Universal API for 3D Printers

Felix Baumann[1], Oliver Kopp[2] and Dieter Roller[3]

Abstract: With this research we propose the implementation of a overlay restful API for 3D printers to expose these machines to the Internet for utilization within cloud services. This is to abstract the underlying communication structure and means for accessing and controlling a 3D printer resource which is performed in one of three ways. The first method of accessing and controlling a 3D printer is via a proprietary protocol or a printer driver in Microsoft Windows. The second method is the control via a USB-serial connection between a controlling computer and the printer resource. This protocol can either be proprietary or based on open standards like GCODE (ISO 6983-1:2009). The third method of control is based on physical storage devices attached to the printer with machining instructions stored on them. This research excludes the communication and control means involving proprietary protocols or drivers due complexity restrictions within the implementation. The approach is designed with extensibility in mind so that future access to proprietary protocols can be added to the control API. Printer resources with only the third control method available are also excluded from this research as they are currently lacking the capability to be remotely controlled. This work describes the design and implementation of an abstraction API layer between varying soft- and hardware components with an extensible architecture for future hard- and software components for within the domain of Additive Manufacturing (AM).

Keywords: 3D Printing, Additive Manufacturing, Cloud Based Service, Cloud Based Manufacturing, CBM, CPS, Abstraction Layer, API

1 Introduction

Additive Manufacturing (AM) denotes a method to fabricate physical objects from digital models by means of layer-wise deposition or curing of a material [KMD00, GL13]. It originates from the technology of Rapid Prototyping (RP) but is no longer restricted to just the creation of prototypes. Its strengths are the creation of objects without specialized tools other than a 3D printer, the possibility to produce objects not restricted by the machining tool geometry, i. e., to allow for design for function, no tool creation beforehand necessary and reduced material waste as the process is additive and not subtractive resulting in fewer material necessary. On the downside the price per object created is much higher than with mass produced items, i. e., it is primarily suitable for low volume series, the quality of additively manufactured objects may vary and be lower than with mass-produced or

[1] University of Stuttgart, Institute of Computer-aided Product Development Systems, Universitätsstr. 38, 70569, Stuttgart, baumann@informatik.uni-stuttgart.de
[2] University of Stuttgart, Institute for Parallel and Distributed Systems, Universitätsstr. 38, 70569, Stuttgart, kopp@informatik.uni-stuttgart.de
[3] University of Stuttgart, Institute of Computer-aided Product Development Systems, Universitätsstr. 38, 70569, Stuttgart, roller@informatik.uni-stuttgart.de

custom machined items, depending on the used technology and the embeddability within the manufacturing process is not sufficiently researched and adopted in comparison to traditional machining methods. AM surmises different technologies such as Fused Deposition Modeling (FDM, or also Fused Filament Fabrication), (Selective) Laser Melting or Sintering (SLM/SLS), Electron Beam Melting (EBM), 3D Printing (3DP), Laminated Object Manufacturing (LOM), or Stereolitography (SLA). Classification schemes are proposed by ASTM F42 working group and ISO/ASTM 52900:2015 but without consensus [GRS15]. The ASTM classification schema distinguishes the technologies by means of material processed. A common technology for consumer grade 3D printers is FDM/FLM where semi-molten thermoplastics are extruded through a nozzle mounted on a gantry system as a print-head movable in two dimensions (X-Y plane). The thermoplastics commonly used are Acrylonitrile Butadiene Styrene (ABC) or Polylactic Acid (PLA). This material is most often supplied as a uni-color filament on spools. The material can also be supplied as granulate but no commercially available systems for consumers exist. The material is pushed into the print-head by metal gears driven by an electro-motor and are heated in the print head to a temperature of approximately 230-245 Degrees Celsius for ABS which is over the materials glass-transition temperature [Bo14]. Pushing additional filament into the print-head extrudes material through the nozzle onto the print-bed or an existing layer of extrudate. After the completion of one layer either the print-bed lowers or the print head moves upwards by a specified distance. The accuracy and precision of the involved electro-motors and gears determine the objects quality and printing duration.

Fig. 1 presents a process for creating an object from a digital model. Following 5 steps are performed:

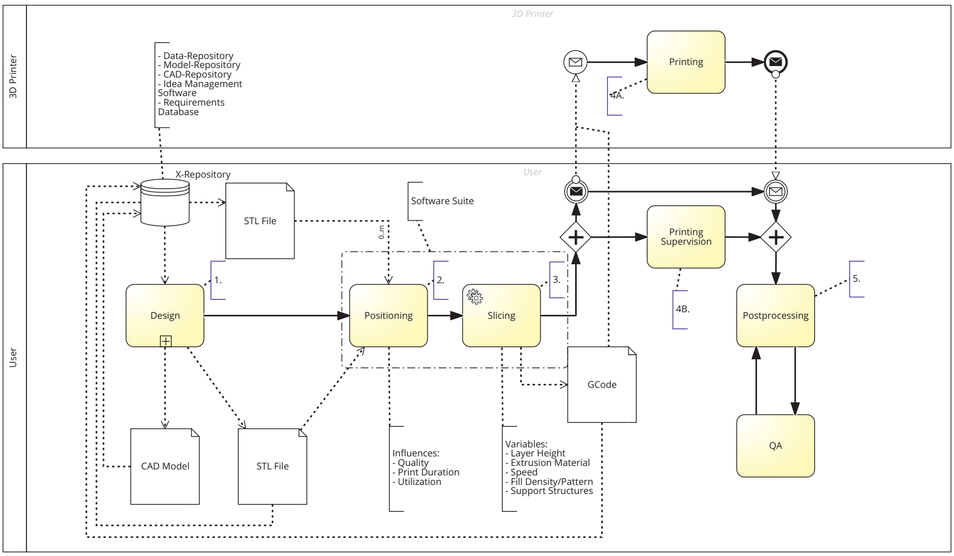

Fig. 1: 3D Printing Process

The process starts with the design of the product [KMD00], which can differ from the design of a subtractive or mass-produced object.

Step two of this process is the positioning of the model in the virtual space that represents the 3D printer and its physical restrictions. Positioning can encompass single objects or multiple objects for increased printer utilization.

After the print object is positioned it is sliced using slicer software. A variety of slicing software exists and they differ in aspects like speed, precision, quality and strategies for printing support structures. The following steps include the upload to the printer if it is a networked device or other means like deployment on memory devices (e. g., SD-Card, USB Stick) and the start of the print which can either require manual interaction or be handled from software. During printing the user is often required to supervise the printing progress as this is error prone especially for consumer grade devices [GS14].

Post-processing and Quality Assurance (QA) follow when the object has been printed and influence each other. These steps involve different hardware (e. g., different printer models) and different software (e. g., proprietary or standardized protocols to communicate with or control of the printer, proprietary or open-source software to transform a digital model from a specific input type to a specific output type or proprietary or open-source software to position and slice the digital model for fabrication). With this research we propose a unifying API that abstracts the controls of the involved soft- and hardware thus enabling a user to communicate with a variety of 3D printers in an uniform way. The proposed API also enables the use of the AM resources as a service. Additive Manufacturing machinery differs in its capabilities regarding material procession and means of communication with and controlling of the machine. A commonly utilized instruction set of 3D printers is GCODE (ISO 6983-1:2009) which is a text and line based machine code with machine instructions identified by a single character code symbol and an opcode. Instrumenting the machine with a pre-computed toolpath in this format differs widely across different printers [ALI12]. The machine code file can either be uploaded to an internal storage device via a protocol over cable (e. g., USB serial communication), manually transferred to a removable storage device, or transferred to the machine ad-hoc through a network or direct-link connection using an open-source or standardized protocol or proprietary means like a device driver. Figure 2 presents an overview over the different methods of transferring information or machine code to a 3D printer. This work is an extension to [BR16] and presents a universal API for 3D printers.

The remainder of this paper is organized as follows: We display current research in this area in Sect. 2 and derive implementation requirements from established approaches. The functionality of the universal API is described in Sect. 3. It is followed by a short discussion of how to add new printers in Sect. 4. An example usage of the API is outlined in Sect. 5. Section 6 presents the implementation and the usage of the API an a real world setting. Then we introduce an example and discuss problems encountered with the implementation. Finally, in Sect. 7, we conclude the paper by discussing our approach, its application, and benefits.

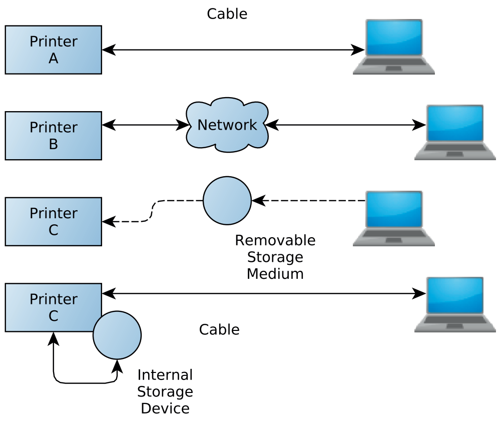

Fig. 2: Communication Variety to Transfer Data to 3D Printers

2 Related Work

Similar systems or services already exist in form of closed source commercial services. Examples are 3D Hubs[3] and 3D Printer OS[4]. Being commercial entities their focus is on financial viability. These services allow adding ones own 3D printer and manage it from within the service with a varying degree of granularity. They lack an extension mechanism or plug-in architecture. In contrast to our approach, they are not intended as open services.

The software octoprint[5] offers remote printing and object management capabilities but does not provide an interface to a business process management systems, user-selectable slicing solutions or support for consolidated information on printing information. Despite these shortcomings we regard this software as a viable possibility for connecting printing hardware to our proposed service.

Further research provides proposals by Wu et al. [Wu15] for CBM systems. Our system differs from those approaches as our focus is the tight integration of business process management (BPM) and 3D printing as well as the sensory upgrade of this technology. From Dong et al. [Do08] we will implement the video supervision approach for the printing process and its remote error detection. Extensions of CBM in the form of Cloud Based Design and Manufacturing [Wu12] provide further insight into the concept of Hardware-as-a-Service (HaaS) and the connection to the broader concept of flexible manufacturing spanning every phase of product development and involvement of different stakeholders. While the availability of affordable consumer grade 3D printers certainly has helped the progression of research in and distribution of 3D printers the scenario where every individual will own a digital fabricator [Mo11] is debatable as the general direction is to offer and consume services [Ba11]. Van Moergestel et al. [Mo15] proved the concept of

[3] `http://www.3dhubs.com`
[4] `http://www.3dprinteros.com`
[5] `http://octoprint.org`

Manufacturing-as-a-Service (MaaS) on cheap, distributed and reconfigurable production machines (equiplets) with a focus on interaction in a multi-agent system.

Kubler et al. [Ku15] provide a theoretical background on the relationship between the concepts of Internet of Things (IoT) and Cloud Manufacturing. With this work they connect concepts from cloud service to the manufacturing domain. Vukovic [Vu15] researches on the role and importance of APIs within the IoT paradigm and with that on cloud manufacturing. This work gives justification to our research on the definition of a API for 3D printing service.

Our work is similar to the work by Harrer et al. [Ha14], but they target the management layer of BPEL engines, whereas we target the management layer of 3D printers.

3 The Universal API for 3D Printers

The goal of the API is to abstract a variety of required software and different hardware in a uniform way in order to be consumed by other services (See Fig. 3). An exemplary list of software and input parameters for it is listed below:

- Positioning Software – Software that positions and orients a model representation within the build envelope of the 3D printer. The positioning and orientation can influence the quality of the printed object, the utilization the printer (e. g., multi object print versus single object print) and the material required for a print (e. g., scaffolding material). Exemplary list of parameters required for abstraction:
  - Position of the object within the build envelope
  - Dimensions of the build envelope
  - Information on possible other objects to be printed concurrently
- Slicing Software – Software that transforms a model representation, e. g., a STL or AMF file, into machine code (e. g., GCODE) to be consumed by the printer. The transformation is performed by a slice-wise segmentation of the original model depending upon a list of user selected parameters e. g., slice height or machine speeds. Exemplary list of parameters required for abstraction:
  - Slice height
  - Machine speed
  - Requirements for scaffolding
  - Slicing strategies
- Control Software – Software that transfers pre-programmed machine code to the printer or instructs the printer to start, stop or pause a print. Furthermore, maintenance operations can be executed through the control software.

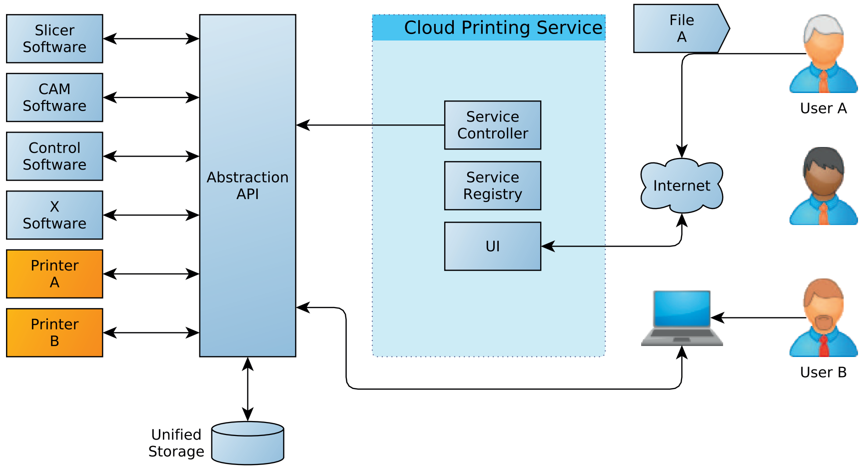

Fig. 3: Overview of the Implemented API

For the API we identified the following six object groups as sufficient for an proof of concept implementation.

- User
- File
- Printer
- Software
- Model
- Print

With the user object operations on users can be executed, which is necessary for accountability. Operations on user objects include:

- Create
- Update
- Delete
- Display information on user
- Associate user with print
- Associate user with printer (e. g., owner or administrator)

With the file object operations on uploaded and stored files can be performed:

- Upload
- Delete
- Update
- Display information on file
- Transform from file format A to file format B
- Associate file with printer (e. g., configuration or description file)
- Associate file with model (e. g., model is described by file)

With the printer object operations on a printer can be performed. Associated operations include:

- Install printer within the system/service
- Mark as available / unavailable
- Remove
- Display information on printer
- Associate printer with print
- Start / Stop / Pause print
- Rate printer quality / reliability / availability or other defined attributes

With the software object operations on software can be performed. The software is installed within the service by an administrator and mappings to the abstracted parameters are defined by the administrator. Usage restrictions for users or file types is defined and persistently stored within the service. In the concept of a 3D printing service the software execution is performed by a scheduler (Fig. 4) that dynamically allocates computing resources necessary for the completion of software tasks:

- Install software within the system / service
- Mark as deprecated
- Remove
- Display information on software
- Execute software

Figure 4 presents an overview of the intended architecture with a business process management system (BPMS) supporting the main service controller.

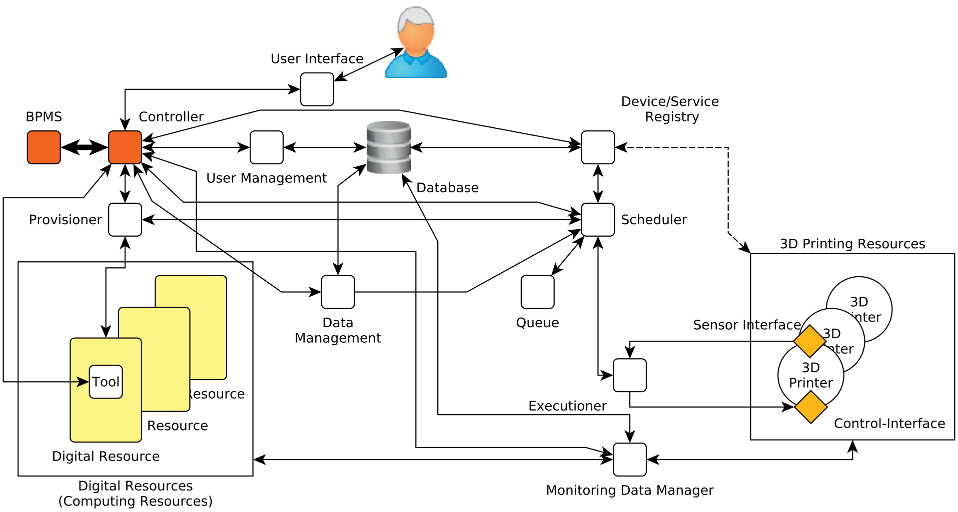

Fig. 4: Architecture for 3D Printing Service

4 Adding Support for new 3D Printers

For integration of various 3D printer types it is necessary to describe their capabilities in an abstract form for re-use. To our knowledge such a description format or language does currently not exist. Resource Description Language (RDL [Sa12]) is a proposition for this issue for the domain of network embedded resources. Capabilities required for interaction with tools includes a) GCode dialect b) Quality settings c) Processing speed and d) Material capabilities. This information is also required for utilization planning and optimization strategies. As a solution for this problem we propose a derivative RDL tailored towards additive manufacturing for subsequent publication. Further problems arise from the firmware of our research printer that limits the transmission speed (ca. 3.5 KiB/s) over the USB connection to the device storage resulting in long transmission times. Solutions include flashing a different firmware and utilizing WiFi enabled SD cards.

5 Example Usage

To clarify the flow of information (Fig. 4) and data within our proposed service we discuss this by an example of a user printing an object. The first process steps of designing and modeling the object with a CAD or modeling tool are not discussed and we assume the user, which already has an account within the service, logs in and has an AutoCAD DXF[6] file stored on his computer. As a first action the file is uploaded through the web-interface to the controller that instructs the data management service to store the file in the database, then the file is transformed into STL and AMF [AS13] format for future use and stored in the database. For this first step the user uploads the provided CAD in the web based UI that then instructs the operations via the provided API endpoint. The user then selects a printer from a list provided through the UI for printing. This list is manually generated in this version as the adaption of printer models requires manual interaction due to the lack of missing abstract hardware description capabilities. This information is provided by the device/service registry. Future implementations can suggest an appropriate printer (based on availability or matching capability for the object) to the user. After selecting the printer the user is able to select slicing parameters and position the object in the virtual build environment. Future implementations can suggest appropriate parameters based on analysis of the model file and positioning on optimization criteria (e. g., strength, build time or utilization) to the user. The user is able to add other tool steps to the processing of the object file which are orchestrated by the provisioner and associated virtual computing resources. These processing steps are limited by the capability of the underlying software package to be instructed without graphical interaction (i. e., headless system, e. g., as libraries or command line tools). After the model file is sliced the printing job is instantiated with the scheduler that checks if the requested printer resource is available and if so sends it to the executioner. If not a queue is used to store the job until the resource becomes available. The executioner communicates with the control interface of the 3D printer in order to transfer

[6] Drawing Interchange Format, `http://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf`

and start the print. Sensor data is transmitted back to the executioner from the sensor interface. Sensor data is then stored in the database via the scheduler and the controller.

During the print the user is informed on the progress and possible failure of the print via web interface and the underlying API for access through other means. After completion of the print the user is informed through a notification by the API or any registered service. Data acquired during the print is stored in the database for later analysis and accessible through the API.

This data includes slicing logs, parameters for slicing and positioning, information on the printer, possible sensor data acquired during print, original files and derivative intermediate transformed files and video or picture data from the print.

6 Evaluation

The proposed API is implemented as a REST API as a proof of concept. We chose REST over Web Services [PZL08], because we aim for integration over the Web. The prototype builds upon the LoopBack[7] framework. This framework is based on node.js[8] and allows for rapid development and deployment of RESTful APIs by utilizing angularJS[9] on the client side, providing persistent storage through a number of services like MySQL[10], MongoDB[11] or other REST services and providing a backend using StrongLoop PM[12]. Through these architectural decisions the implementation satisfies the requirements of scalability, reliability and ease of deployment.

We evaluated it on a Makerbot Replicator 2X[13] and a German RepRap Model X150[14] to assess the capabilities of controlling different 3D printers. For the assessment of the capabilities of multiple software packages versions of Slic3r[15], MiracleGrue[16], FreeCAD[17], GPX[18], and a custom GCODE parse are implemented. The user is able to select a printer from the web based user interface and add processing steps for an uploaded file. The following example list of operations are possible for users to perform:

- Analyze geometry of provided STL file with GPX
- Convert DXF file to STL with FreeCAD
- Slice STL file with Slic3r or MiracleGrue

[7] `http://loopback.io`
[8] `http://nodejs.org`
[9] `http://angularjs.org`
[10] `http://www.mysql.com`
[11] `http://www.mongodb.org`
[12] `http://strong-pm.io`
[13] `http://www.makerbot.com`
[14] `https://www.germanreprap.com/produkte/3d-drucker/x150-3d-drucker`
[15] `http://slic3r.org`
[16] `http://makerbot.github.io/Miracle-Grue`
[17] `http://www.freecadweb.org`
[18] `http://github.com/whpthomas/GPX`

- Position and transpose object in virtual build envelope with user interface
- Upload file from printing service to 3D printers internal memory

The evaluation is performed by a group of 5 students with low to medium experience in the domain of AM. This group is observed performing a set of the following tasks:

- Upload DXF file to print service and convert it to a printable file (GCODE) for a specific printer
- Rotate and rescale an existing file by a set of provided parameters
- Share a model file and a build log with another user of the service
- Analyze and name the failure reason for a build log provided by the service

7 Conclusion and Future Work

To the best of our knowledge no open source 3D printing service or API for it is published yet. There are existing solutions that focus on separate parts and provide solutions to different aspects of the 3D printing process.

Our approach is characterized and differs from other approaches by: a) Focus on 3D printer, b) Focus on communication with manufacturing device, c) Expandable support for future 3D printers and software, d) User customizable selection of processing operations for files associated with the printing process, e) Platform for testing BPMN extension, f) Smartifying 3D printer, g) Platform for testing sensor array, and h) Interchange format for print related information. This 3D printing API and the associated service is designed as an open research platform for academic users to embed experiments and utilize distributed resources. Further projects are aimed at 1. providing means of control of 3D printers from within process models as we are writing an BPMN extension, based on the work by Zor et al. [ZSL11] tailored for 3D printers and 2. utilize sensors for print status observation and as a means for quality research into 3D printing (see ICRM 2016[19]). Those projects are to be incorporated in the umbrella project described in this work. As a related project we develop a BPMN extension for 3D printer integration into BPMN where the hardware resources and data flows can be modeled using the extension.

Acknowledgments

This work is partially funded by the BMWi project “SmartOrchestra” (01MD16001F). We would also like to thank Anna Kulischkin, Marc Tuscher, Jonas Koss, and Niklas Kleinhans for their participation in this work.

[19] `http://icrm-aachen.com`

References

[ALI12] Afsharhanaei, Ali; Leman, Zulkifle; Ismail, N.: Comparison of STEP-NC with ISO6983 in Terms of Manufacturing Interoperability. Applied Mechanics and Materials, 229–231:2556–2561, 2012.

[AS13] ASTM ISO: . ASTM52915-13, Standard Specification for Additive Manufacturing File Format (AMF) Version 1.1. ASTM_International, West Conshohocken, PA, 2013.

[Ba11] Banerjee, Prith; Friedrich, Richard; Bash, Cullen; Goldsack, Patrick; Huberman, Bernardo; Manley, John; Patel, Chandrakant; Ranganathan, Parthasarathy; Veitch, Alistair: Everything as a Service: Powering the New Information Economy. Computer, 44(3):36–43, 2011.

[Bo14] Bonten, Christian: Kunststofftechnik. Carl Hanser Verlag GmbH & Co. KG, 2014.

[BR16] Baumann, Felix; Roller, Dieter: 3D Printing Process Pipeline on the Internet. In: Proceedings of the 8th ZEUS Workshop. pp. 29–36, 2016.

[Do08] Dong, Baoli; Qi, Guoning; Gu, Xinjian; Wei, Xiuting: Web service-oriented manufacturing resource applications for networked product development. Advanced Engineering Informatics, 22(3):282–295, 2008.

[GL13] Guo, Nannan; Leu, Ming C.: Additive manufacturing: technology, applications and research needs. Frontiers of Mechanical Engineering, 8(3):215–243, 2013.

[GRS15] Gibson, Ian; Rosen, David; Stucker, Brent: Additive Manufacturing Technologies. Springer-Verlag New York, 2 edition, 2015.

[GS14] Gribbins, Cassandra; Steinhauer, Heidi M.: Experimental Analysis on an Additively Manufactured ABS Living Hinge. In: Proceedings of the 2016 Annual International Solid Freeform Fabrication Symposium. 2014.

[Ha14] Harrer, Simon; Lenhard, Jörg; Wirtz, Guido; van Lessen, Tammo: Towards Uniform BPEL Engine Management in the Cloud. In: Informatik. pp. 259–270, 2014.

[KMD00] Kulkarni, Prashant; Marsan, Anne; Dutta, Debasish: A review of process planning techniques in layered manufacturing. Rapid Prototyping Journal, 6(1):18–35, 2000.

[Ku15] Kubler, Sylvain; Holmström, Jan; Främling, Kary; Turkama, Petra: Technological Theory of Cloud Manufacturing. Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, 2015.

[Mo11] Mota, Catarina: The Rise of Personal Fabrication. In: C&C. ACM, 2011.

[Mo15] van Moergestel, Leo; Puik, Erik; Telgen, Daniël; Meyer, John-Jules: Implementing Manufacturing as a Service: A Pull-Driven Agent-Based Manufacturing Grid. In: ICTERI. 2015.

[PZL08] Pautasso, Cesare; Zimmermann, Olaf; Leymann, Frank: RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision. In: WWW '08. ACM, 2008.

[Sa12] Santos, AndrÃ© C.; Pedrosa, LuÃs D.; Kuipers, Martijn; Rocha, Rui M.: Resource Description Language: A Unified Description Language for Network Embedded Resources. International Journal of Distributed Sensor Networks, 2012:11, 2012.

[Vu15] Vukovic, Maja: Internet Programmable IoT: On the Role of APIs in IoT: The Internet of Things. Ubiquity, pp. 3:1–3:10, November 2015.

[Wu12] Wu, Dazhong; Thames, J. Lane; Rosen, David W.; Schaefer, Dirk: Towards a Cloud-Based Design and Manufacturing Paradigm: Looking Backward, Looking Forward. In: Computers and Information in Engineering Conference. 2012.

[Wu15] Wu, Dazhong; Rosen, David W.; Wang, Lihui; Schaefer, Dirk: Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. Computer-Aided Design, 59:1–14, 2015.

[ZSL11] Zor, Sema; Schumm, David; Leymann, Frank: A Proposal of BPMN Extensions for the Manufacturing Domain. In: Proceedings of the 44th CIRP International Conference on Manufacturing Systems. 2011.