A Framework for Self-configuration in Manufacturing Production Systems

Hamood Ur Rehman[1,2](✉), Jack C. Chaplin[1], Leszek Zarzycki[2], and Svetan Ratchev[1]

[1] University of Nottingham, Nottingham, UK
{Hamood.Rehman,Jack.Chaplin,svetan.ratchev}@nottingham.ac.uk
[2] TQC Ltd., Nottingham, UK
leszek.zarzycki@tqc.co.uk

Abstract. Intelligence in manufacturing enables the optimization and configuration of processes, and a goal of future smart manufacturing is to enable processes to configure themselves – called self-configuration. This paper describes a framework for utilising data to make decisions for the self-configuration of a production system device in a smart production environment. A data pipeline is proposed that connects the production system via a gateway to a cloud computing platform for machine learning and data analytics. Agent technology is used to implement the framework for this data pipeline. This is illustrated by a data oriented self-configuration solution for an industrial use-case based on a device used at a testing station in a production system. This research presents possible direction towards realising self-configuration in production systems.

Keywords: Data analytics · Smart manufacturing · Agent technology · Configuration

1 Introduction

Dynamic manufacturing solutions are required to satisfy the demands of part customisation, facility optimisation and equipment efficiency. A newer direction being explored in this domain is intelligent production systems [1]. The majority of research that targets intelligent systems has dealt with the study and modelling of whole systems and often makes assumptions about the constituent elements of the system (machines, instruments, and components) including assumptions on machine integration, capabilities and performance.

Different types of products with differing features and characteristics may require operations with different configuration settings in a production system. These changes in configuration may be due to changes in physical parameters or calibrations, or due to system settings like data logging, or communication settings. The configuration change may also be due to a change in physical infrastructure, therefore there can be different configurations on the same machine. Significant work has been done that deals with change in configuration at the infrastructure level [2–4]. In contrast, the self-management

Published by Springer Nature Switzerland AG 2021
L. M. Camarinha-Matos et al. (Eds.): DoCEIS 2021, IFIP AICT 626, pp. 71–79, 2021.
https://doi.org/10.1007/978-3-030-78288-7_7

of the internal configuration of the machine by the machine itself is explored in fields like computer science and telecommunications, but there remain significant gaps in manufacturing.

The research outlined in this paper aims to close this gap in knowledge. It presents a framework with data models that establish a self-configuration mechanism for existing manufacturing equipment. Initially, the research focuses on an industrial use-case production system device and equipment used in testing processes in a manufacturing setup, that later could be used to address self-configuration in all production systems in the setup. In the remainder of the paper Sect. 2 gives a short literature review on the topic followed by an introduction to the developed framework for self-configuration in Sect. 3. Section 4 discusses a use-case implementation and Sect. 5 concludes the paper.

Research Question: Can artificial intelligence be used for internal configuration of a production system, allowing it to be managed by the production system itself (self-configuration)? What are the data model and technologies for it?

2 Literature Review

2.1 Artificial Intelligence for Smart Production Systems

Industry 4.0 is a shift towards smart digital manufacturing utilising concepts such as internet of things (IoT), cloud computing and cyber physical systems [1] to enhance manufacturing productivity. Conceptually, a smart manufacturing system should be able to gather information, parameters, and perceptions about itself for production and monitoring purposes and use it to make intelligent decisions.

Meyer et al. [5] described intelligence in procurement, manufacturing and transport as a way for the systems to influence operations. The criteria for an intelligent production system is based on five base characteristics: a component that can uniquely identify itself, communicate with other devices, retain data, deploy language to display features and is capable of participating in or making decisions [6].

Zhong et al. [7] defines intelligent production systems as technologies that embed and utilise informatics and advanced manufacturing techniques to address the dynamic and customised nature of products by achieving flexibility, re-configurability and scalability. Artificial Intelligence (AI) is a means to introduce such aspects of intelligence in production systems [5, 7, 9].

2.2 Self-concept in Manufacturing

The UK Government presented its new data strategy policy that addresses the move towards establishing a 'data-ecosystem' along with the potential use of data driven technologies in industry [8]. In manufacturing environments, there exists a general lack of homogeneity between different platforms and sources, which inhibits the implementation of smart production systems [9]. This heterogeneity necessitates a collaborative framework for device interaction. The collaborative framework must address the environment in a multi-level format. For the manufacturing environment these levels are

Table 1. Value extraction levels for intelligent systems [10]

Level	Description	Consideration
Level I – smart connection level	The effective acquisition of data from a source. Includes a platform that accepts data from a source and communication protocol for data transfer	This level relies heavily on data transparency and accessibility [11]
Level II – data-to-information level	Converting data to information. Capability should be present for data accumulation from multiple sources [12]	Data should be consistent and structured [13]
Level III – cyber level	Extracting meaning from gathered information for smart decision making	The gathered information may follow a trend in values [14]. This helps in developing knowledge base for smart decisions
Level IV – cognition level	Algorithms that are used to make approximations for objective functions	The possible solutions developed by this level should complement the goals of process/system [10]
Level V – configuration level	This level deals with acting on cognitive inferences	This level should make the system robust and resilient against loss and capable of making smart decision by itself

the smart connection-level, data-to-information level, cyber level, cognition level and configuration level [10] (see Table 1).

The Configuration level (V) deals with developing systems where each machine has the capability to decide its own parameters and operations, hence acting on 'cognition' - the generation of insight by AI algorithms based on useful information extracted from data. This is a new domain witnessing significant growth, where applications are capable of automatically adjusting and adapting themselves to varying scalability, complexity, and insight needs. The properties which make a system capable of such adaptability are referred to as self-* properties [15] and include self-management [3], self-stabilisation [2], self-healing, self-organisation [16], self-protection [17], self-optimisation, self-configuration, self-scaling, self-awareness, self-immunity and self-containment. For the purpose of this research, the concept of self-configuration is defined as:

– *The property that deals with response. It is the ability of system to change its configuration (i.e., the connection between different system modules, parameters, and calibration) in order to improve or restore system functionality in response to actions.*

2.3 Enabling Technologies for Self-configuration

Manufacturing is being transformed by data acquisition technologies (like smart sensors), data management systems, data processing techniques (such as cloud computing, big data, artificial intelligence and machine learning), information and communication management methods, and digital twins. Embedding and utilising these technologies on manufacturing objects (like machines, tools, and products) bridges the gap between physical and cyber, enabling smart equipment [18].

Technologies such as multi-agent systems, grid computing, control theory (distributed resources connected through information technology network [16]) and component-based development show promise towards enabling the self-configuration objective. The dominant approaches that are currently being researched to address the issue are multi-agent systems [4] and component-based development [19]. Multi-agent system approach will be taken for this implementation of self-configuration capability in production system. It will be deployed on an industrial use-case and validated for readiness in introducing self-capabilities for industrial adoption.

3 Framework for Self-configuration in Production Systems

In general, a production system comprises numerous pieces of equipment and processing stations or stages. Initially, this concept of a framework for self-configuration is limited to a single (configurable) device or stage, and is presented here in the context of a production testing process. Current approaches require expert knowledge for configuration, and show clear need for self-configuration processes.

Table 2. Agent description in the context of a testing system

Agent	Description
Geometrical Agent (GA)	Agent responsible for part feature identification and feature comparison with cloud-based feature libraries. Based on features identified it also identifies the part that is being tested
Criterion Agent (CA)	Searches and implements potential testing configurations that satisfy the testing criteria. Interacts with the Function Agent (FA) to select the best configuration possible during iteration loops
Function Agent (FA)	Responsible for functionality execution and comparison with objective function. It interacts with CA to select best possible configuration during iteration loops. It is also responsible for communicating the selected configuration to the user

The framework involves seven actors; the *user*, the testing production *system device*, *Geometric Agents* (GA), *Criterion Agents* (CA), *Function Agents* (FA) (see Table 2), a *database/Machine Learning* (ML) *pipeline*, and a *cloud platform.* The cloud platform is a group of services hosted in a cloud environment that forms the part of data pipeline for self-configuration. The services along with formation of a general data pipeline for self-configuration is elaborated in the next subsection. The developed framework is outlined in Fig. 1. The self-configuration framework and the sequence of operations is explained in Fig. 2.

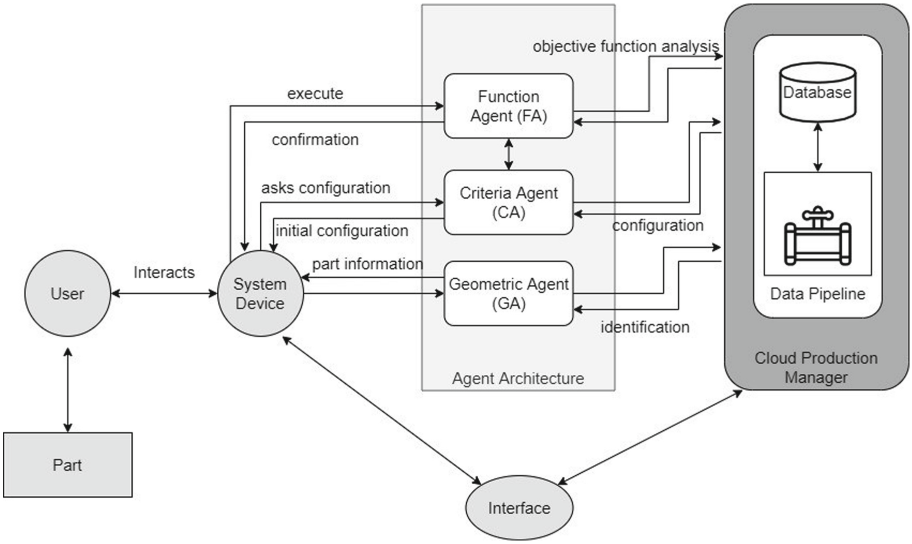

Fig. 1. Architecture of the framework for self-configured production system

3.1 Sequence of Proposed Self-configuration Strategy

The process parameters, communication/connectivity settings, and calibration data are the main constituents of a configuration in production testing systems. The movement of data in the operation loop is managed by agent technology that oversees, monitors, initiates, and terminates the operations happening during execution as needed. These agents interact and communicate with the equipment in conjunction with the cloud platform to perform self-configuration. The chain of interactions between agents, systems, and users is shown in the sequence diagram (Fig. 2).

– **Part Introduction:** The sequence commences with the user introducing the part to the system device (e.g., the production test equipment). This event triggers a response from the Geometric Agent (GA) that extracts information about features from the system device and sends them for a look-up to the cloud platform. The cloud platform is aided by ML pipelines that refer to features stored in a database. The ML pipeline looks up the features and confirms to the cloud platform of features recognised. The set of the features are compared with known parts having those features. The part that has maximum feature mapping is recognised as the part submitted for testing by the user. This information is sent to the system device to be displayed to the user by the GA.

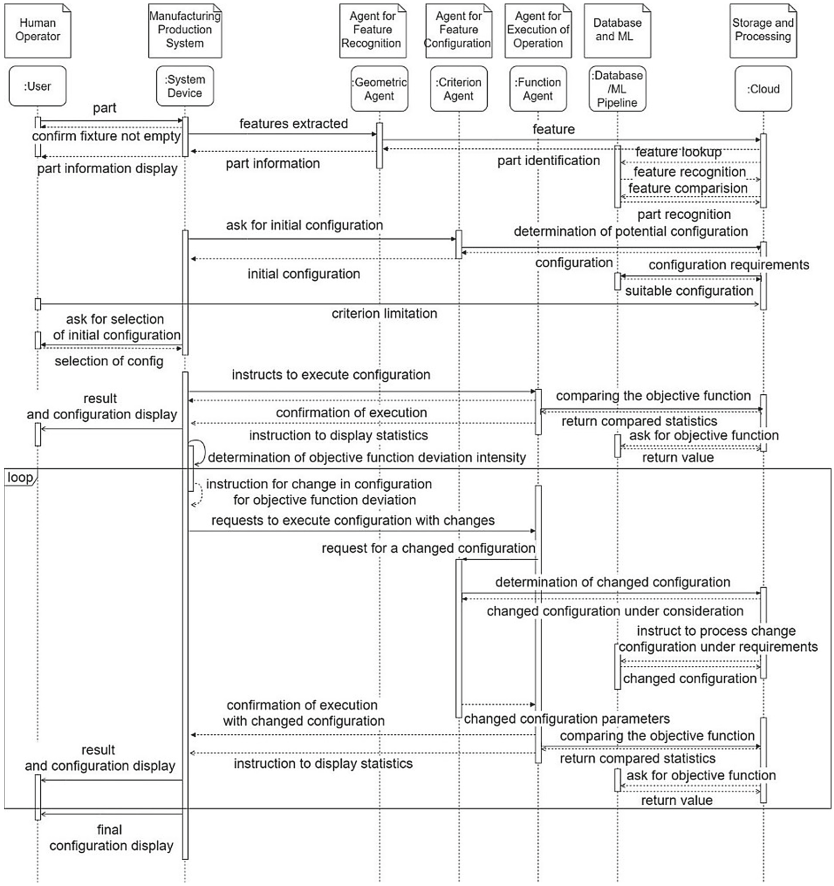

Fig. 2. Sequence diagram of proposed self-configuration framework

– **Look for Configuration:** The next step for the system is to look for a suitable configuration. The system uses the Criterion Agent (CA) to accomplish this. The CA sends an inquiry to the cloud platform that utilises the help of the database to look for a previous or a similar configuration for the part or similar parts. The database returns the testing configuration as a standard schema to the system device through the CA. Any limitations specified by the user is incorporated in the configuration file at this stage.
– **Operation Execution:** The device executes the configuration guided by the Function Agent (FA). The FA logs the execution start and confirms the finish. The results obtained from the operation are sent to the cloud platform by the FA.
– **Result Analysis and Comparison:** The cloud platform uses a ML pipeline to provide a comparison between the result and a suitable objective function. The objective function selected depends on the operation being performed, outcome desired and operation specifics. The comparison is returned by the cloud platform and relevant statistics displayed to the user. If there is no unacceptable deviations from the threshold

then the initial configuration is selected as the final configuration and displayed to user. This deviation in testing is the effectiveness of the pass/fail criteria based on current parameters and calibration settings in system configuration.

– **Reiteration of Operation:** If significant deviation is present then the process is repeated by the system. The function agent (FA) requests the criterion agent (CA) for a new/changed configuration. The CA forwards this request to cloud platform ML pipeline and database for changed configuration values. This lookup to new values is dependent on the instruction obtained from the system device for change in configuration as per objective function deviation. The changed configuration is sent back to the FA from CA and through FA to the system device. Execution of the operation is carried out.
– **Operation Loop:** The results are again compared to objective function. The statistics are displayed to the user by the system as required. If the deviation is acceptably low then this becomes the final configuration else the process is looped till an optimum or a maximum iteration limit is reached.

4 Implementation and Deployment to an Industrial Use-Case

Validation of the framework can be carried out by demonstrating the data pipeline connected to a physical production device through a gateway to the cloud platform services (Fig. 3). The gateway device interfaces with the production equipment, sends the data to the cloud platform, and hosts the agents. Data transferred to storage is retrieved and loaded into a data warehouse service where the table and schema are defined. For the purpose of analysis, data is retrieved from the data warehouse into the coded ML model in the ML supported service on the cloud platform. The insights generated by the ML model is sent back to the warehouse service (to structure the data) and then to storage hosted in the cloud. The cloud storage sends the data to the gateway device from where the agents control the working of the equipment.

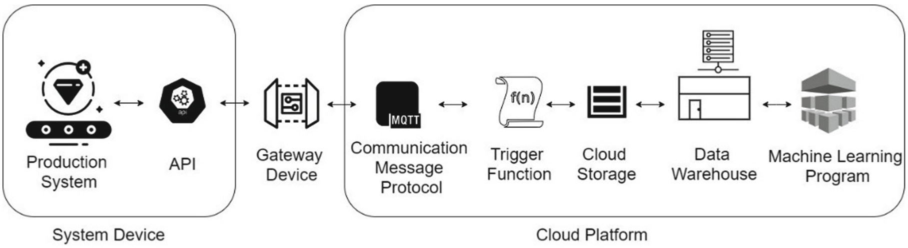

Fig. 3. Data pipeline deployment for self-configuration in production systems

Pipeline automation is achieved by an event-based service called a trigger function. In the cloud, single purpose functions are attached to events that happen in real-time. These functions are triggered as events being observed are executed. The gateway device that connects the system device to the IoT services via a communication message protocol can have regular set of events logged, and these events could trigger functions that co-ordinate the components of data pipeline for self-configuration. An Application Programming Interface (API) is also needed between the testing production system and the gateway device that makes the functionality of the testing system accessible to the rest of the pipeline.

A data pipeline (Fig. 4) was set up connecting an industrial (TQC Ltd.) leak testing rig TAMI (Test bench for leakage identification on aircraft fluid mechanical installations) and dry-air leak testing device MALT (Micro Application Leak Test) with a Raspberry Pi (the gateway device) through the API. The physical hardware was connected to Google Cloud Platform through MQTT Pub/Sub Protocols and an IoT gateway. The future work on self-configuration of production systems will involve this pipeline to optimise parameters (stabilisation time, test pressures and differential pressures) for pneumatic leak testing. Based on this setup the self-configuration framework is deployed and will be validated for testing applications.

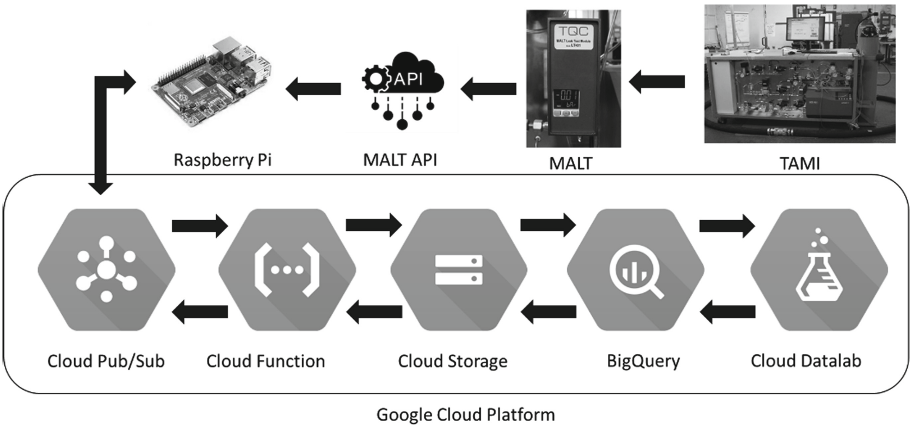

Fig. 4. Complete pipeline deployed for the use-case with API: an initial pipeline for self-configuration research

5 Conclusion and Future Work

The research presents a framework for self-configuration in production systems. This approach is beneficial in saving expert worker's time, costs and reducing management overheads for common testing scenarios. A cloud-based data pipeline is integrated with multi-agent system to support a machine learning process for self-configuring and executing testing processes. Future work involves the full implementation of this framework with state transition management under real-time constraints. The solution will be generalised with a common ontology and semantics developed to enable broader application. The industrial use-case will continue to be used as validation platform for the research, the solution will be iteratively developed and deployed on it.

Currently, this approach is limited to a single device or stage and is tailored for testing applications. It is envisaged that with further research it can encompass multiple interconnected devices and the production system as a whole and abstracted to other production applications. Another limitation observed in the research is related to interoperability and adaptation requirements of the production system. Enforcement of the necessary self-configuration capability is subject to a production system's interface with agents via gateway devices, and to maintain effective data transmission across the cloud pipeline.

References

1. Uhlmann, E., Hohwieler, E., Geisert, C.: Intelligent production systems in the era of industrie 4.0–changing mindsets and business models. J. Mach. Eng. **17** (2017)
2. Botygin, I., Tartakovsky, V.: The development and simulation research of load balancing algorithm in network infrastructures. In: 2014 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS), pp. 1–5. IEEE (2014)
3. Antzoulatos, N.: Towards self-adaptable intelligent assembly systems. Ph.D. thesis, University of Nottingham (2017)
4. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D.: Dynamic self-organization in holonic multiagent manufacturing systems: the ADACOR evolution. Comput. Ind. **66**, 99–111 (2015)
5. Meyer, G.G., Framling, K., Holmström, J.: Intelligent products: a survey. Comput. Ind. **60**(3), 137–148 (2009)
6. McFarlane, D., Sarma, S., Chirn, J.L., Wong, C., Ashton, K.: The intelligent product in manufacturing control and management. IFAC Proc. Vol. **35**(1), 49–54 (2002)
7. Zhong, R.Y., Xu, X., Klotz, E., Newman, S.T.: Intelligent manufacturing in the context of industry 4.0: a review. Engineering **3**(5), 616–630 (2017)
8. GOV.UK: Policy paper UK National Data Strategy, pp. 1–73 (2020)
9. Sztipanovits, J., et al.: Toward a science of cyber–physical system integration. Proc. IEEE **100**(1), 29–44 (2011)
10. Kao, H.A., Jin, W., Siegel, D., Lee, J.: A cyber physical interface for automation systems—methodology and examples. Machines **3**(2), 93–106 (2015)
11. Aazam, M., Hung, P.P., Huh, E.: Smart gateway based communication for cloud of things. In: 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 1–6 (2014)
12. García, C.G., Meana-Llorián, D., Lovelle, J.M.C., et al.: A review about smart objects, sensors, and actuators. Int. J. Interact. Multimedia Artif. Intell. **4**(3) (2017)
13. Mourtzis, D., Vlachou, E., Milas, N.: Industrial big data as a result of IoT adoption in manufacturing. Procedia CIRP **55**, 290–295 (2016)
14. Curcurú, G., Cocconcelli, M., Rubini, R., Galante, G.M., Miraglia, V.M.: Bayesian approach in the predictive maintenance policy. In: Proceedings of 9th International Conference Surveillance, pp. 22–24 (2017)
15. Sanchez, M., Exposito, E., Aguilar, J.: Implementing self-* autonomic properties in self-coordinated manufacturing processes for the industry 4.0 context. Comput. Ind. **121**, 103247 (2020)
16. Khalgui, M., Mosbahi, O.: Intelligent distributed control systems. Inf. Softw. Technol. **52**(12), 1259–1271 (2010)
17. Yan, J., Vyatkin, V.: Extension of reconfigurability provisions in IEC 61499. In: 2013 IEEE 18th Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–7. IEEE (2013)
18. Xu, G., Huang, G.Q., Fang, J.: Cloud asset for urban flood control. Adv. Eng. Inform. **29**(3), 355–365 (2015)
19. Jann, J., Browning, L.M., Burugula, R.S.: Dynamic reconfiguration: basic building blocks for autonomic computing on IBM pSeries servers. IBM Syst. J. **42**(1), 29–37 (2003)