A Distributed Software Platform for Additive Manufacturing

Rafael Natalio Fontana Crespo*, Davide Cannizzaro*, Lorenzo Bottaccioli*,
Enrico Macii*, Edoardo Patti* and Santa Di Cataldo*
*Politecnico di Torino, Turin, Italy. Emails: name.surname@polito.it

***Abstract*—Additive Manufacturing (AM), a cornerstone of Industry 4.0, is expected to revolutionise production in practically all industries. However, multiple production challenges still exist, preventing its diffusion. In recent years, Machine Learning algorithms have been employed to overcome these hurdles. Nonetheless, the usage of these algorithms is constrained by the scarcity of data together with the challenges associated with accessing and integrating the information generated during the AM pipeline. In this work, we present a vendor-agnostic platform for AM that enables collecting, storing, analysing and linking the heterogeneous data of the complete AM process. We conducted an extensive analysis of the different AM datatypes and identified the most suitable technologies for storing them. Furthermore, we performed an in-depth study of the requirements of different AM stakeholders to develop a rich and intuitive Graphical User Interface. We showcased the specific usage of the platform for Powder Bed Fusion, one of the most popular AM processes, in a real industrial scenario, integrating specific existing modules for in-situ monitoring and real-time defect detection.**

***Index Terms*—Industry 4.0, Platform, Additive Manufacturing, Powder Bed Fusion, Machine Learning**

I. Introduction

Additive Manufacturing (AM) is the process of producing an object by layering raw materials one on top of another employing one of the various 3D printing processes. AM is opposite to the traditional subtractive processes, in which an object is subjected to multiple machining procedures reducing its volume [1]. The AM production process starts with a computer-aided design (CAD) file, containing the model of the desired object to be produced. The CAD file is then transformed into a Stereo Lithography (STL) file. This file is transferred to the machine, where it is scanned, sliced into layers, and constructed. Once the production process is completed, the object is detached from the building plate and post-processing is carried out to achieve the desired properties.

AM has shown to have significant advantages over subtractive processes in terms of production procedures, development time, cost and material usage as well as considerably greater versatility. However, there is still a significant gap between AM current state and the maturity required for its widespread adoption due to production hurdles such as process repeatability and reliability [2]. The recent usage of powerful machine learning (ML) algorithms to support real-time process control and fault detection during builds is helping to overcome these limiting factors. Nonetheless, these algorithms require data.

During an AM building process, a vast amount of heterogeneous data are generated. To comprehensively characterize the process, valuable information must include more than just the data collected from various sensors and cameras mounted on the machine. Among others, these data include machine settings, CAD and STL files. However, these data may have varying formats. This, together with the absence of a standardized data structure and Application Programming Interfaces (APIs) for managing information from different sources, creates significant challenges in collecting, storing, synchronizing, and correlating AM data [3]. Due to these challenges, AM data are generally not collected, limiting the application of the ML algorithms previously mentioned [4].

The development of platform solutions able to collect and manage AM data is attracting increasing attention in the research community. In [5], a cloud-based framework combining digital-twins technology with a collaborative data management was proposed. Data were stored in clouds organized by an AM life-cycle data schema. What is more, the platform enabled the integration of other applications using the Representational State Transfer (REST) [6] interface. On the other hand, in [7] a platform to store in a scalable manner the huge amount of data generated during the AM process was developed. The platform performed analysis and presented the information through customizable visualizations. To the best of our knowledge, the aforementioned platforms have a quite limited scope, since they target specific technologies and applications. What is more, according to [8], the framework's Graphical User Interfaces (GUI) are a bit hard to understand for normal non-expert users and the available data visualizations are weakly related to the general industry requirements. Lastly, the framework proposed in [5] relied on a proprietary license platform - Material Data Curation System (MDCS).

To deal with the specific problem of AM heterogeneous data integration, storage and management, in this work we present a distributed software platform which supports all the steps of the AM process. The platform is designed with a microservices design pattern, where discrete services implement specific functionalities and can be created, deployed and scaled independently. This paper builds upon our previous concept presented in [9] and presents the practical implementation of the proposed microservices-based AM monitoring platform, which includes significant improvements. Firstly, we perform an extensive analysis of the heterogeneous AM datatypes, which enables the selection of the most suitable technologies for their storage. What is more, we develop a rich GUI following a series of specifications identified through an in-depth study of the requirements of different AM stakeholders. The developed GUI enables non-expert users to intuitively interact with the platform functionalities and manage AM-related data. Lastly, we showcase a particular implementation of the platform in a real industrial scenario for Powder Bed Fusion (PBF), one of the most prevalent techniques for Metal Additive Manufacturing. To this end, the platform integrates a plugin component for the detection of defects generated during the PBF process which was presented in [10]. Platform users can easily execute this defect detection analysis as well as visualize the results in a layer-by-layer fashion or through aggregated statistics for the complete object.

II. Platform Description

In this section, we present the developed AM platform for data collection and management, which schema is shown in Fig. 1. Firstly, we analyse the data the platform should store and manage. Later, we provide an overview of its different microservices.

A. Data Type and Structure

Before starting the platform development, it is crucial to comprehend the origin, structure, and type of the data the platform will handle. To this end, an entire AM lifecycle process was analysed. Data were grouped into five distinct categories: i) *Design parameters* include all the parameters that are specified during the product design, such as CAD files, raw material and support shapes; ii) *Process parameters* refer to the input settings for the AM process established during machine setup, which may be predefined or set by the machine operator (e.g. type of laser, powder heat capacity, shield gas used); iii) *Process characteristics* involve the data that can be used to characterise the building process, either as-is or derived from sensors mounted on the machine, such as temperature measurements and layer surface topography; iv) *Post-Processing* includes the different post-processing methods and their corresponding parameters; v) *Product analysis* encompasses the outcomes of the tests done on the AM fabricated part to determine its final properties (e.g. surface roughness and porosity).

These categories encompass a variety of data types, such as text, images, reports, CADs and other file types, which must be handled appropriately to enable the storage and exchange of these data between the different stakeholders.

B. Architecture and Microservices

We aim to develop a platform based on the microservices design pattern that guarantees: i) *scalability*; ii) *reliability*; iii) *flexibility*; iv) *modularity*; v) *interoperability*; vi) *extendibility*; vii) *decentralization*; viii) *standarization*; ix) *security*; x) *synchronous communication*; xi) *asynchronous communication*. Additionally, the platform must implement suitable technologies to store the heterogeneous AM datatypes.

For this reason, we developed an AM platform, which architecture is shown in Fig. 1. The platform is built with the microservices approach, and comprises the whole AM data lifecycle. Moreover, the framework provides REST APIs for external and internal communications. The platform design enables the integration of new functionalities (e.g. defect detection) in a plug-and-play fashion, which can communicate with the other components using the Message Queuing Telemetry Transfer (MQTT) or the REST protocol, thus providing both Publish/Subscribe and Request/Response communication paradigms. Last but not least, it implements modern security approaches, such as authentication using JsonWebTokens (JWT).

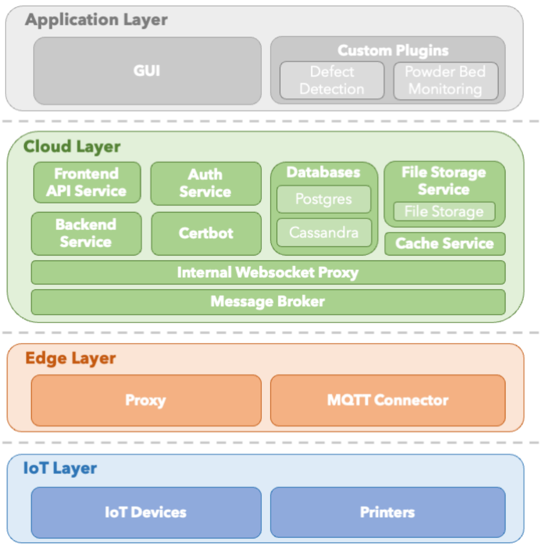

Fig. 1. AM data collection and management platform: proposed architecture

As shown in Fig. 1, the platform architecture consists of four layers: i) *IoT Layer*, ii) *Edge Layer*, iii) *Cloud Layer*, and iv) *Application Layer*. The *IoT Layer* comprises the devices in the field, such as the *printers* and other *IoT devices* involved in the AM process (e.g. sensors and actuators). On the other hand, the *Edge Layer* includes the *Proxy* and the *MQTT Connector*. The *Proxy* handles synchronous (REST) communication between the ”field” devices and the platform components while the *MQTT Connector* manages the asynchronous (MQTT) communication.

The *Cloud Layer* is the platform’s core and comprises several components which provide different functionalities. The *Message Broker* is responsible for handling asynchronous (MQTT) communication between the platform components. On the contrary, the *Internal Web Socket Proxy* manages synchronous (REST) communication between the platform components and balances loads between them. The *Auth Service* manages user and device authentication using JWT tokens. Furthermore, to guarantee secure external connections, *Certbot* automates the generation of certificates enabling the use of the Hypertext Transfer Protocol Secure (HTTPS). On the other hand, the *Frontend API Service* performs primary checks, submitting requests to the *Backend Service* and returning asynchronous responses from it. The *Backend service* is responsible for handling internally the available devices and databases, and transmitting data to and from the *Databases* and the *Cache Service*. The *Cache Service* stores *Process* *characteristics* time-series data, enabling immediate access to this information. Regarding the *Databases*, two are employed: i) *PostgreSQL*, a relational database, is used to store non-time series data. This includes data of the five categories described previously in Section II-A (e.g. raw material, type of laser, and porosity); ii) *Cassandra DB*, a non-relational database, is employed to store time-series data. This comprises mainly the *Process characteristics* data, such as temperature and oxygen concentration. Lastly, the *File Storage* saves file data types, including files of the *Design Parameters* (e.g. STL files), *Process Characteristics* (e.g. machine quality report), and *Product Analysis* (e.g. porosity report, mechanical report).

The *File Storage System* is the microservice responsible for handling files. This microservice exposes REST APIs for sending, retrieving and deleting files in the *File Storage*. What is more, it provides a REST API for downloading a *PDF* file which combines the data of a job and its reports (e.g. materials reports, mechanical reports). In addition, it provides a REST API for retrieving a *zip* file to download all the files related to a job. Since this file may have a considerable size (e.g. defect detection monitoring involves hundreds of images, in order of GBs), the file compression is performed "on the fly" to provide better performance in terms of time, memory, and user experience. Furthermore, the *File Storage System* integrates a tool for combining STL files and correlating the digital STL model with the images obtained on a layer basis. This may seem trivial for the latest AM machines with this technology already available, but it is significantly useful for older ones allowing back compatibility. Moreover, it enables a 3D layer-by-layer visualization of the printing phase in the *GUI*.

Last but not least, the *Application Layer* provides a set of tools to enable users to manage and access information coming from the underlying layers. In addition, the platform enables the extension of its functionalities in a plug-and-play fashion with the development of lightweight applications, called *Custom Plugins*. The *Custom Plugins* are subscribed to the *Message Broker*, consume messages from it and perform their tasks. Two particular implementations for the PBF case will be presented in Section III: *Powder Bed Monitoring* and *Defect Detection*. All the platform functionalities can be accessible with an intuitive GUI (described in Section III-C) that has been customized for the particular case under study.

III. Use Case: Camera-Based Monitoring for Powder Bed Fusion

Powder Bed Fusion (PBF) is one of the most diffused metal additive manufacturing techniques in the industrial sector. In this technique, a metallic powder is spread over previous layers to be later selectively melted by a heat source. In this section, we present the platform implementation for PBF in a real industrial scenario. The platform comprises dedicated modules for in-situ monitoring using a low-cost camera and real-time defect detection. Moreover, after an extensive study of the requirements of different AM stakeholders, we developed a rich GUI for enhancing platform usability and accessibility.

A. Powder Bed Monitoring

To monitor the progress of the PBF building process, it is required to capture images of the building plate for every layer. The particular machine configuration in our study lacked this functionality. Consequently, a system for acquiring powder bed images was developed: the *Powder Bed Monitoring* (PBM) module. The hardware of the PBM consisted of an Arduino Uno microcontroller, a low-cost camera and an embedded PC running Linux (e.g. Raspberry Pi). The module acquired two images for every layer: one before the laser selectively melted the powder and one after. All valuable information at run time (e.g. time at which monitoring started, camera used, execution errors) is stored in a local file in the embedded PC. The module is accessible via a REST API. In this way, the monitoring module can be controlled directly from the platform GUI. The PBM enables the synchronization of external non-property hardware enhancing AM machines (lacking the layer-by-layer monitoring functionality) with the use of a low-cost camera.

B. Defect Detection

Defect Detection is a custom plugin that detects powder bed defects raised during the PBF production process [10]. This plugin analyses the layer-by-layer images of the building process through image processing and ML algorithms. The module allows the detection of five powder bed defects known to cause porosity and microstructural alterations during the building process on the PBF-produced parts: incandescence, spattering, horizontal, waves and holes. The *Defect Detection* module exposes REST APIs for launching and stopping the analysis as well as for retrieving useful information during run-time (e.g. the number of layers analyzed, execution time and defect detection results of each layer).

C. Graphical User Interface

The way the final user interacts with the platform is of vital importance. Consequently, to identify the user requirements that the platform must fulfil, we interviewed a series of AM stakeholders: CAD designers, supply engineers, process engineers, and post-process quality specialists. The first requirement was the possibility to download a *PDF* file which combined all the reports and the data of a particular file. Moreover, there should be an option for downloading a *zip* file containing all job-related data and files, including reports, monitoring powder bed images and STL files. On the other hand, the possibility to launch the *Defect Detection* analysis as well as a layer-by-layer visualization of the detected defects and graphs, showing a summary of the obtained results. Lastly, users should have different levels of access to restricted information according to their credentials.

After identifying the requirements the platform must fulfil, a rich and intuitive GUI was built using *React*, one of the most popular front-end libraries for building web applications. The platform supports two different types of users: Normal users, who can only handle jobs of the printers they have been assigned, and Admin users who can manage other users, manage printers, and manage the available jobs. Users

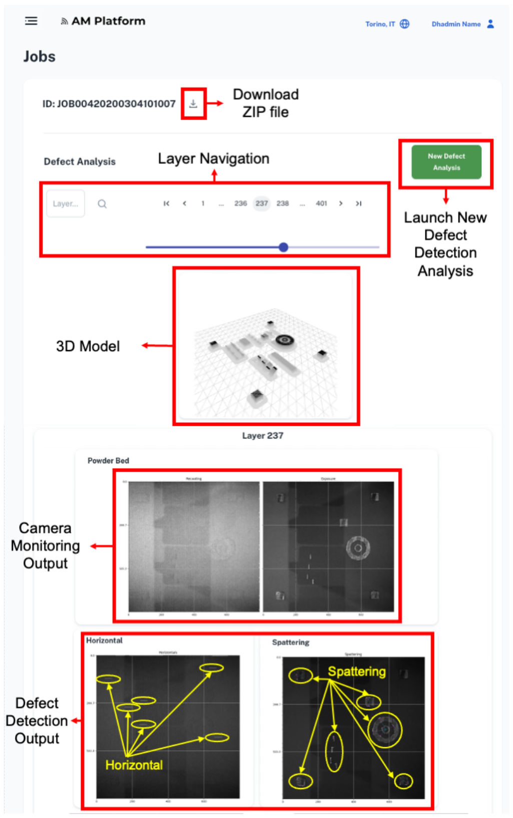

Fig. 2. Layer-by-layer visualization of Defect Detection results

are required to log in to access the platform functionalities. Through the GUI it is possible to manage printers, jobs and users: create, list, update and delete these resources. Moreover, it is possible to interactively upload, explore and visualize AM data. New data can be uploaded manually by filling out different forms and uploading files, or automatically through machine-readable files. Finally, it is possible to download a *PDF* file containing all job data and reports.

The platform enables the possibility to directly execute the *Defect Detection* analysis through the GUI, enabling non-experts users to run them without any concern about how they run or the resources they implement. For the visualization of the defect detection results, users can choose the best visualization for them: in graphs, showing a summary of the obtained results, or in a layer-by-layer fashion. Fig. 2 presents an example of this layer-by-layer visualization. This visualization comprises the *3D model* of the part being built, the *Camera Monitoring Output* and the *Defect Detection Output* for the selected layer. As shown in Fig. 2, it is possible to navigate through defect detection results of the different layers using one of the *layer navigation* options: slider, or directly searching for a particular layer of interest. The *3D model* enables a perspective rotation to visualize the produced part from different angles. What is more, the *3D model* is directly correlated to the layer under visualization: the black areas in the *3D model* represent the surface built in that layer while the grey ones are the ones built previously. The *Camera Monitoring Output* includes the two raw images captured with the PBM for that particular layer. On the other hand, the *Defect Detection Output* comprises images in which the detected defects are highlighted. The example shown in Fig. 2 comprises the images of the detected *Horizontal* and *Spattering* defects in that layer. Lastly, as can be seen in Fig. 2, it is even possible in this visualization to launch a new defect analysis for the selected job as well as download a *zip* file containing all job-related data and files.

IV. Conclusions and Future Works

This paper presented a real implementation of a distributed platform built with a microservices design pattern that supports all the stages of the AM process. The platform allows collecting, linking, storing and analysing the heterogeneous data generated throughout the AM lifecycle and takes a significant step in ensuring AM applications reach their full potential. The platform architecture enables the extension of its functionalities in a plug-and-play fashion. The rich GUI developed provides normal non-expert users with an easy-to-understand and intuitive way to interact with AM data. Moreover, the GUI integrates advanced analytics like the *Defect Detection* module, which enables non-IT experts to perform complex analyses without being aware of the details of its execution. Future works will include the integration of new analytics modules to take further advantage of the stored data (e.g. the usage of ML algorithms to establish a correlation between the defects found in image analysis and the process parameters used). In conclusion, the developed platform presents an enabling technology for onsite monitoring of AM processes as well as for process control.

Acknowledgment

This study was carried out within MICS (Made in Italy Circolare e Sostenibile) and received funding from the MUR-M4C2 1.3 of PNRR – D.D. 1551.11-10-2022, PE00000004.

References

[1] W. E. Frazier, "Metal additive manufacturing: a review," *Journal of Materials Engineering and performance*, vol. 23, pp. 1917–1928, 2014.
[2] S. A. Tofail *et al.*, "Additive manufacturing: scientific and technological challenges, market uptake and opportunities," *Materials today*, vol. 21, no. 1, pp. 22–37, 2018.
[3] A. Simeone, A. Caggiano, and Y. Zeng, "Smart cloud manufacturing platform for resource efficiency improvement of additive manufacturing services," *Procedia Cirp*, vol. 88, pp. 387–392, 2020.
[4] S. D. Cataldo *et al.*, "Optimizing quality inspection and control in powder bed metal additive manufacturing: Challenges and research directions," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 326–346, 2021.
[5] C. Liu *et al.*, "Digital twin-enabled collaborative data management for metal additive manufacturing systems," *Journal of Manufacturing Systems*, vol. 62, pp. 857–874, 2022.
[6] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
[7] K. S. Aggour *et al.*, "Federated multimodal big data storage & analytics platform for additive manufacturing," in *2019 IEEE international conference on big data (big data)*. IEEE, 2019, pp. 1729–1738.
[8] Y. Zhang *et al.*, "A systematic review on data of additive manufacturing for machine learning applications: the data quality, type, preprocessing, and management," *Journal of Intelligent Manufacturing*, pp. 1–36, 2022.
[9] D. Cannizzaro *et al.*, "In-situ defect detection of metal additive manufacturing: an integrated framework," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 74–86, 2021.
[10] ——, "Image analytics and machine learning for in-situ defects detection in additive manufacturing," in *Proc. of DATE*, 2021, pp. 603–608.