Cyber-physical System Control via Industrial Protocol OPC UA

Felix W. Baumann[1], Michael Falkenthal[2], Sebastian Hudert[1], Ulrich Odefey[1], and Michael Zimmermann[2]

[1]TWT GmbH Science & Innovation,
Stuttgart, Germany
[firstname].[lastname]@twt-gmbh.de

[2]Institute of Architecture of Application Systems,
University of Stuttgart, Germany,
[lastname]@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings{Baumann2017,
  author    = {Baumann, Felix W. and Odefey, Ulrich and Hudert, Sebastian and
               Falkenthal, Michael and Zimmermann, Michael},
  title     = {Cyber-physical System Control via Industrial Protocol OPC UA},
  booktitle = {Proceedings of the Eleventh International Conference on Advanced
               Engineering Computing and Applications in Sciences},
  year      = {2017},
  pages     = {45--49},
  publisher = {Xpert Publishing Services (XPS)}
}
```

The full version of this publication has been presented at ADVCOMP 2017.
http://www.iaria.org/conferences2017/ADVCOMP17.html

Cyber-physical System Control via Industrial Protocol OPC UA

Felix W. Baumann,
Ulrich Odefey, Sebastian Hudert

TWT GmbH Science & Innovation
70565 Stuttgart, Germany
Email: {firstname}.{lastname}@twt-gmbh.de

Michael Falkenthal,
Michael Zimmermann

Institute of Architecture of Application Systems
University of Stuttgart
70569 Stuttgart, Germany
Email: {lastname}@iaas.uni-stuttgart.de

***Abstract*—The integration of cyber-physical systems (CPS) is gaining more and more momentum due to the advent of Industry 4.0 endeavours. Thereby, one of the main challenges is to cope with the connection to arbitrary machinery in order to monitor and control them automatically to flexibilize production processes by enabling quick adaptions of production steps. Therefore, in this work, a system is described that enables the control of a 3D printer via the industrial standardized M2M communication protocol Open Platform Communications Unified Architecture (OPC UA). The system is implemented on the basis of a micro computing platform, in this case a Raspberry Pi 2, and utilises open-source libraries and tools. The implementation creates a cyber-physical system, consisting of a 3D printer, its control system, sensor data acquisition systems and their respective digital representation. With this control system, the usage of consumer-centric 3D printers, such as Fused Deposition Modeling (FDM) printers, in enterprise-like scenarios is enabled. This abstract and universal control mechanism facilitates research in 3D printing control structures and industrial application.**

Keywords*—*Cyber-Physical Systems, 3D Printer, System Control, OPC UA

I. Introduction

In the industrial domain, especially in current endeavours of Industry 4.0 projects, a common protocol to communicate with machines is the *Open Platform Communications Unified Architecture* (OPC UA). This protocol allows accessing data from machinery in a read and write manner. It is standardised as IEC 62541 [1] and, thus, provides a robust basis for sustainable integration scenarios. 3D printers are machines that create physical objects from digital models by a variety of technologies and materials. Thereby, they can be connected and controlled in a number of ways, such as via (i) USB-serial cable-bound connection, (ii) WiFi or Ethernet network-bound connection, (iii) a controlling computer, or (iv) manually through interaction at a local control panel. Integrated, abstract and coherent means to control such systems are integral to the application and integration of this technology [2]. OPC UA provides a common way of control and the availability of this abstract interface to this 3D printer facilitates research and industrial application.

3D printers are regarded as CPS because they form a system that matches the physical reality, acquired through sensors, with a digital representation of the 3D printer and the object creation.

In Figure 1, the schematic view of the implementation of the system is depicted. This figure provides the layout and connectivity of the discussed system. The application of this control system within the *SePiA.Pro* project [3], as partially described by Falkenthal et al. [4] and Pfeil et al. [5], is designed to allow for research into process structure analysis and improvement.

The challenges for this integration lie in the diversity of control mechanisms for 3D printers and the fact that most 3D printers are not intended for networked operation natively.

Sensorial data can be integrated into the exposed data on the micro computer and extend the information provided through this system. The data acquisition is typically performed over I^2C from a digital sensor, similar to systems described by Baumann et al. [6]. This control system can extend a collaborative 3D printing system, such as described by Baumann, Eichhoff, and Roller [7]. Furthermore, it can be utilised to extend or support a common means of communication in distributed 3D printing systems, such as described with the Application Programming Interface (API) for 3D printing by Baumann, Kopp, and Roller [8]. One use-case is the integration of multiple 3D printers in a demonstrator for scheduling of object creation founded on smart service data analytics from 3D printer sensor and additional sensor data. As an example, the 3D print can be paused to allow for manual interaction in case of sensor reading for the temperature sensor in the printhead exceeding a threshold, which can be indicative of jammed filament.

This work describes the connection of existing control mechanisms for 3D printers, a messaging infrastructure for industrial applications and the extension to further data sinks and sources, such as Web Services.

The remainder of this paper is structured as following: In Section I-A, an introduction to the concepts of 3D printing and Additive Manufacturing is provided. Following in Section I-B, an introduction to the Open Platform Communication Unified

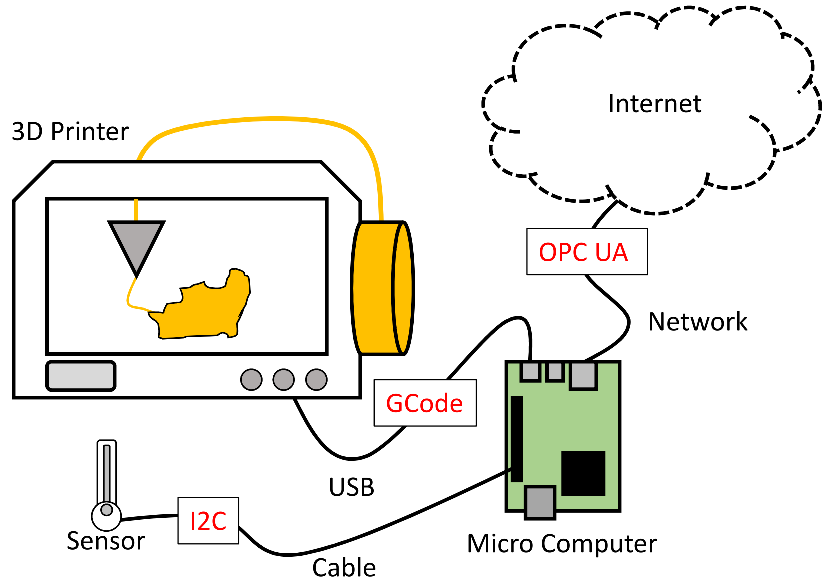

Fig. 1. Schematic View of Implementation

Architecture is presented. In Section II, the work is placed within existing research and the objectives are stated. The Section III details the required and intended capabilities for the proposed system. The main part of this work is located in Section IV, with details on the architecture and design choices. The paper concludes with a summary in Section V.

A. 3D Printing/Additive Manufacturing

The following short overview on 3D printing is provided for the reader to understand its basics concepts and place this implementation. 3D printing and Additive Manufacturing (AM) are commonly used as synonyms. Both terms describe a variety of technologies to create physical objects from digital models without the requirement for specialised tools, except a 3D printer [9]. The resulting objects can be created from a variety of materials, such as plastics, ceramics, and metals, with commonly only one type of material per resulting object. Different kinds of materials require different kinds of 3D printers and underlying technologies. The objects are commonly created by a directed, layer-wise stacking, curing or extrusion of a material [10], [11]. The 3D printing process and the corresponding results can be influenced by a number of environmental and inherent factors, such as vibration, temperature or quality of material. To assess and possibly counteract these influences, sensors are employed to acquire data from the 3D printing process.

B. Open Platform Communications Unified Architecture

Open Platform Communications Unified Architecture (OPC UA) is a platform-independent standard for machine-to-machine (M2M) communication, e.g., between clients and servers on various types of networks [12]. It is intended to facilitate information exchange between machines. Thereby, OPC UA is an extension of the older OPC standards and adds, among others, the ability to semantically describe the transmitted machine data. Due to its fundamental design on the basis of a service oriented architecture [13], OPC UA has been adopted in a number of domains such as industrial machinery, power grids, home or building automation, and smart devices [14]. This way, it facilitates the interoperability of the involved systems and evolves as a multi-standard platform, which it initially was intended and specified for by the OPC Foundation.

II. Related Work

For an overview of existing control mechanisms and architectures for the remote control of AM machinery, we refer to Baumann, Kopp and Roller [15], where a detailed discussion of related work is presented. Concepts such as Cloud Based Manufacturing, Hardware and Manufacturing as a Service are also provided in the overview by Baumann and Roller [16]. In the work by Berteslmeier, Schöne and Trächtle [17], the authors presented a similar control system using OPC UA with focus on intelligent products. Existing control mechanisms for 3D printers lack the ability incorporate physical feedback from the machine and lacks extensibility. With the provided paper, we provide a control mechanism for 3D printers that is based on an established industrial communication protocol, thus enabling the integration of AM machinery into existing software and control ecosystems. This integration requires the presence of a micro computing platform that facilitates networking and translation services to machines that were previously not networked but connected directly to control systems via USB. The utilization of an existing protocol reduces the implementation efforts and enables a low-cost solution for remote access, management and control of these machines. Due to flexibility in the design of the approach, other machines and sensors can be integrated and controlled or used as data sources. With this added control capability, 3D printers can be integrated into smart environments and production systems.

III. Capabilities of the Control System

The following capabilities are already implemented within the control system running on the micro computer as illustrated in Figure 1. The capabilities are specifically justified to cover the basic requirements of operation of a 3D printer.

- Direct control of printhead: The movement of the printhead can be can be controlled by defining movement accuracy, movement speed, as well as x, y, z coordinates.
- Control and monitoring of temperatures and other printer inherent sensors: Current printbed and tool temperatures as well as their historical data can be obtained.
- Upload of files to 3D printer: Files containing the model to print can be uploaded and stored to the 3D printer's SDCard or to the controlling micro controller depending on the user's needs.
- Control of 3D printer operation: General commands to control the operation mode of the 3D printer, such as start, pause and stop operations, can be triggered.
- Status control and monitoring of 3D printer: Inherent 3D printer status information and external sensor data acquisition is implemented and can be obtained.
- File availability information: To manage already uploaded files, the data structure of control system's and the 3D printer's SDCard can be retrieved.

These features are implemented in the system and described in the following Section IV.

IV. Implementation Basis

The system is implemented utilising the *node-opcua* library [18], available for *NodeJS* [19]. This library provides both, server and client bindings for *JavaScript*. The server component, which exposes the *OPC UA* data from the 3D printer and the system to the client, consists of a *JavaScript* file that implements the required functionality. The system exposes internal properties, such as CPU temperature, CPU utilisation, memory availability, as historical data nodes, thus, enabling clients to subscribe to data change events for this data. Furthermore, external or 3D printer inherent data, such as bed and tool temperature, are exposed as historical data nodes. The sensor data acquisition is performed directly from the *JavaScript* server program, utilising I^2C communication, provided via the *i2c-bus NodeJS* library (see https://www.npmjs.com/package/i2c-bus).

The acquisition of data from and control of the 3D printer is performed by using the *OctoPrint* [20] software. This software is available for the Linux platform and suitable for deployment on micro computers, such as a *Raspberry Pi 2*. The software provides control mechanism for a large number of different 3D printers, mostly over a USB serial connection. The micro computer provides the capabilities to interface the 3D printer via network, as the 3D printer itself does not necessarily provide a networking interface.

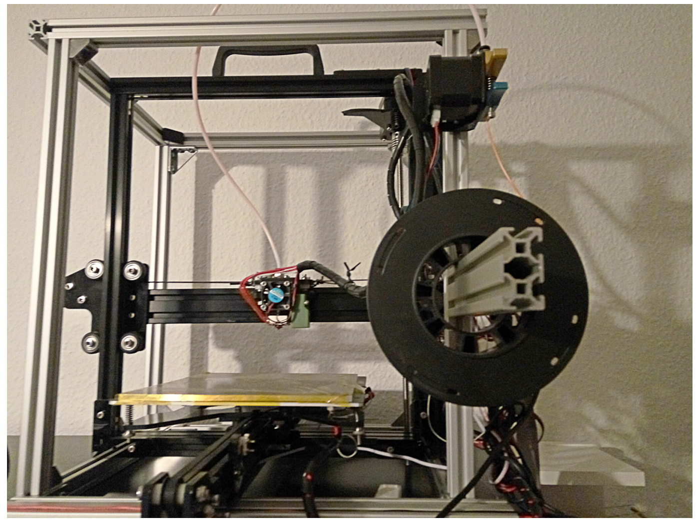

Fig. 2. 3D Printer

The control of a printer is implemented following the GCode protocol, which is a line based control format. GCode is a numerical control programming language standardised as ISO 6983 [21], however, variations in the implementation by the manufacturers exist. For this experiment, a *Tevo Tarantula* [22] 3D printer (see Figure 2) is flashed with the *Marlin* [23] firmware to be used with the *OctoPrint* software. *OctoPrint* exposes all its functionality over a Hypertext Transfer Protocol (HTTP) Representational State Transfer (RESTful) API. This API is utilised to communicate and control the 3D printer via local API calls from the server component. Authorisation in *OctoPrint* is disabled as only local communication is allowed.

The communication flow of the client, the implemented server and the 3D printer are depicted in Figure 3. This figure shows, that the polling of status from the 3D printer and its subsequent processing and dispatch to the user is performed in a loop while the issued command is performed. The *OctoPrint* software does not support Websocket technology for continuous connection to its API. As an example, the command to print a specific file from the SDCard is discussed in the following.

When the user or client issues this command, the system checks with the 3D printer and then actually issues the command on the 3D printer. Thereby, the system must check for connectivity, availability of the SDCard, and the correct status, i.e., no-error, print-ready state. For asynchronous communication over the API, the *request* library (see https://www.npmjs.com/package/request) in conjunction with the *form-data* (see https://github.com/form-data/form-data) library is used.

To provide, direct and synchronous communication to the user, most communication with the API is performed synchronously, using the *sync-request* library (see https://www.npmjs.com/package/sync-request), which blocks the execution of the server while waiting for the communication results and,

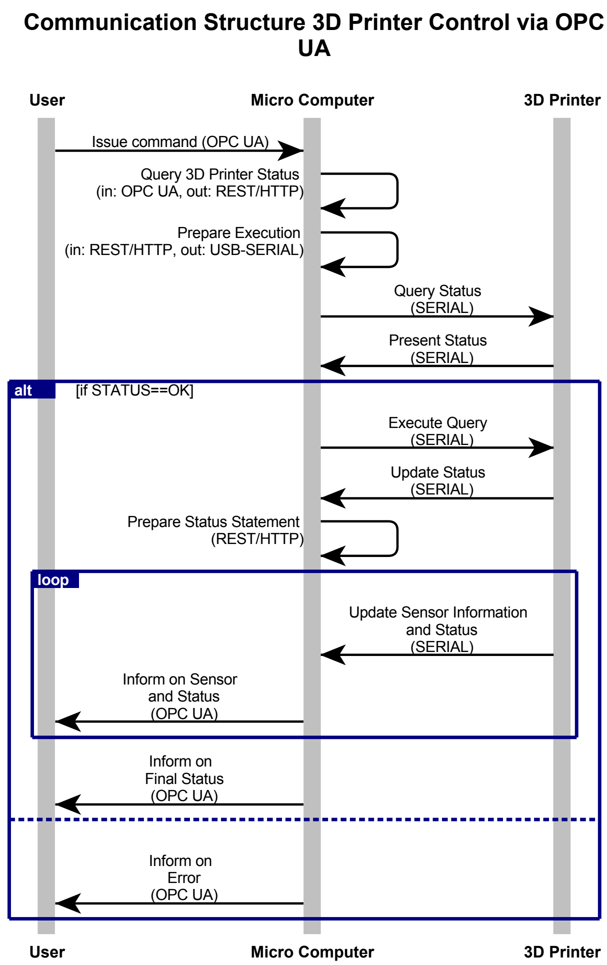

Fig. 3. Communication Structure between User, Client, and 3D Printer

thus, is unsuitable for scalable application deployment. The processing of the status information of the printer, such as print completion rate, 3D printer status, and temperatures is performed by a pull-mechanism that is triggered by the server.

The Figure 4 displays the stacking structure of the components within the system. The basis is the micro computing system, or single board computer, on which a Linux operating system is running. The individual *NodeJS* libraries are displayed at the topmost layer.

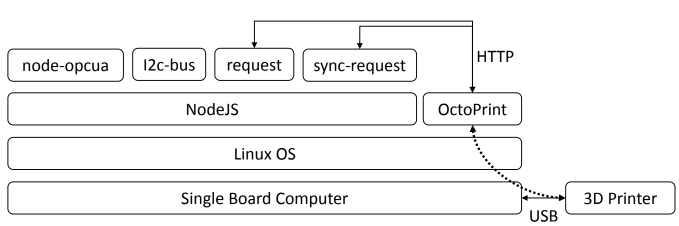

Fig. 4. System Structure

Push notification is possible with *OctoPrint*, but limited to two messages per second. The interfacing of a client with the implemented server is shown as a screenshot in Figure 5. In this figure, three subscribed data nodes and the expanded tree structure of the nodes is visible.

The implementation is available as a file-system image upon request from the authors, pre-configured for use on a *Raspberry Pi 2* system and a *Marlin* firmware based 3D printer, as listed as supported on the *OctoPrint* website.

V. Conclusion

This paper describes the design of the implementation of a control system for a CPS, a 3D printer and its associated sensors. This system is based on the widely used standard *OPC UA* for industrial M2M communication. Through the exposure of sensor and machine data via *OPC UA* the implementation in industrial settings is facilitated. The system developed facilitates the complete control of a variety of 3D printers using *OPC UA*. It was shown how the control of such a system is enabled by transformation of one common protocol to another common protocol by open-source software. The software is implemented with the described capabilities and the potential for extension in future iterations. Future iterations of this software are planned to allow control and communication over other protocols, such as *MQTT*, see http://mqtt.org and *Woopsa*, see http://www.woopsa.org. Furthermore, data source integration using asynchronous connections, such as Websockets are to be integrated.

Acknowledgments

This work is partially funded by the project SePiA.Pro (01MD16013F) of the BMWi program Smart Service World.

References

[1] TC 65/SC 65E, "IEC 62541-100:2015 OPC Unified Architecture - Part 100: Device Interface," International Standard, International Electrotechnical Commission, IEC 62541-100, 3 2015. [Online]. Available: https://webstore.iec.ch/publication/21987

[2] Y. Lu, K. C. Morris, and S. P. Frechette, "Current Standards Landscape for Smart Manufacturing Systems," National Institute for Standards and Technology (NIST), NISTIR 8107, 2016. [Online]. Available: https://dx.doi.org/10.6028/NIST.IR.8107

[3] Deutsches Forschungszentrum für Künstliche Intelligenz *et al.*, "SePiA.Pro – Service Plattform für die intelligente Anlagenoptimierung in der Produktion," last accessed on 6th Sept. 2017. [Online]. Available: http://projekt-sepiapro.de

[4] M. Falkenthal *et al.*, "Requirements for policies in industrial data sharing scenarios," in *Proceedings of the 11th Symposium and Summer School On Service-Oriented Computing*, 2017, in Press.

[5] M. Pfeil *et al.*, "Smart services - the smart implementation of Industry 4.0," in *NAFEMS Seminar Simulation von Composites – Bereit für Industrie 4.0?*, 10 2016. [Online]. Available: https://www.researchgate.net/publication/317687064_Smart_services_-_the_smart_implementation_of_Industry_40

[6] F. W. Baumann *et al.*, "Sensors on 3D Printers for Cloud Printing Service," in *Proceedings of the 2016 International Conference on Sustainable Energy, Environment and Information Engineering*. DEStech Publications, Inc., 3 2016, pp. 571–577. [Online]. Available: http://dx.doi.org/10.12783/dteees/seeie2016/4689

[7] F. W. Baumann, J. Eichhoff, and D. Roller, *Collaborative Cloud Printing Service*. Springer International Publishing, 2016, pp. 77–85. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46771-9_10

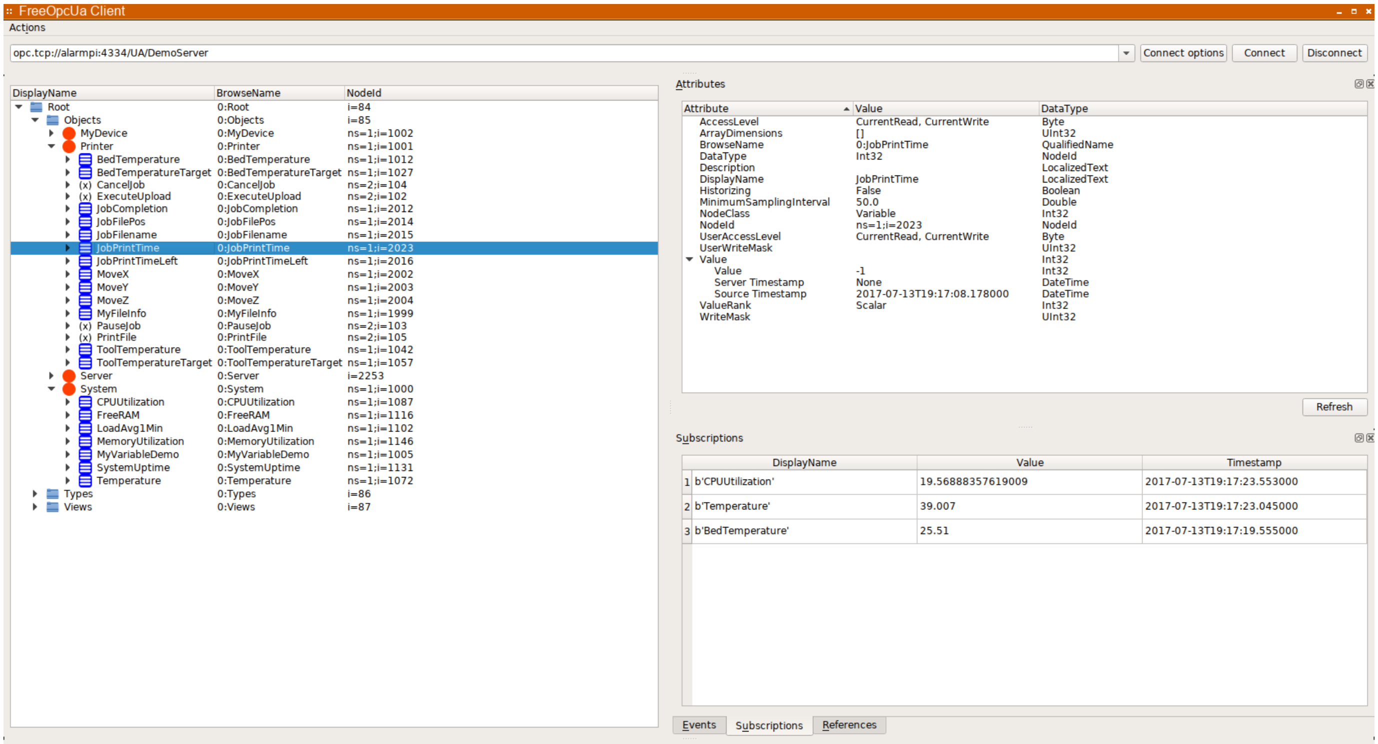

Fig. 5. Screenshot of OPC UA Client

[8] F. W. Baumann, O. Kopp, and D. Roller, "Universal API for 3D Printers," in *INFORMATIK 2016. Lecture Notes in Informatics (LNI)*, H. C. Mayr and M. Pinzger, Eds., vol. P-259. Gesellschaft für Informatik, 2016, pp. 1611–1622. [Online]. Available: http://subs.emis.de/LNI/Proceedings/Proceedings259/article155.html

[9] I. Gibson, D. W. Rosen, and B. Stucker, *Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[10] N. Guo and M. C. Leu, "Additive manufacturing: technology, applications and research needs," *Frontiers of Mechanical Engineering*, vol. 8, no. 3, pp. 215–243, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11465-013-0248-8

[11] P. Kulkarni, A. Marsan, and D. Dutta, "A review of process planning techniques in layered manufacturing," *Rapid Prototyping Journal*, vol. 6, no. 1, pp. 18–35, 2000. [Online]. Available: https://doi.org/10.1108/13552540010309859

[12] F. Palm *et al.*, "open62541 - der offene OPC UA Stack," in *Jahreskolloquium "Kommunikation in der Automation" (KommA)*, 2014, pp. 1–9. [Online]. Available: http://publica.fraunhofer.de/dokumente/N-322220.html

[13] S.-H. Leitner and W. Mahnke, "OPC UA - Service-oriented Architecture for Industrial Applications," *Softwaretechnik-Trends*, vol. 26, 2006.

[14] T. Hadlich, "Providing device integration with opc ua," in *2006 4th IEEE International Conference on Industrial Informatics*, 10 2006, pp. 263–268.

[15] F. W. Baumann, O. Kopp, and D. Roller, "Abstract API for 3D Printing Hardware and Software Resources," *International Journal of Advanced Manufacturing Technology*, pp. 1–17, 2016.

[16] F. W. Baumann and D. Roller, "3D Printing Process Pipeline on the Internet," in *Proceedings of the 8th Central European Workshop on Services and their Composition (ZEUS 2016)*, 2016, pp. 29–36. [Online]. Available: http://ceur-ws.org/Vol-1562/paper4.pdf

[17] F. Bertelsmeier, S. Schöne, and A. Trächtler, "Development and design of intelligent product carriers for flexible networked control of distributed manufacturing processes," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 6 2016, pp. 755–760.

[18] E. Rossignon, "Build OPC UA applications in JavaScript and NodeJS," last accessed on 6th Sept. 2017. [Online]. Available: http://node-opcua.github.io

[19] Node.js Foundation, "Node.js," last accessed on 6th Sept. 2017. [Online]. Available: https://nodejs.org

[20] G. Häußge, "Octoprint - the snappy web interface for your 3d printer," last accessed on 6th Sept. 2017. [Online]. Available: http://octoprint.org

[21] ISO/TC 184/SC 1 Physical device control, "ISO 6983-1:2009 Automation systems and integration – Numerical control of machines – Program format and definitions of address words – Part 1: Data format for positioning, line motion and contouring control systems," International Standard, International Organization for Standardization, ISO 6983-1, 12 2009. [Online]. Available: https://www.iso.org/standard/34608.html

[22] J. Choo, "Marlin Firmware for Tevo Tarantula 3d Printer," last accessed on 6th Sept. 2017. [Online]. Available: https://github.com/josephchoo/TevoTarantulaMarlin

[23] Marlin dev team, "Optimized firmware for RepRap 3D printers based on the Arduino platform," last accessed on 6th Sept. 2017. [Online]. Available: https://github.com/MarlinFirmware/Marlin