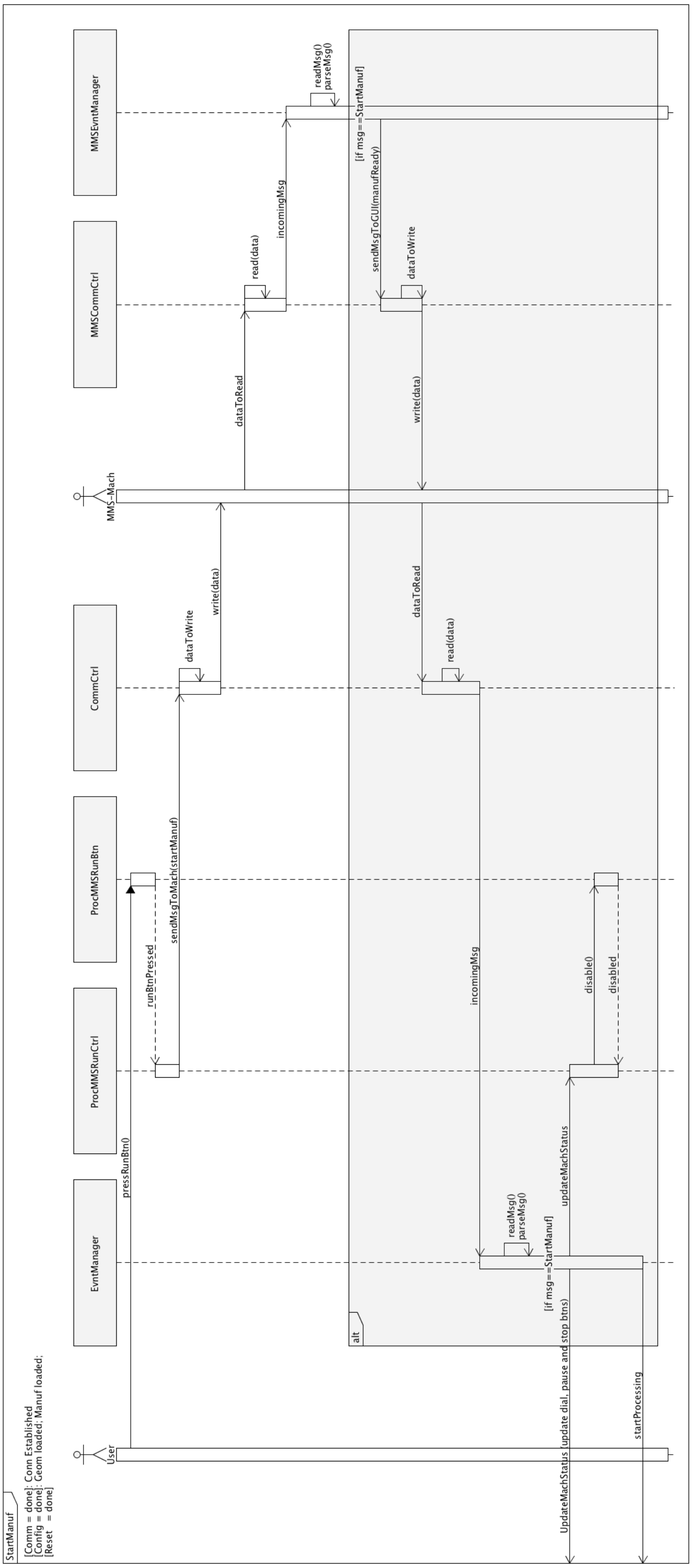


Figure B.1.: Sequence diagram for the StartManuf use case (Part 1)

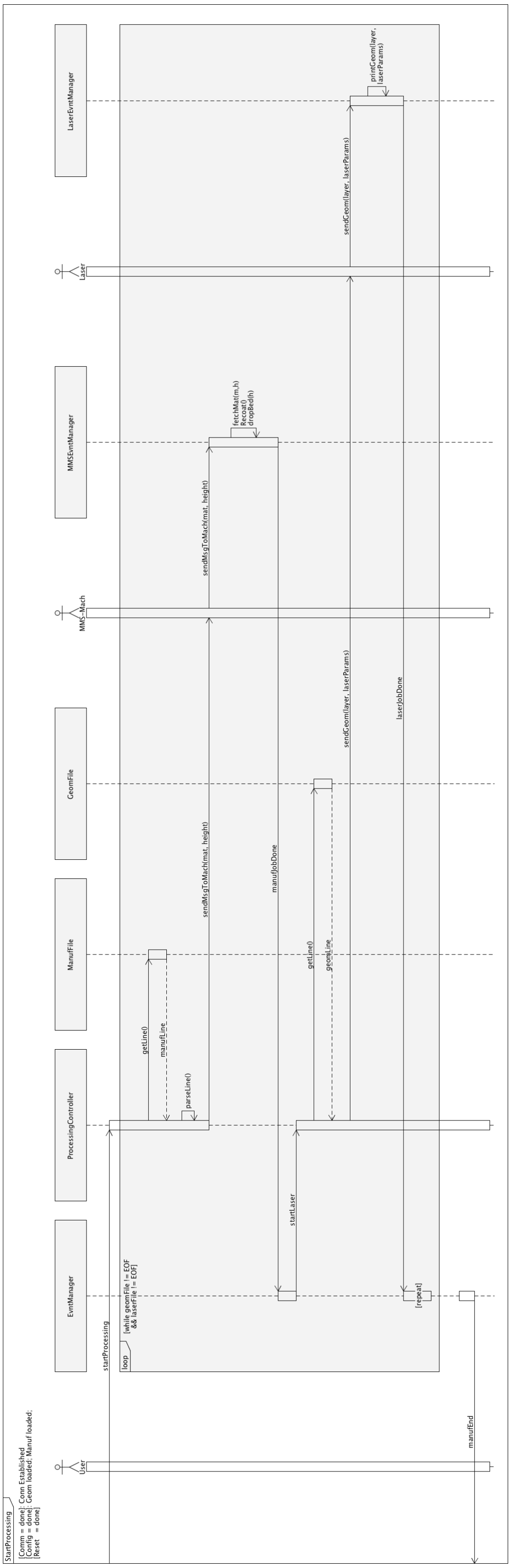


Figure B.2.: Sequence diagram for the StartManuf use case (Part 2)

The author gratefully acknowledges the financial support of this project through the research projects NORTE-01-0145-FEDER-000018 (HAMaBICo — Hybrid Additive Manufacturing for Bio Inspired Component) and POCI-01-0145-FEDER-030498 (FunImp — Design and Fabrication of MultiFunctional Implants).