Journal of Intelligent Manufacturing (2020) 31:1837–1859
https://doi.org/10.1007/s10845-020-01539-4

The architecture development of Industry 4.0 compliant smart machine tool system (SMTS)

Byeongwoo Jeon[1] · Joo-Sung Yoon[2] · Jumyung Um[3] · Suk-Hwan Suh[1]

Received: 8 June 2019 / Accepted: 17 January 2020 / Published online: 29 January 2020

Abstract
As part of the fourth industrial revolution, the movement to apply various enabling technologies under the name of Industry 4.0 is being promoted worldwide. Because of the wide range of applications and the capacity of manufacturing workpieces flexibly, machine tools are regarded as essential industrial elements. Hence, much research has been concerned with applying various enabling technologies such as cyber-physical systems to machine tools. To realize a machine tool suitable for Industry 4.0, development should be done in a systematic manner rather than the ad-hoc application of enabling technologies. In this paper, we propose a functional architecture for the Industry 4.0 version of machine tools, namely smart machine tool system. To reflect the voices of various stakeholders, stakeholder requirements are identified and transformed into design considerations. The design considerations are incorporated into the conceptual model and functional modeling, both of which are used to derive the functional architecture. The implementation procedure and an illustrative case study are presented for the application of the functional architecture.

Keywords Smart machine tool system (SMTS) · Cyber-physical manufacturing system (CPMS) · Machine tool cyber system (MTCS) · Cyber-physical system (CPS) operator · Monitoring · Analysis · Plan · Execution/big data analytics and AI, digital twin (MAPE/BD)

Abbreviations

AI	Artificial intelligence
API	Application programming interface
BD	Big data analytics and AI, digital twin
BDA	Big data analytics
CAD	Computer-aided design
CAE	Computer-aided engineering
CAM	Computer-aided manufacturing
CAI	Computer-aided inspection
CAPP	Computer-aided process planning
CAX	Computer-aided X
CNC	Computerized numerical controller
CPMS	Cyber-physical manufacturing system
CPS	Cyber-physical systems
DAQ	Data acquisition
DT	Digital twin
HMI	Human–machine interface
IDEF0	Part of the IDEF modeling languages to model the function of the system. Integration definition (IDEF) is a series of modeling languages in the field of systems/software engineering
IoT	Internet-of-Things
KPI	Key performance indicator
M2M	Machine-to-machine
MAPE	Monitoring, analysis, plan, and execution

✉ Suk-Hwan Suh
shs@postech.ac.kr

Byeongwoo Jeon
jhs@postech.ac.kr

Joo-Sung Yoon
jsyoon@kitech.re.kr

Jumyung Um
jayum@khu.ac.kr

[1] Research Center for Smart Factory, POSTECH, 77 Cheongam-Ro, Nam-Gu, Pohang, Gyeonsangbuk-Do 790-784, South Korea

[2] Smart Manufacturing Technology Group, Korea Institute of Industrial Technology, 89, Yangdaegiro-Gil, Ipjang-Myeon, Seobuk-Gu, Cheonan, Chungcheongnam-Do 31056, South Korea

[3] Smart Manufacturing Laboratory, Kyung Hee University, 1732 Deogyeong-Daero, Giheung-Gu, Yongin, Gyeonggi-Do 446-701, South Korea

MAPE/BD	Monitoring, analysis, plan, execution/big data analytics and AI, digital twin
MTBF	Mean time between failure
MTCS	Machine tool cyber system
MTTR	Mean time to repair
OPC-UA	Object linking and embedding for process control unified architecture (IEC/TR 62541)
PLC	Programmable logic controller
RAMI 4.0	Reference architecture model for Industry 4.0
RUL	Remaining useful life
SMTS	Smart machine tool system
STEP-NC	The standardized data model for computerized numerical controllers (nickname of ISO 14649)

Introduction

Since it started in 2011 as part of Germany's manufacturing revival policy, Industry 4.0 has become the manufacturing model of the fourth industrial revolution or the smart factory model. In response to the changing modern manufacturing environment such as personalized production, flexible production lines, etc., much effort was made to fit the current manufacturing environment into the Industry 4.0 philosophy by applying Industry 4.0 enabling technologies. As part of this, the reference architecture model for industry 4.0 (RAMI 4.0) is a three-dimensional model showing how to realize Industry 4.0 in a structured manner (Bitkom 2016). RAMI 4.0 allows all participants involved in Industry 4.0 discussions to share a common understanding. All enabling technologies and elements are mapped thanks to the three-axis concept, consisting of Hierarchy levels, Life cycle, and Value stream layers.

One important component of manufacturing is the machine tool. A machine tool is defined as a mother machine because it is a machine that makes machines (Suh et al. 2008). It is a key manufacturing device in manufacturing since it is indispensably used in the production of machine parts used in virtually all industries and can be flexibly adapted to orders and product group changes. Zero downtime and zero defects are key objectives of machine tools in modern times because mass production should be realized while satisfying the various and stringent product quality conditions. Interaction environments among shop floor components is also a key objective because, from the manufacturer's point of view, coordination of shop floor operations in response to machine tool status is crucial. Many institutions have long been devoted to technological development from zero downtime, zero-defect points of view and are considering Industry 4.0 enabling technologies as a new means. In this sense, the Industry 4.0 version of a machine tool needs to be developed for the true realization of manufacturing with Industry 4.0. This paper is concerned with developing the system architecture and the purpose of the system architecture is to reflect intelligent functions systematically that can realize zero downtime and zero defects of the machine tool itself and provide an interaction environment among shop floor components for contributing to shop floor operation improvement.

Previous research on intelligence of the machine tool itself has been concerned with downtime/defect reduction, such as tool wear prediction (Siddhpura and Paurobally 2013; Li 2012; Gajate et al. 2012), thermal precision on machine (Ruijun et al. 2012; Li et al. 2014), machine tool fault diagnosis (Goyal and Pabla 2015), chatter control (Quintana and Ciurana 2011; Siddhpura and Paurobally 2012; Tangjitsitcharoen et al. 2015), cutting parameter optimization (Chandrasekaran et al. 2010; Mellal et al. 2016), servo tuning (Singh et al. 2015; Dong et al. 2012), surface roughness prediction (Khorasani et al. 2012; Çaydaş et al. 2012). They focus on the implementation of specific functions rather than realizing these in a systematic framework.

Architecture studies that can embrace CNC intelligence have also been conducted before Industry 4.0. The CNC architecture combined with STEP-NC was the focus. Suh and Cheon (2002) developed an intelligent, STEP-compliant CNC system architecture and data model that can embrace seamless integration of the CAD–CAM–CNC chain and provide an open and modular capability for access and utilization of internal functions of CNC and development of intelligent functions. It was developed before Industry 4.0, i.e., before IoT and CPS were emphasized.

Studies on the machine tool in line with the Industry 4.0 view have also been conducted. Liu and Xu (2017) proposed "machine tool 4.0", i.e., machine tool with the Industry 4.0 enabling technologies such as CPS and IoT. This study shows the architecture that integrates machine tools, computation, and networking, where computation and networking technologies can conduct monitoring and intelligent control of machining processes. It focuses on the integration of some Industry 4.0 enabling technologies with existing technologies related to intelligence on the machine tool itself.

Even though the machine tool itself is a key element in manufacturing, it is also important for the machine tool to have close interaction with other shop floor elements. Although there is a very little research on interactions among shop floor components focusing on machine tools, research on the autonomous shop floor itself has been conducted based on theories such as holonic manufacturing systems, multi-agent systems, etc. (Lin and Solberg 1992; Leitão 2009) Development of a dynamic scheduling system (Sousa and Ramos 1999; Gou et al. 1998; Kouiss et al. 1997), framework for representing shop floor elements and operations (Van Brussel et al. 1998; Shen et al. 2000), autonomous shop floor control (Brennan et al. 1997; Schild and Bussmann 2007; Colombo 1998) etc., have been conducted. Most of these studies were conducted before industry 4.0 and it is necessary to study autonomic control, smart factory implementation technologies and operational technologies in the manufacturing field in combination.

In deriving our proposed system, it is not desirable to simply apply Industry 4.0 enabling technologies to the machine tool domain. In other words, it is necessary to derive methods under the coordination of the Industry 4.0 vision and a concrete framework of realization. Given this, our research team has derived the CPMS paradigm, as shown in Fig. 1 (Suh 2017) CPMS is a "compact-implementable" model of Industry 4.0, based on RAMI 4.0 and its implementation techniques. The CPMS consists of: (1) a physical system consisting of the office floor and shop floor, (2) a cyber system consisting of MAPE/BD supporting the operation of the physical system. MAPE/BD means monitoring KPIs, analysis (prediction, diagnostics), plan (reconfiguration, simulation), execution (feedback to physical function based on output from monitoring, analysis, plan) based on big data analytics and AI (BDA and AI), digital twin (DT). The cyber system consists of a monitoring, analysis, plan, and execution (MAPE) loop that is based on BD (big data analytics/AI and digital twin), and seeks to maximize the KPIs of the manufacturing system. The cyber system is developed based on MAPE-K, an autonomic computing model proposed by IBM (2005). K in MAPE-K means knowledge.

Based on the CPMS, we propose a reference architecture of the Industry 4.0 version of a machine tool, namely smart machine tool system (SMTS). In Sect. 2, stakeholder requirements are identified and transformed into design considerations of SMTS in Sect. 3. In Sect. 4, an architecture based on the design considerations is developed. In Sect. 5, for the implementation/customization of the reference architecture, the implementation procedure is proposed, followed by an illustrative case study in Sect. 6.

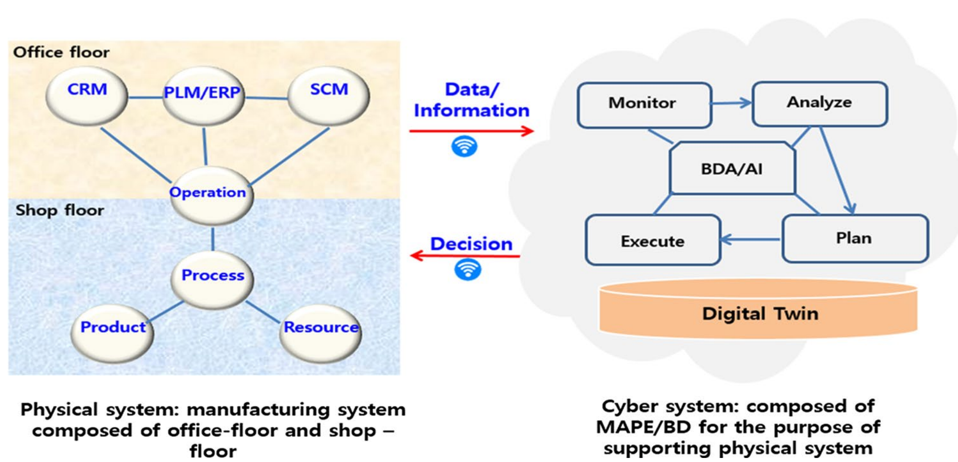
Stakeholder requirements

It is necessary to recognize what kinds of features the SMTS should have in order to develop a machine tool system that is based on Industry 4.0 and that satisfies stakeholders. For such a purpose, we present stakeholder requirements identified through interviews, benchmarking, technical conferences, etc. These stakeholders were largely identified from both the demand and supply industries. In the supply industry, there are machine tool builders/CNC vendors and system integration companies providing shop floor solutions. In the demand industry, there are machining workers, maintenance engineers and production managers. Since this paper is concerned with zero downtime and defects on the machine tool itself and contribution to efficient shop floor operation with Industry 4.0 perspective, requirements related to these are highlighted.

Machine tool builder/CNC vendor perspective

- [ReqC #1] It is necessary for the system to have sensors installed. This means that the system to be developed can collect raw signals representing the machine tool status.
- [ReqC #2] It is necessary for the system to have the means for acquiring data from the machine tool. This means that the system to be developed can collect signals for further processing.
- [ReqC #3] It is necessary for the machine tool system to transfer the collected data to an intelligent system that processes/utilizes the data. This means that the system to be developed can connect the machine tool to the data-handling module.
- [ReqC #4] It is necessary for the system to organize the data so that it can be used for machine tool intelligent functions. This means that the system to be developed can manage all data for universal usage.

Fig. 1 Cyber-physical manufacturing system (CPMS) paradigm



- [ReqC #5] It is necessary for the system to connect with legacy systems. This means that the system to be developed can get CAD, CAE, CAM, and CAI data on parts, machine tool H/W, cutting tools, etc., for running/building intelligent algorithms.
- [ReqC #6] It is necessary for the system to have a function execution environment. This means that the system to be developed can process organized data ([ReqC #4]) to produce meaningful decision-making.

System integrator perspective

- [ReqS #1] It is necessary for the system to have an interface for data acquisition from the machine tool for shop floor level coordination. This means that the system to be developed can provide an interfaces for data acquisition.
- [ReqS #2] It is necessary for the system to cope with different communication interfaces on machine tools. This means that the system to be developed can provide interfaces to support a variety of machine tool types for shop floor level coordination.
- [ReqS #3] It is necessary for the system to overcome different information formats on machine tools. This means that the system to be developed can handle different machine tool information formats. Even though there is the part program standard (ISO 6983), each machine tool builder and CNC vendor has developed their own part program specifications. Also, each machine tool builder and CNC vendor has developed their own CNC and sensor information model specifications because there is no official standard except the companion specification.

Machining operator perspective

- [ReqO #1] It is necessary for the system to report the machining status to the operator in real-time. This means that the system to be developed can help the operator know the machine tool status and make the right decision.
- [ReqO #2] It is necessary for the system to detect machining process defects or predict occurrences. This means that the system to be developed can help to reduce product defects by giving a warning before a defect occurs.
- [ReqO #3] It is necessary for the system to provide a strategy to avoid machining defects. This means that the system to be developed can help reduce product defects by giving avoidance strategies.
- [ReqO #4] It is necessary for the system to diagnose the cause of phenomena in the case of machining defects. This means that the system to be developed can help to indicate causes so that workers can adopt different strategies later.

Maintenance engineer perspective

- [ReqM #1] It is necessary for the system to detect and notify immediately when machine tool failure occurs. This means that the system to be developed can help maintenance engineers and workers respond rapidly.
- [ReqM #2] It is necessary for the system to diagnose the causes of phenomena in the case of machine tool failure. This means that the system to be developed can help provide causes to workers and maintenance engineers so that they use less time finding root causes.
- [ReqM #3] It is necessary for the system to predict when machine tool failure will occur. This means that the system to be developed can help maintenance engineers and workers prepare in advance.
- [ReqM #4] It is necessary for the system to present processing conditions that can minimize machine tool failure. This means that the system to be developed can help increase the MTBF.

Production manager perspective

- [ReqP #1] It is necessary for the system to communicate with adjacent material handlers. This means that the system to be developed can automatically interact with preceding and succeeding processes.
- [ReqP #2] It is necessary for the system to perform machining operations so that its work coincides with the overall shop floor level production plan. This means that the system to be developed should be in line with the shop floor.
- [ReqP #3] It is necessary for the system to contribute to changes in the shop floor production plan in the case of machine tool failure. This means that the system to be developed should contribute to reducing shop floor operation delay.
- [ReqP #4] It is necessary for new machine tools to contribute to changes in shop floor production plans in the case of machining defects. This means that the system to be developed should contribute to reducing shop floor operation delays.

Design considerations

The stakeholder requirements identified in Sect. 2 need to be transformed into design considerations for the SMTS architecture. Design considerations are derived from the perspective of three categories: (1) Industry 4.0 components, (2) intelligence of the machine tool itself, (3) contribution to the autonomous shop floor. An Industry 4.0 component (Grangel-Gonzalez et al. 2016) is defined as a shop floor device equipped with an administration shell. The administration shell is defined as a systematic organization of data elements, functions of objects to be used by other systems and components. It mainly consists of a manifest and component manager. Manifest means an externally accessible set of structures on information, function. Component manager means the module that practically manages information, functions in shop floor devices. This paper proposes these perspectives so that all of the stakeholder requirements can be reflected in the architecture design.

Industry 4.0 component perspective

- [DCi #1] Machine tool data acquisition: This means a mechanism for acquiring data from the machine tool controller, sensors and HMI (Human Man Interface). This consideration is essential for various intelligent functions and recognizing machine status. For this, a DAQ board and controller API based on TCP/IP should be provided.
- [DCi #2] CAX information interface environment for machine tool hardware: This means a collection environment in which computer-based design data for machine tool hardware and parts are provided. This consideration is needed for various intelligent functions on the machine tool. For this, a platform-independent engineering data exchange format, i.e., AutomationML (IEC TC65/SC65E 2018) is required.
- [DCi #3] Multi-type communication protocol that can communicate with the machine tool: This means various communication environments which are universally applied to various types of the machine tool. This consideration is essential for universal system development. Not only existing protocols such as RS-232, RS-485, TCP/IP, etc. but also recent ethernet-based protocols such as Ethercat, etc., can be supported.
- [DCi #4] Interoperability environment: This means an environment that data and functions can be utilized by external systems. This consideration is critical for coordinating the shop floor, which includes different kinds of facilities, information contents, etc. For this, an interoperability protocol such as OPC-UA (IEC TC65/SC65E 2016) can be used.
- [DCi #5] Standardized information model: This means an information model that can be applied to develop various types of intelligent functions using common data specifications. This consideration is important for developing functionalities related to machine tool operation. For this, a common cyber system information model should be established before intelligent functions are implemented.
- [DCi #6] An externally accessible set of information structures: This means building an externally accessible set of information structures for inquiry of available information from external information systems. This consideration is essential for external systems to find out what kind of data exists and how they are structured. For this, XML file-based schema can be used. This corresponds to the manifest in the asset administration shell (Wagner et al. 2017).

Machine tool's own intelligence perspective

- [DCs #1] Data analytics infrastructure: This means a computing environment capable of collecting, storing and processing data. This consideration is essential for intelligent functions that utilize data for decision-making. For this, the data interface to the machine tool, storage, processing infrastructure are required.
- [DCs #2] Quantification of indicators for intelligent functions: This means quantification of indicators that are targeted by various intelligent functions. This consideration is critical for intelligent functions because it can contribute to the specification of the performance goal for each intelligent function. For this, KPIs for machine tool operations are required.
- [DCs #3] KPI calculation: This means calculating KPIs for target intelligent functions based on recently collected data. This consideration is needed for quantifying the current operating state of the machine tool. For this, various KPI calculation mechanisms are required.
- [DCs #4] KPI prediction: This means predicting KPI values for target intelligent functions. This consideration is needed for quantifying the future operating state of the machine tool so that deterioration of conditions can be anticipated. For this, various KPI prediction mechanisms including AI (Artificial Intelligence) are required.
- [DCs #5] Diagnosis of KPI: This means analyzing the causes of why the KPI values do not meet the target range or value. This consideration is needed for finding out why the current operational state is in bad condition. For this, various KPI diagnosis mechanisms including AI or heuristic approaches are required.
- [DCs #6] Re-parameterization for KPI: This means making decisions that can optimize the KPI value for each intelligent function. This consideration is needed for deriving countermeasures so that the machine tool can return to a desired operational state. For this, various re-parameterization mechanisms including meta-heuristic approaches are required.
- [DCs #7] Simulation environment: This means a simulator that can verify the suitability of the derived decision. This consideration is needed for verifying whether the proposed solution is appropriate for transmission to the machine tool. For this, a machining simulator with kinematic or dynamic factors is required.
- [DCs #8] Execution of decision: This means the transmission of the decision based on output from KPI calculation, prediction, diagnosis, reconfiguration with verifi-

cation through simulation. This consideration is needed for delivering control instructions that contribute to enhancing machine tool operation. For this, an interface to the machine tool and data transformation logic for the machine tool are required.

Contribution to autonomous shop floor perspective

- [DCa #1] Shop floor production schedule reception: This means an environment that can receive and interpret a shop floor-level operation schedule. This consideration is important for integration with shop floor operations. For this, an interface to a shop floor coordinating system and interpretation logic is required.
- [DCa #2] Communication environment with adjacent material handling system: This means an environment where M2M communication is possible with adjacent material handlers such as gantry, robot, etc. This consideration is needed for cooperation with preceding and succeeding processes on machine tools on the shop floor. For this, an interface to the adjacent material handling system is required.
- [DCa #3] Interaction with adjacent material handling systems: This means interaction with adjacent material handling systems based on the received shop floor operation schedule and status of the machine tool itself. This consideration is critical for interaction with the preceding and succeeding processes of the machine tool. For this, a mechanism for interpreting interaction signals between material handling systems and the machine tool and a mechanism for generating interaction signals are required.
- [DCa #4] Minimizing interference with shop floor operation plan: This means continuous check on machining conditions to make decisions in the best way to minimize performance deterioration of shop floor operations. This consideration is critical for contributing to dynamic coordination on the shop floor. For this, a mechanism for checking current machine status and generating request messages for shop floor coordination is required.

Since the design considerations are derived in response to stakeholder requirements, it is important to ensure that all stakeholder requirements are reflected. To do so, mapping work is performed to ensure that all of the requirements are mapped to design considerations, as shown in Table 1.

Table 1 Relationships of components in design consideration and stakeholder requirement

DC id	Related stakeholder requirement
[DCi #1]	[ReqC #1], [ReqC #2], [ReqS #1]
[DCi #2]	[ReqC #5]
[DCi #3]	[ReqC #3], [ReqS #2]
[DCi #4]	[ReqS #2], [ReqS #3]
[DCi #5]	[ReqC #4],[ReqS #3]
[DCi #6]	[ReqC #4]
[DCs #1]	[ReqC #6]
[DCs #2]	[ReqO #1 ~ 4], [ReqM #1 ~ 4]
[DCs #3]	[ReqO #1], [ReqO #2], [ReqM #1]
[DCs #4]	[ReqO #2], [ReqM #3]
[DCs #5]	[ReqO #4], [ReqM #2]
[DCs #6]	[ReqO #3], [ReqM #4]
[DCs #7]	[ReqO #3], [ReqM #4]
[DCs #8]	[ReqO #3], [ReqM #4]
[DCa #1]	[ReqP #2]
[DCa #2]	[ReqS #2], [ReqP #1]
[DCa #3]	[ReqP #1]
[DCa #4]	[ReqP #3], [ReqP #4]

Developing the architecture for SMTS

In this section, the SMTS architecture is developed by providing vision, conceptual model, IDEF0 functional modeling. Referring to stakeholder requirements and design considerations, the vision of SMTS is derived. This proposes the definition of SMTS. The SMTS concept model is derived based on the SMTS vision. This model organizes the functionalities in the conceptual level based on the SMTS vision, referring to IDEF0 modeling. The IDEF0 model is to derive functionalities of the system architecture from the design considerations presented in Sect. 3. Derived functions from the IDEF0 model are represented in the SMTS functional architecture with the concept model under consideration.

Vision and conceptual model of SMTS

SMTS is the machine tool system that reflects the vision of Industry 4.0 from the viewpoint of MAPE/BD within the CPMS paradigm. It includes two aspects, that is: (1) operation of the machine tool, (2) contribution to shop floor operation. For the operation of the machine tool itself, KPI based improvement on machine tool operation, which can aim for zero downtime and defects for the machine tool, is realized by integrating enabling technologies of Industry 4.0 under the MAPE/BD frame with CAX information (from CAM, CAE, CAD, and CAI) considered. For the contribution to shop floor operation, an interaction environment with the shop floor coordination system, adjacent material handlers, and information systems can help receive shop floor level decision-making and give feedback to these according to the machine tool condition. It can contribute to improving the shop floor's KPIs in terms of time, quality, etc. Based on this vision, we derived the conceptual model shown in Fig. 2.

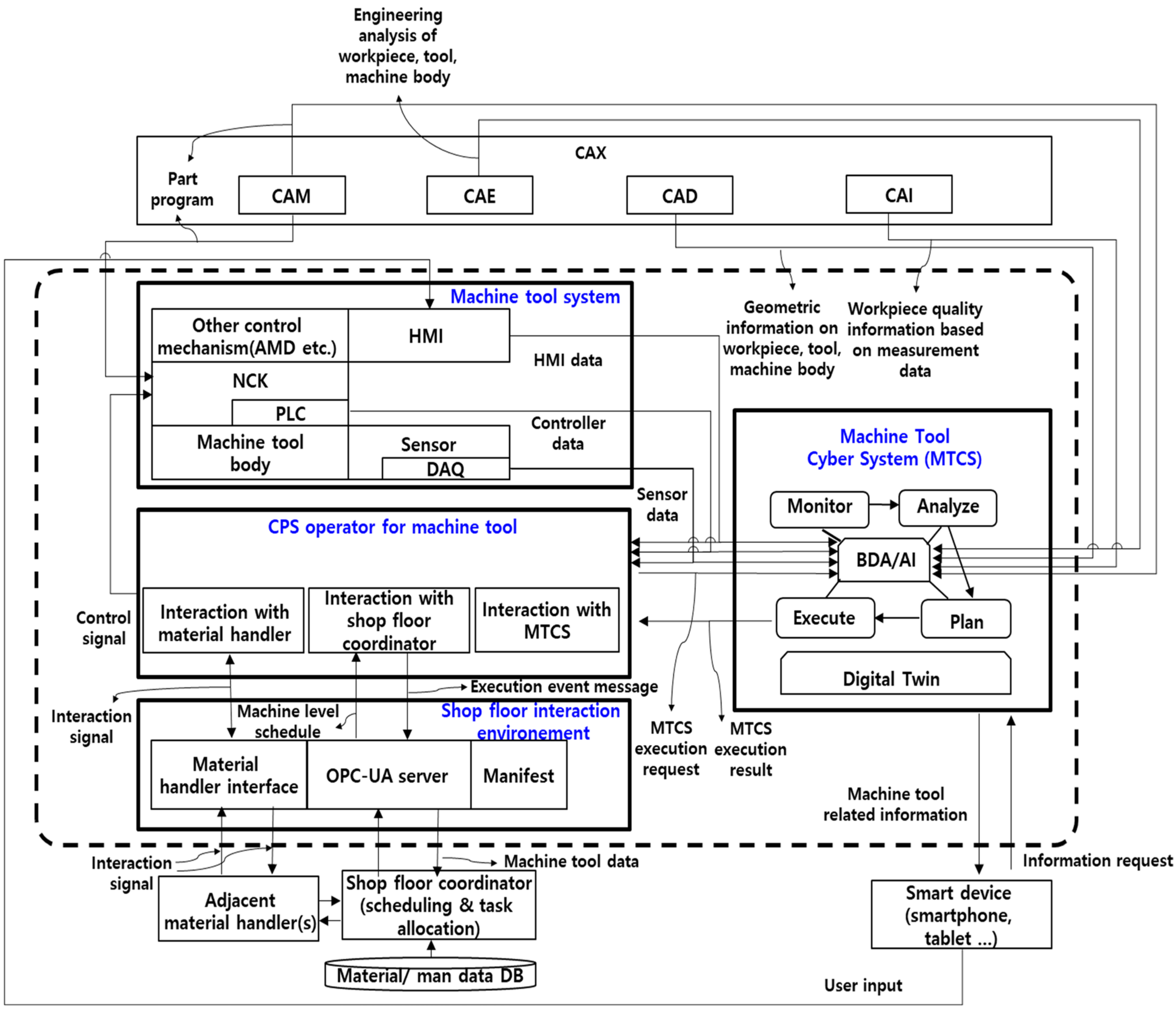


Fig. 2 SMTS conceptual model

The SMTS can be divided into four major subsystems according to functionality type. The machine tool system refers to existing machine tools that do not reflect the Industry 4.0 perspective. It takes part program and control signals from the CPS operator for machine tool and outputs sensor, controller, and HMI data. Machine tool cyber system (MTCS) is the realization of a machine tool level cyber system in the CPMS paradigm. MTCS provides big data analytics and digital twin based monitoring, analysis, planning and execution services to the machine tool. BDA/AI takes the sensor, controller, HMI data, and CAX information (part program, engineering analysis of workpiece, tool, machine body, Geometric information on workpiece, tool, machine body, and workpiece quality information based on measurement data) and generates the refined or high-level information on the machine tool that helps to run MAPE. The MAPE loop runs in terms of resolving abnormalities of the machine body, tool, workpiece, environment, etc., and the output of the MAPE loop goes to the CPS operator for the machine tool, which generates control instructions for the machine tool and adjacent shop floor devices. Humans can access MAPE results via smart devices and based on that information, humans can manipulate the machine tool via the HMI. The shop floor interaction environment is the interface to the shop floor coordinator and adjacent material handling systems. The CPS operator for the machine tool is the subsystem that can interact with the MTCS, shop floor coordinator and adjacent material handling systems. It takes the sensor, controller, HMI data, MTCS execution result, interaction signals, machine-level schedule and outputs control signals for the machine tool, interaction signals for adjacent shop floor devices, MTCS execution requests.

Also, the CPS operator for the machine tool delivers the sensor, controller, HMI data and MTCS execution results to the shop floor coordinator. This paper describes the shop floor coordinator which is defined as a system that receives data from shop floor elements and legacy systems and provides production schedule and equipment allocation plans. Commercial shop floor controllers are examples of this.

Functional design of SMTS

Before deriving the functional architecture, functional modeling based on design considerations and the SMTS vision is required. This paper uses IDEF0 modeling for functional modeling, and Figs. 3, 4 and 5 respectively show the topmost level, first level, and second level functional modeling based on IDEF0 notation.

Figure 3 shows the SMTS inputs, outputs, controls, and mechanisms based on relations with external systems and related Industry 4.0 enabling technologies. For the intelligence of the machine tool itself, CAX information (part program, engineering analysis of workpiece, tool, machine body, Geometric information on the workpiece, tool, machine body, and workpiece quality information based on measurement data) together with CNC, sensor, HMI (including user input) data is needed. For contribution to shop floor operation, signals from adjacent material handlers and shop floor coordinator execution results are needed. Also, raw material and the part program is essential for machining operation. Thus, the above information is the input of the topmost function.

A communication protocol is needed to access machine tool information. In addition, an information model is used

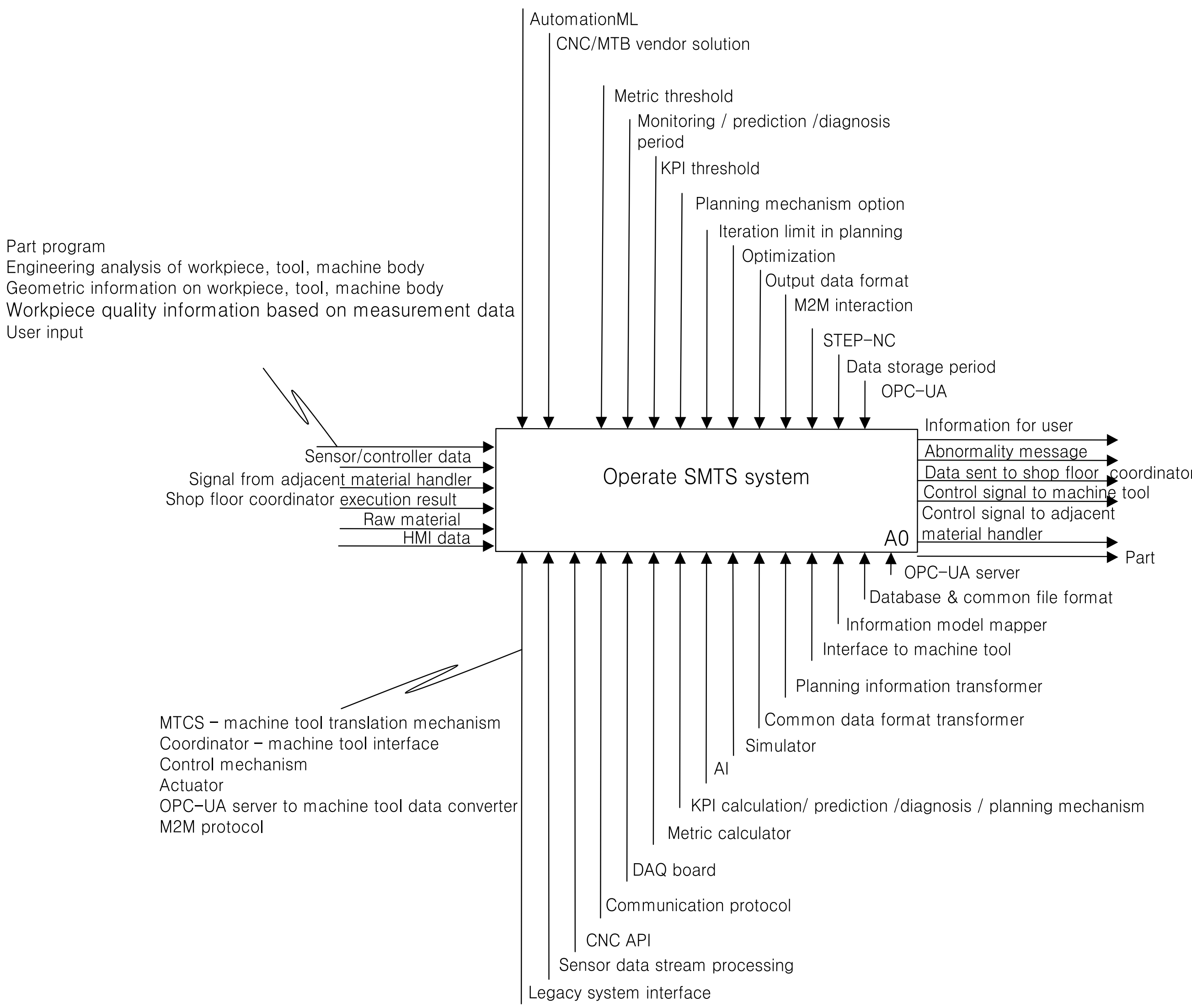


Fig. 3 Topmost level IDEF0 diagram of SMTS

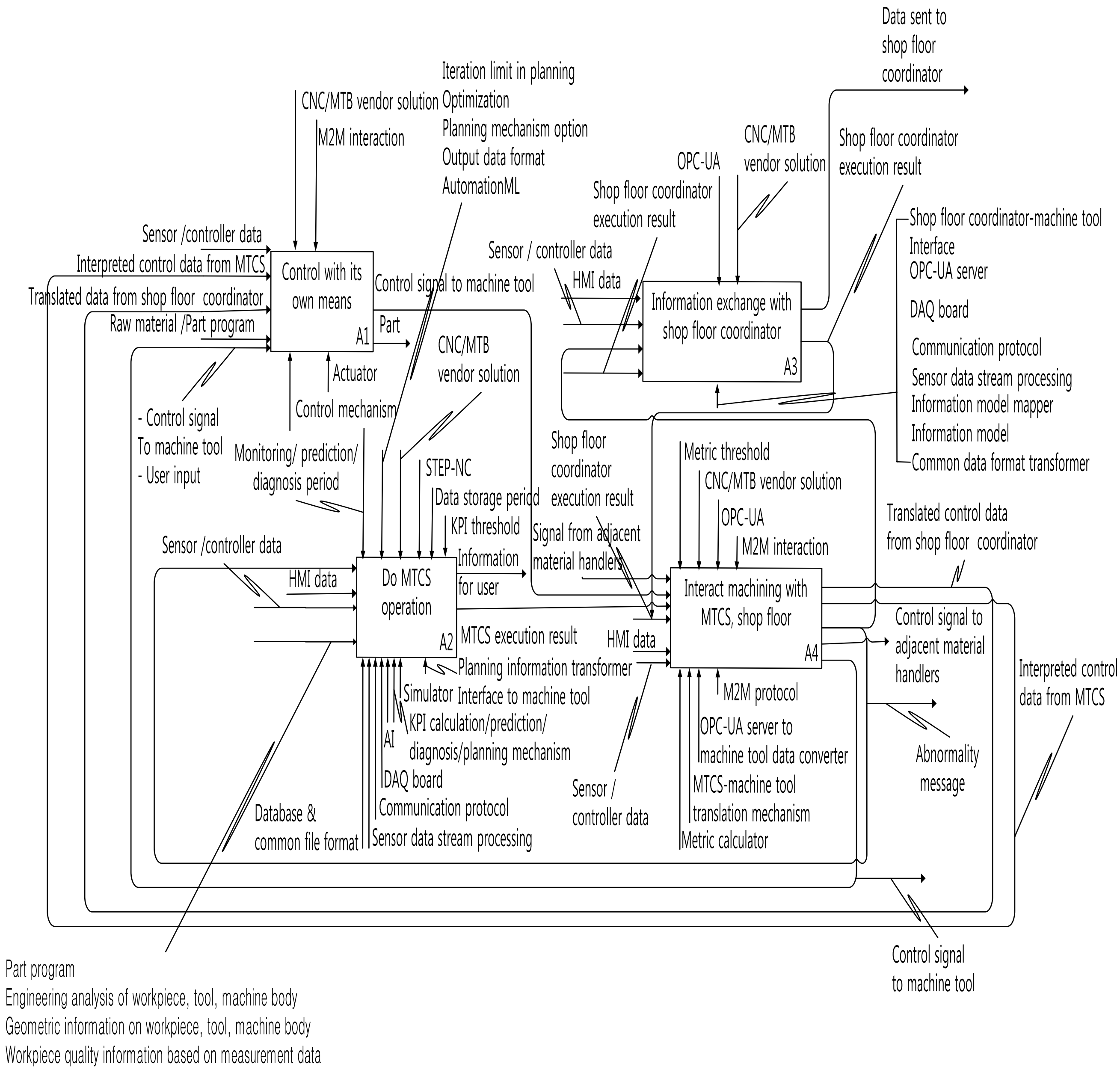


Fig. 4 First level IDEF0 diagram of SMTS

as a common dataset for intelligent functions in SMTS and a means by which external systems can refer to and understand machine tool information. These are the controls of the topmost function of SMTS. Other conditions on MAPE/BD, such as optimization method, iteration limit in planning, KPI thresholds, etc., are also considered as controls of SMTS.

The mechanism for operating the SMTS involves a communication interface means such as legacy system interface, communication protocol, etc., and MTCS functionalities such as KPI calculation/prediction/diagnosis/planning mechanism and their methods. Finally, the outputs which are the result of SMTS operation are machined part, control signals to machine tool as feedback, control signals to adjacent material handlers and data sent to the shop floor coordinator as shop floor interaction, abnormality messages as notifications for defect/failure.

The topmost level function of SMTS can be achieved by the four sub-functions shown in Fig. 4, which are: (1) control with its own means, (2) do MTCS operation, (3) information exchange with shop floor coordinator, (4) interact machining with MTCS, shop floor. These functionalities are aligned with four subsystem functionalities in the conceptual model of SMTS.

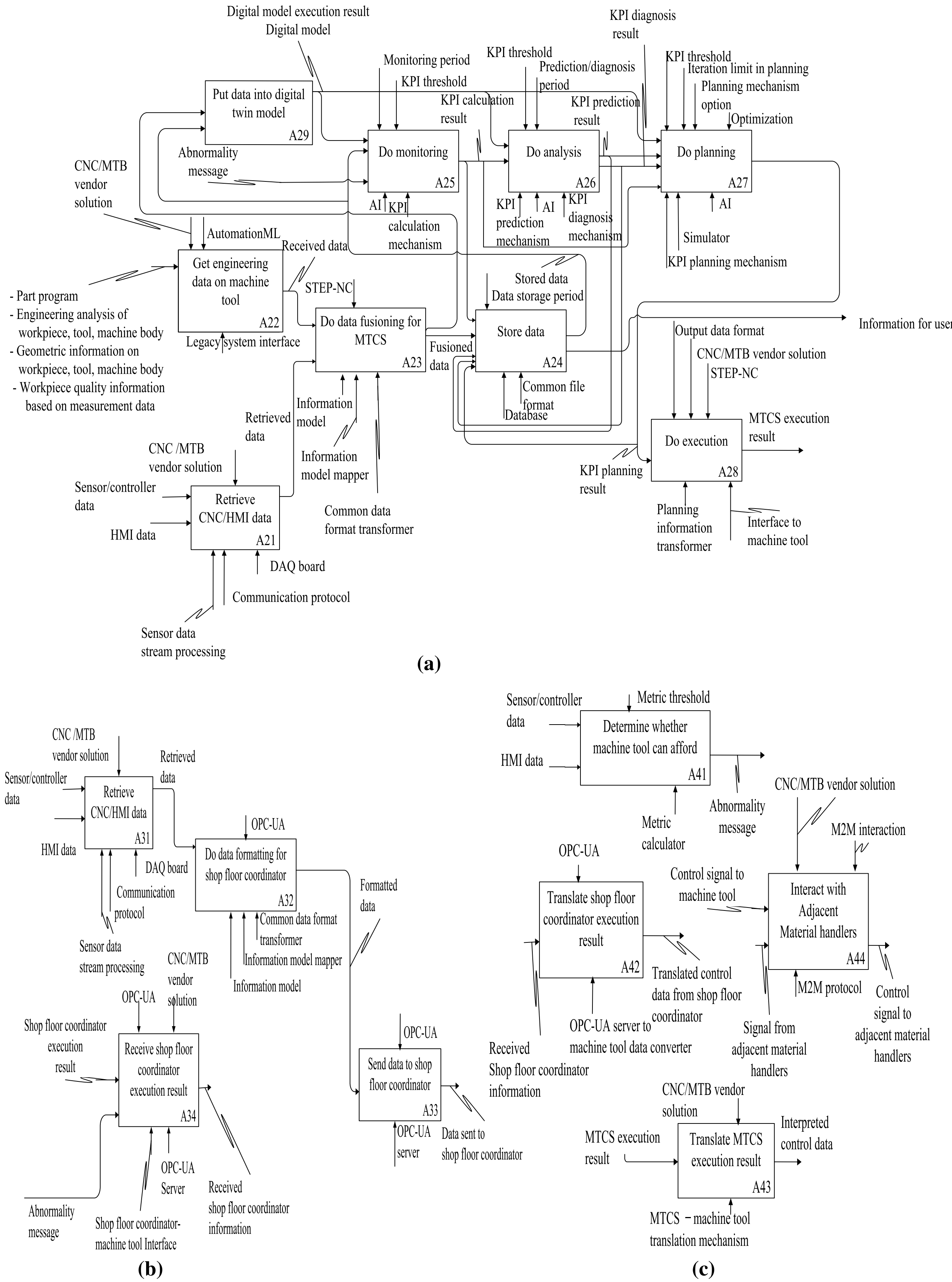


Fig. 5 Second level IDEF0 diagram of SMTS: **a** A2, **b** A3 and **c** A4

The first level IDEF0 functions (A1–A4) can be achieved by the sixteen sub-functions as shown in Fig. 5. The A1 function can be achieved by itself. That is, no detailed functionalities are deployed because machining control itself is something to be included by default. The A2 function can be achieved by the nine sub-functions: (1) retrieve CNC/HMI data, (2) get CAX data on machine tool, (3) do data fusion for MTCS, (4) store data, (5) do monitoring, (6) do analysis, (7) do planning, (8) do execution, (9) put data into digital twin model. The A3 function can be achieved by the four sub-functions: (1) retrieve CNC/HMI data, (2) do data formatting for shop floor coordinator, (3) send data to shop floor coordinator, (4) receive shop floor coordinator execution result. The A4 function can be achieved by the four sub-functions: (1) determine whether the machine tool can afford, (2) translate shop floor coordinator execution result, (3) translate MTCS execution result, (4) interact with adjacent material handlers. Table 2 shows the mapping among second-level IDEF0 block, design consideration item, and related subsystems in the conceptual model to make sure that the resulting IDEF0 model is compatible with the design considerations and conceptual model.

Functional architecture of SMTS

This section derives the SMTS functional architecture based on the conceptual model and IDEF0 modeling derived above and explores the detailed functions. As mentioned in the SMTS vision, the SMTS functional architecture is composed of four subsystems: (1) machine tool system, (2) MTCS, (3) shop floor interaction environment, (4) CPS operator for machine tool. The definition of each subsystem is described here and the architecture in the direction of defining the modules that constitute each subsystem derived. Figure 6 shows the functional architecture of the SMTS.

The details of each subsystem are as follows:

- *Machine tool system* Machine tool system refers to a conventional machine tool that mainly consists of machine tool hardware structure, control unit (CNC, PLC, others), sensor, HMI. Machine tool hardware is a mechanical structure capable of moving a tool in order to machine parts to have the desired shape and surface roughness. The control unit consists of CNC, PLC, and other control mechanisms. The CNC means a controller that interprets the machining program and performs interpolation, position/speed control, error compensation, etc., for moving multiple axes at the same time by providing a servo control signal. The PLC is responsible for machining operation control except for CNC functions such as tool change, coolant on/off, etc. Other control mechanisms refer to control mechanisms such as active mass damper (AMD) (Ganguli et al. 2005) other than CNC and PLC. HMI means the interface between the machine tool and users, providing operation command, status display, part program editor, etc.
- *Machine tool cyber system (MTCS)* MTCS is the realization of the machine tool level cyber system in the CPMS

Table 2 Relationship of components in design consideration, IDEF0 functions, subsystem in the conceptual model

IDEF0 no.	Function name	Related DC id	Related subsystem in conceptual model
A1	Control with its own means	[DCa #3]	Machine tool system
A21	Retrieve sensor/controller data	[DCi #1], [DCi #3]	MTCS
A22	Get engineering data on machine tool/part	[DCi #2]	MTCS
A23	Do data formatting for MTCS	[DCi #5], [DCs #1], [DCi#5]	MTCS
A24	Store data	[DCs #1]	MTCS
A25	Do monitoring	[DCs #2], [DCs #3]	MTCS
A26	Do analysis	[DCs #2], [DCs #4], [DCs #5]	MTCS
A27	Do planning	[DCs #2], [DCs #6], [DCs #7]	MTCS
A28	Do execution	[DCs #2], [DCs #8], [DCi#5]	MTCS
A29	Put data into digital twin model	[DCs #3 ~ 7]	MTCS
A31	Retrieve sensor/controller data	[DCi #1], [DCi #3]	Shop floor interaction environment
A32	Do data formatting for shop floor coordinator	[DCi #5],[DCi #6]	CPS operator for machine tool, shop floor interaction environment
A33	Send data to shop floor coordinator	[DCi #4]	Shop floor interaction environment
A34	Receive shop floor coordinator execution result	[DCa #1]	Shop floor interaction environment
A41	Determine whether a machine tool can afford	[DCa #4]	CPS operator for machine tool
A42	Translate shop floor coordinator execution result	[DCa #1]	CPS operator for machine tool
A43	Translate MTCS execution result	[DCs #8]	CPS operator for machine tool
A44	Interact with shop floor elements	[DCa #2], [DCa #3]	CPS operator for machine tool

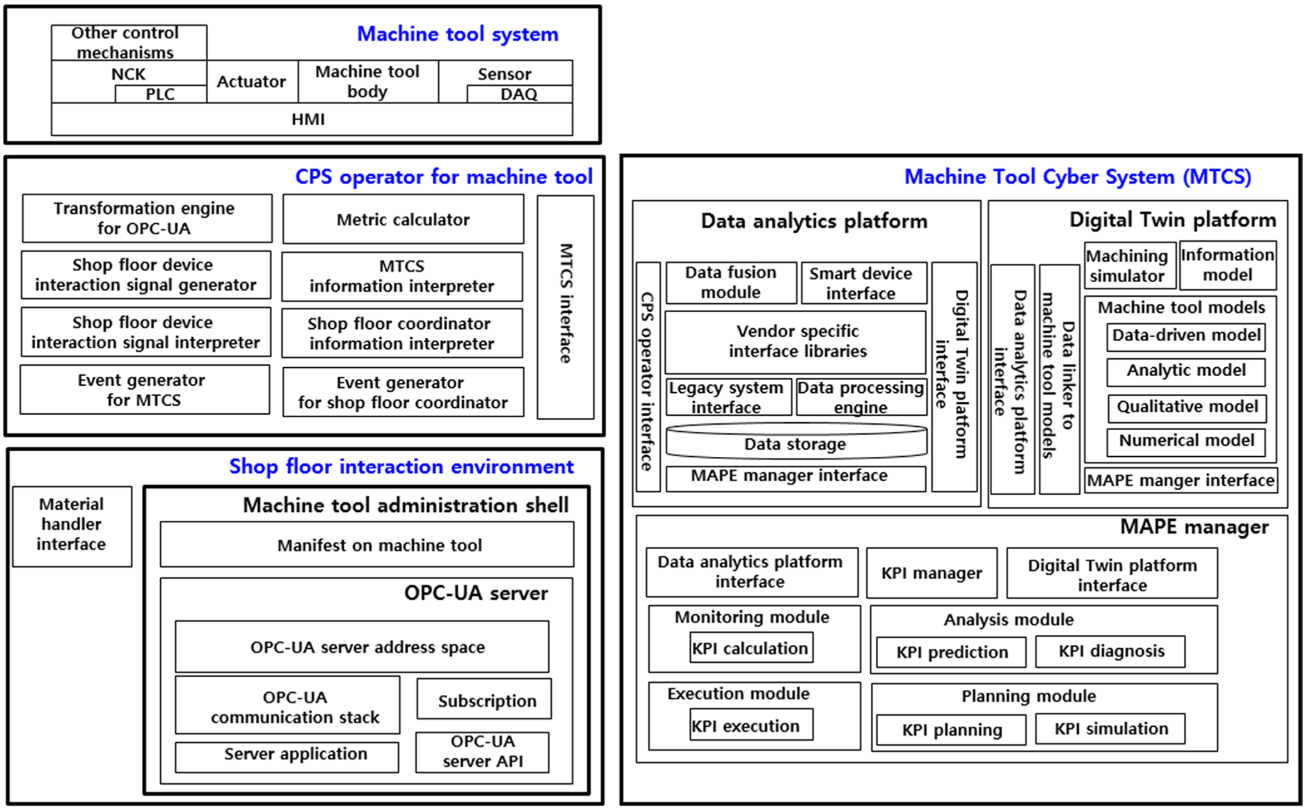


Fig. 6 SMTS functional architecture

paradigm. In other words, it is a software system based on the MAPE/BD framework that is based on the digitalization of the machine tool, applying IoT, ICT, big data, AI, digital twin, etc., which are enabling technologies of Industry 4.0. Through them, various KPIs are improved in terms of machine tool operation toward zero downtime and zero defects. The MTCS can be divided into three major components: (1) MAPE manager, (2) data analytics platform, (3) digital twin platform.

- *MAPE manager* This consists of monitoring, analysis, planning, execution functions, and some other auxiliary functions. Monitoring means calculating KPIs on the machine tool, and determining whether the values lie outside the specified range/standard. Analysis means to provide KPI prediction in future and status diagnosis functions. Planning means re-parametrization of machining in the direction of optimizing the machine tool KPI, and simulation for verification. Execution means delivering decision making from planning to the machine tool. In addition, the KPI manager manages the list of KPIs relevant to the MTCS.
- *Data analytics platform* This refers to the base environment for the utilization of data, which are collected from legacy systems and machine tools. Its functionalities consist of: (1) collecting and storing machine tool data via multi-type interface from the machine tool, (2) collecting machine tool hardware and work-piece part CAX information from legacy systems, (3) data fusion for collected CAX and machine tool data, (4) saving MAPE application execution results, (5) execution environment for MAPE, (6) connection environment with external smart devices, (7) connection environment with digital twin platform.
- *Digital twin platform* This refers to the digital representation and support environment of the machine tool, which is the basis for the improvement of manufacturing system performance. Digital representation means (1) an information model that reflects actual measurement of the machine tool and CAX information of the machine tool, (2) a model describing device behavior in the operational phase from the MAPE point of view, (3) periodical output of updates to the digital behavior model by receiving information from the data analytics platform and putting it into the model. A support environment means a simulation environment that can execute and verify device behavior on the cyber system.

- *CPS operator for machine tool* This refers to the subsystem that receives information and data from the MTCS, shop floor coordinator and material handlers and generates control commands for the machine tool and adjacent material handlers. This subsystem can also check the machining status of the machine tool in the form of quantitative indicators (metrics) such as $\bar{X}$-R, to check whether the machining condition is normal or not and to request execution of the MTCS or shop floor coordinator. Also, the CPS operator for machine tool delivers the sensor, controller, HMI data and MTCS execution result to the shop floor coordinator.
- *Shop floor interaction environment* This refers to the communication environment for interacting with adjacent material handlers and the shop floor coordinator. For interaction with the adjacent material handlers, it includes a material handler interface. For interaction with the shop floor coordinator, it has an OPC-UA server and the manifest on the machine tool. The OPC-UA server consists of: (1) OPC-UA communication stack, (2) OPC-UA server API that isolates the server application from the OPC-UA communication stack, (3) OPC-UA server address for managing machine tool data, (4) subscription for periodically transferring selected data, (5) server application for sending and receiving OPC-UA messages from the OPC-UA client that is in the shop floor coordinator. Manifest on the machine tool is an externally accessible set of information structures so that the shop floor coordinator can recognize the structure of the information model and find data that the shop floor coordinator wants.

Implementation procedure

The SMTS architecture presents the overall functional structure for trying to achieve zero downtime and zero defects of the machine tool and contribution to shop floor operation. However, not all manufacturing companies need to introduce all of the functions all at once when implementing SMTS. Normally, they need some specific functions and services which are highly important for them. Therefore, this section defines the implementation procedure of the SMTS architecture in specific fields in the industry.

This procedure provides activity analysis for a target manufacturer and a small number of views of SMTS for the target manufacturer, which consists of information view, functional view, and operational scenario view. The detailed design of the changed SMTS architecture for a specific company is out of scope since each company has a different application environment. Figure 7 shows the SMTS implementation procedure, which consists of seven steps and some steps which can be performed iteratively depending on conditions. These iterative steps mean that the implemented

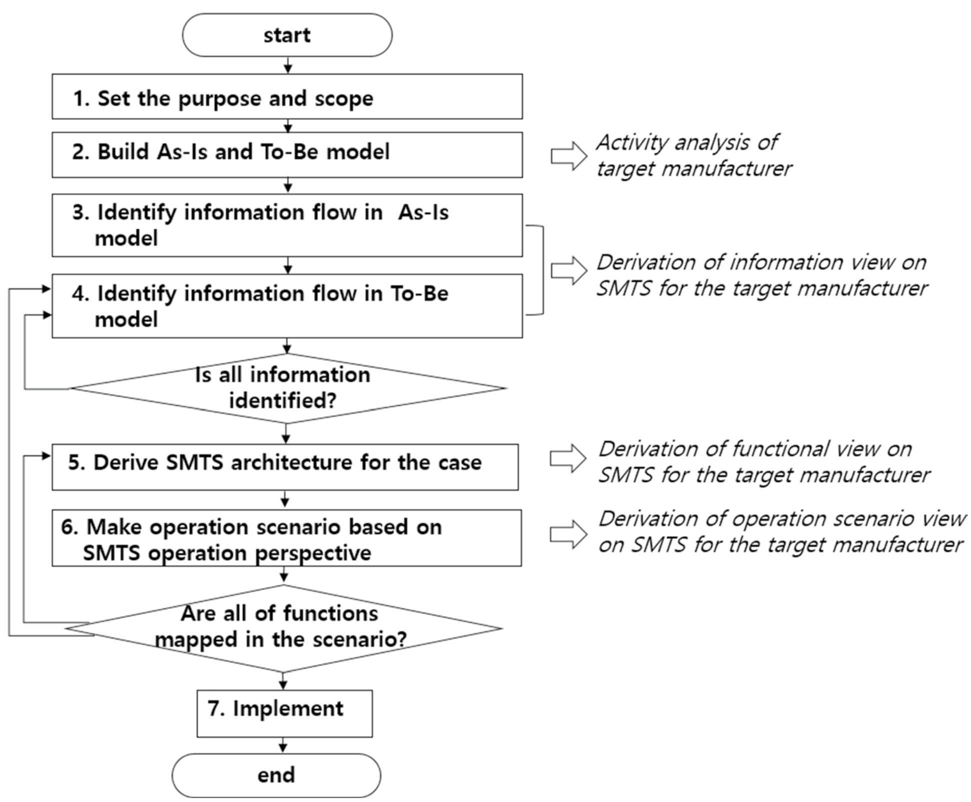


Fig. 7 Implementation procedure

SMTS design is reviewed repeatedly and supplemented to find the necessary functions and information. The following shows the detailed descriptions for each step.

1. *Set the purpose and scope* Define the purpose and scope of the applied SMTS. The purpose is to specify the KPI of interest, such as mean usage time of tool, chatter rate, etc. The scope refers to the machine tool to which it applies and the site where the machine tool is located.
2. *Build the As-Is and To-Be model* The As-Is model describes how activities are performed to achieve the objective described above step in the target field. The To-Be model describes how the objective is achieved after the introduction of SMTS in the target field. Each model has to show the factory element and related activity flow.
3. *Identify information flow in the As-Is model* Identify and describe the information flow for the As-Is model. This shows the information flow involved in As-Is activities.
4. *Identify information flow in the To-Be model* Identify and describe the information flow for the To-Be model. This shows the information flow involved in To-Be activities. Information should be derived in a way that can achieve specified KPI improvement via ICT, IoT, etc. When all information corresponding to the To-Be model is derived, go to step 5. Otherwise, continue with step 4.
5. *Derive SMTS architecture for the case* Define the SMTS architecture corresponding to the To-Be model case. Assign specific MAPE functions corresponding to activities in the To-Be model. In addition, determine whether each function in the SMTS architecture is needed or not for the case.
6. *Make operation scenario based on SMTS operation perspective* It describes the operation scenario that can realize the process described in the To-be model from the perspective of SMTS architecture. While making an operation scenario, the examination on whether all information defined above is reflected in, and whether functions in the architecture are appropriately mapped should be conducted. If either is not satisfied, go back to step 4 and 5.
7. *Implement* Implement SMTS for the target field.

Case study

To show the validity of the SMTS architecture and corresponding implementation procedure, it is necessary to apply these to the specific field and show the advantages. This section develops an SMTS case study on the rotor of an automobile motor manufacturing site by describing the case and applying an implementation procedure.

Case description

Since the rotor is part of the motor, it is necessary to describe the manufacturing of the motor and rotor all at once for holistic understanding, which is shown in Fig. 8a,

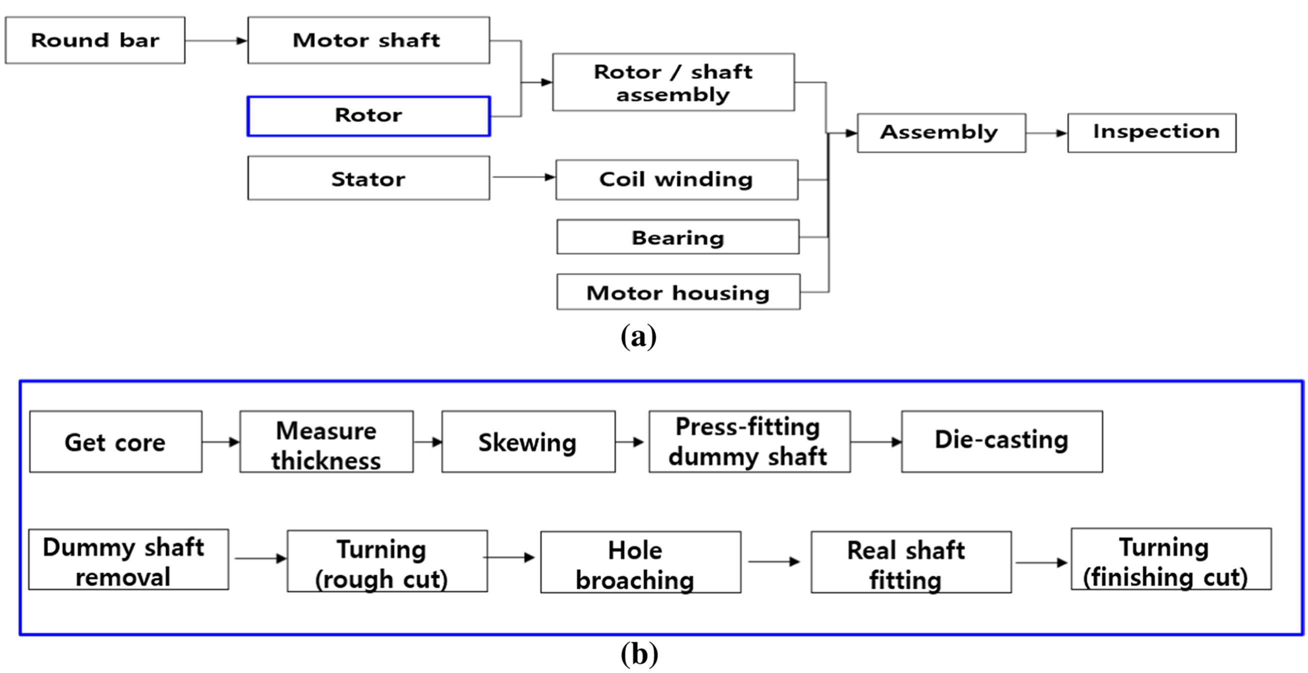


Fig. 8 **a** Motor manufacturing process and **b** rotor manufacturing process

b, respectively. The rotor is manufactured using a die casting process and turning operation. During the turning machining, there is an occasion for downtime due to axis servo motor failure. Currently, the operator manually pushes the emergency stop button after the fault happens, assigns work to another turning machine and performs repair work. Since both detection of the failure symptom and the response in the case of failure are accomplished through the intuition and intervention of the operator, delay to the shop floor production schedule is inevitable and the necessity for prompt response is recognized.

Since it is important to secure a certain amount of rotor production and provide stable supply to the automobile maker, the rotor manufacturer intends to install a function that can detect the failure of the machine tool axis quickly and quickly modify the operation plan in response to the failure. By doing so, the intention is to minimize faults, material and time waste by pre-detecting the axis fault indication and re-allocating tasks to other turning machines.

Application of implementation procedure for the case study

To meet the improvement objective described in Sect. 6.1, this section applies the implementation procedure to that target field. By going through the procedure, related activities are analyzed and the necessary information defined. A modified SMTS architecture is derived related to the information view and activities corresponding to the case manufacturer. Finally, the operational scenario view of the case manufacturer is provided. Each activity in the implementation procedure is described below. The detailed design of the changed SMTS architecture and following implementation for a specific company is out of scope since each company has a different application environment.

1. *Set the purpose and scope* In this case, the first goal is to minimize the downtime of the machine tool itself by predicting/detecting the point where the KPI shows a fault indication by quantifying axis motor health as a KPI in the form of remaining useful life. Remaining useful life (RUL) is defined as the time range from the current time to the end of useful life. Useful life means the time range where the failure rate is relatively consistent. Besides, the second goal aims to provide a countermeasure in the case of failure symptom detection to minimize work delay time compared to the KPI (amount of production delay) standard of shop floor level perspective. The scope of the application is the rotor shop manufacturing process (shop floor) specified in the previous section and the turning machines on the shop floor.
2. *Build the As-Is and To-Be model* This section builds the As-Is and To-Be models for the target field, as shown in Figs. 9 and 10, respectively. Each model includes the main activity flow and interaction among the constituent elements. In the As-Is model, machine tool operators, maintenance engineers and production managers with the shop floor coordinator play a major role and show the process of solving problems according to their intuition/experience and mutual consultation using the shop floor coordinator. In the To-Be model, instead of the operator's real-time observation and action, abnormalities in the tool axis health are detected, predicted, diagnosed by the MTCS and CPS operators in the SMTS. If the current tool axis motor is faulty, then an emergency message is generated. When the current tool axis motor is not in a fault state but there is an indication of degradation, then the rpm of the motor is reduced to alleviate motor load and operation is stopped when the work is done. In addition, an alternative production plan considering the problematic machine tool is proposed from the shop floor coordinator and dispatching is done automatically. To sum up the above, the comparison of each activity in the As-Is and To-Be models is conducted as shown in Table 3.
3. *Identify information flow in As-Is model* The whole process is conducted with physical shop floor devices, shop floor coordinator and workers who manipulate and manage them, as shown in Fig. 11a. Information given from the legacy system to the shop floor coordinator (d1) is CAPP, operation schedule, maintenance schedule. Information in d1 is preliminary information for the shop floor coordinator. For making an operation schedule reflecting current shop floor status, data provided by machine tools (d2) and other production machines (d3) are necessary. These consist of task elapsed time, remaining task time, current task id, current status, lot id, work count. Axis motor alarm signal is the data collected from the machine tool in d2.
4. *Identify information flow inTo-Be model* In the To-Be model, the MTCS and CPS operator of SMTS are involved and have an interworking environment with the shop floor coordinator as shown in (Fig. 11b). Since the material handler for the machine tool is a pallet rather than another machine such as a robot, gantry, etc., the exchange of signals with the adjacent material handler is excluded. Information given from the legacy system to the shop floor coordinator (d1) is CAPP, operation schedule, maintenance schedule. Information in d1 is preliminary information for the shop floor coordinator. Information given from the legacy system to MTCS (d2) is motor specification data such as rotation speed-torque characteristics for using in remaining useful life calculation, prediction, and diagnosis, planning. Information in d2 is preliminary information for MTCS. Data provided by the machine tool (d3) are motor current, bear-

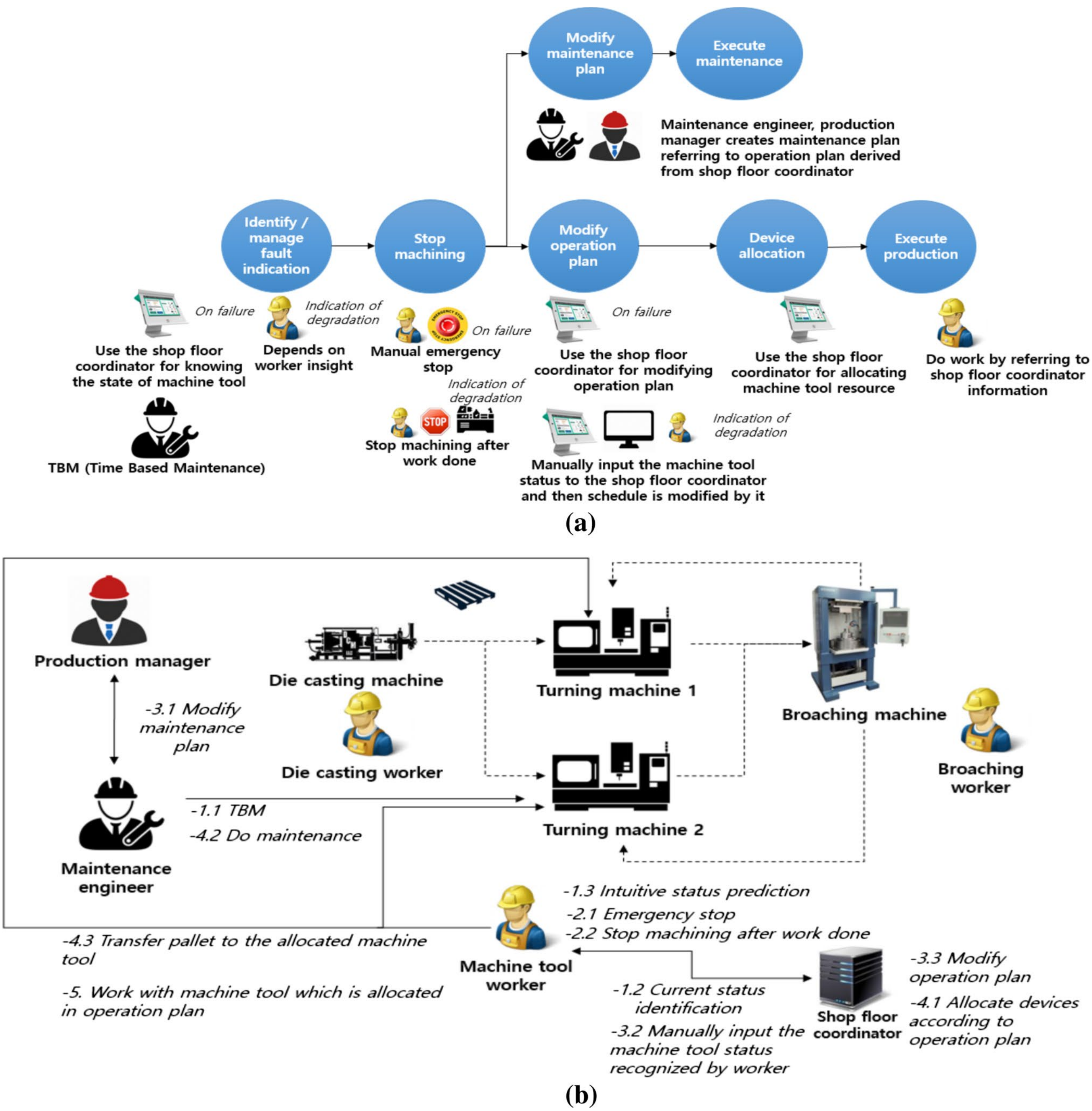


Fig. 9 **a** As-Is process and **b** interaction among elements in As-Is model

ing/motor rotation acceleration, motor temperature, etc. After MTCS execution, it returns emergency stop message, stop message, rpm change message (d4) to CPS operator. The machine tool worker can monitor the currently acquired data and execution result of MAPE functions (d6). CPS operator transforms acquired data from MTCS and control machine tool. Diagnosis result (d8) is shared with the maintenance engineer for providing diagnosis information. In addition, when the abnormality is detected by a rapid metric check by the CPS operator, the CPS operator generates a shop floor coordinator execution request (d5). For making an operation schedule reflecting current shop floor status, data provided by the machine tool (d10) and other production machines (d11) are necessary. These consist of task elapsed time, remaining task time, current task id, current status, lot id, work count. Axis motor alarm signal is the data collected from the machine tool in d10. These data are sent to the shop floor coordinator. The maintenance plan is derived through cooperation with the production man-

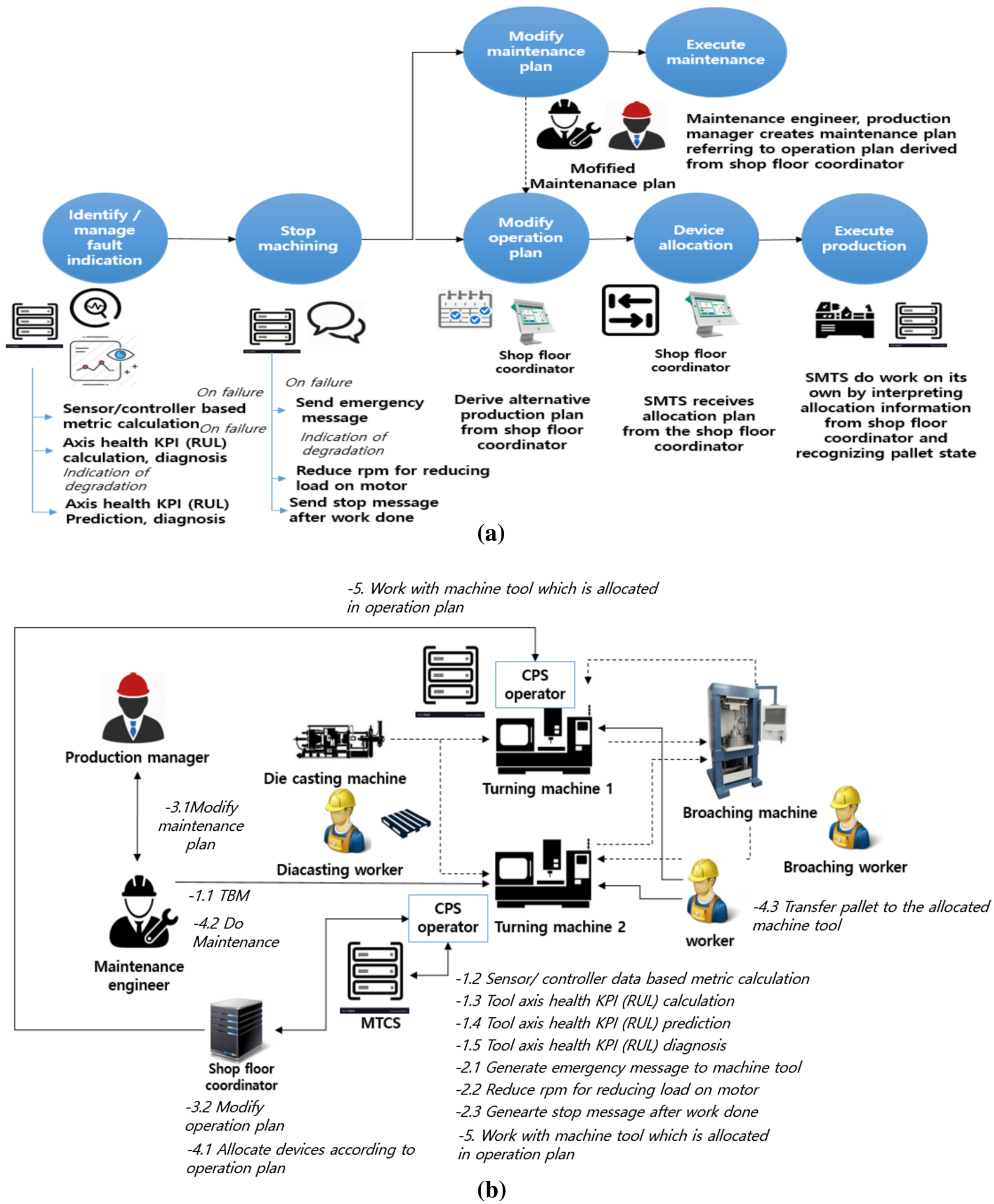


Fig. 10 **a** To-Be process and **b** interaction among elements inTo-Be model

Table 3 As-Is and To-Be model comparison

Activity	As-Is	To-Be	Advantage in To-Be
Identify/manage fault indication	The operator constantly monitors the shop floor coordinator to check the machine tool state Detect indication of degradation based on operator intuition The maintenance engineer carries out time-based maintenance regularly	Rapidly check whether machine tool data exceeds predefined standards or not Do remaining useful life calculations, diagnosis, and prediction to find out whether current or future axis motor state is good or not	It can reduce variation among workers on detecting axis motor faults and signs of degradation It can reduce waste of downtime due to time-based maintenance
Stop machining	Conduct an emergency stop when the axis motor fault happens Stop the machine after work done when the axis motor shows signs of degradation	Generate emergency messages to stop the machine tool automatically when the axis motor fault happens Reduce rpm for reducing the load on the motor, and generate stop message after work is done when axis motor shows signs of degradation	It can contribute to reducing the motor burden when the sign of degradation is detected It can respond rapidly due to automated operation thanks to MTCS and CPS operator
Modify maintenance plan	The maintenance engineer and production manager refer to the production plan from the shop floor coordinator and modify the maintenance plan	The maintenance engineer and production manager refer to the production plan from the shop floor coordinator and modify the maintenance plan	–
Execute maintenance	The maintenance engineer performs maintenance on the failed machine tool	The maintenance engineer performs maintenance on the failed machine tool, referring to diagnosis results from MTCS	Sharing diagnosis results from MTCS can help to reduce MTTR
Modify operation plan	The shop floor coordinator detects only the faulty state of the machine tool and modifies the schedule When the worker detects signs of degradation, manual input to shop floor coordinator should be done	The shop floor coordinator detects the faulty state and signs of degradation on the machine tool and modifies the schedule	The shop floor coordinator can automatically reflect signs of degradation on the machine tool, reducing shop floor level response time
Device allocation	The shop floor coordinator assigns the work to resources according to the changed operation plan The worker can view that information via the monitor	The shop floor coordinator assigns the work to resources according to the changed operation plan SMTS can receive that information	SMTS can interpret the device allocation information, reducing shop floor level response time
Execute production	The operator moves the pallet placed in the failed machine to the replacement machine tool The worker does work by referring to the shop floor coordinator monitor	The operator moves the pallet placed in the failed machine to the replacement machine tool SMTS does work on its own by interpreting allocation information from shop floor coordinator and recognizing pallet state	SMTS can interpret the device allocation information and operate automatically It can contribute to reducing shop floor level response time

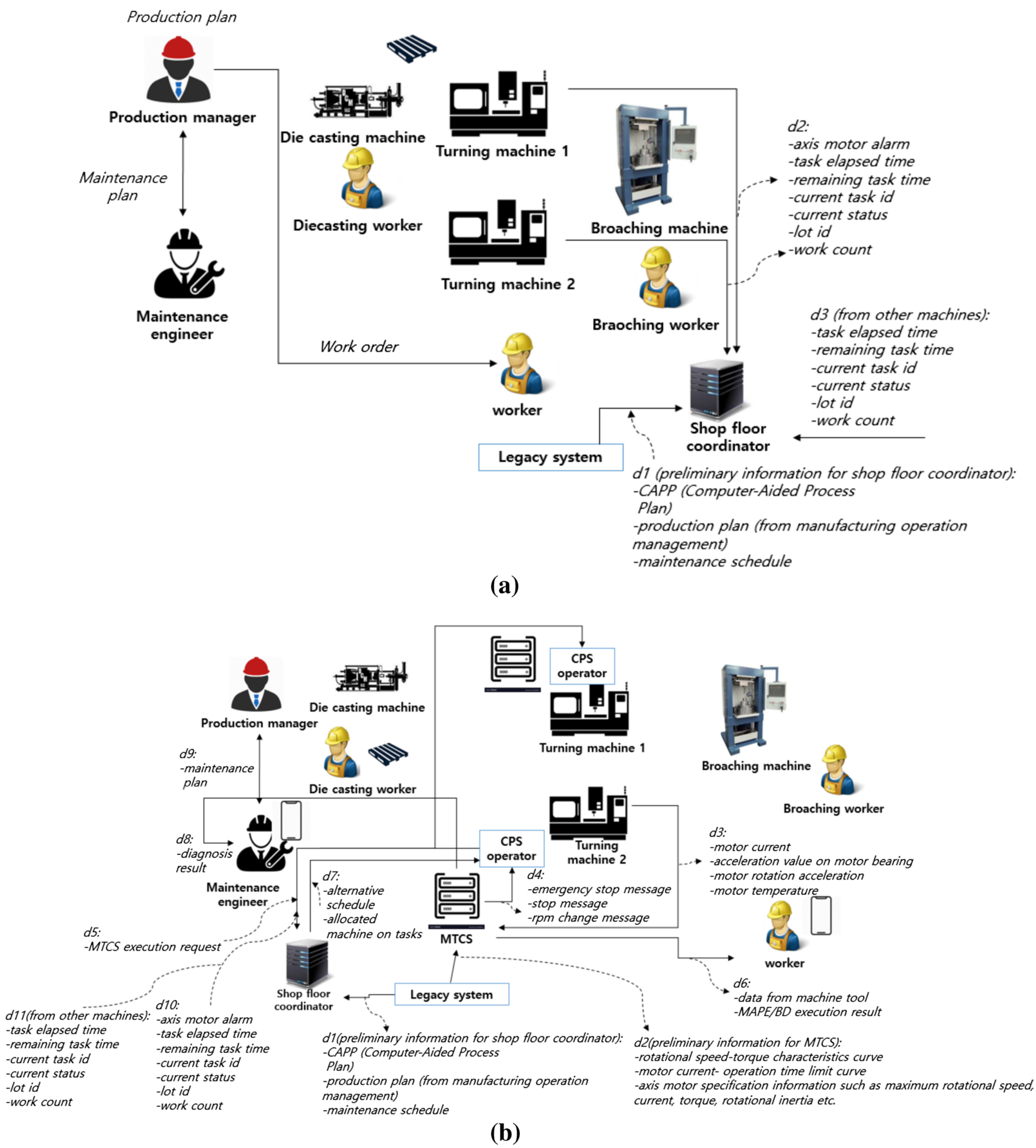


Fig. 11 **a** Information flow of As-Is model and **b** Information flow of To-Be model

ager and maintenance engineer (d9). After the execution of the shop floor coordinator, an alternative schedule and allocated machine on tasks are distributed all around the shop floor (d7).

5. *Derive SMTS architecture for the case* Based on derived functions and related information, this paper derives SMTS architecture corresponding to the case as shown in Fig. 12. Because there is no interaction with adjacent material handling machines, the communication environment in the shop floor interaction environment and ability to interpret and generate these signals in the CPS operator have been removed. Moreover, since there is no

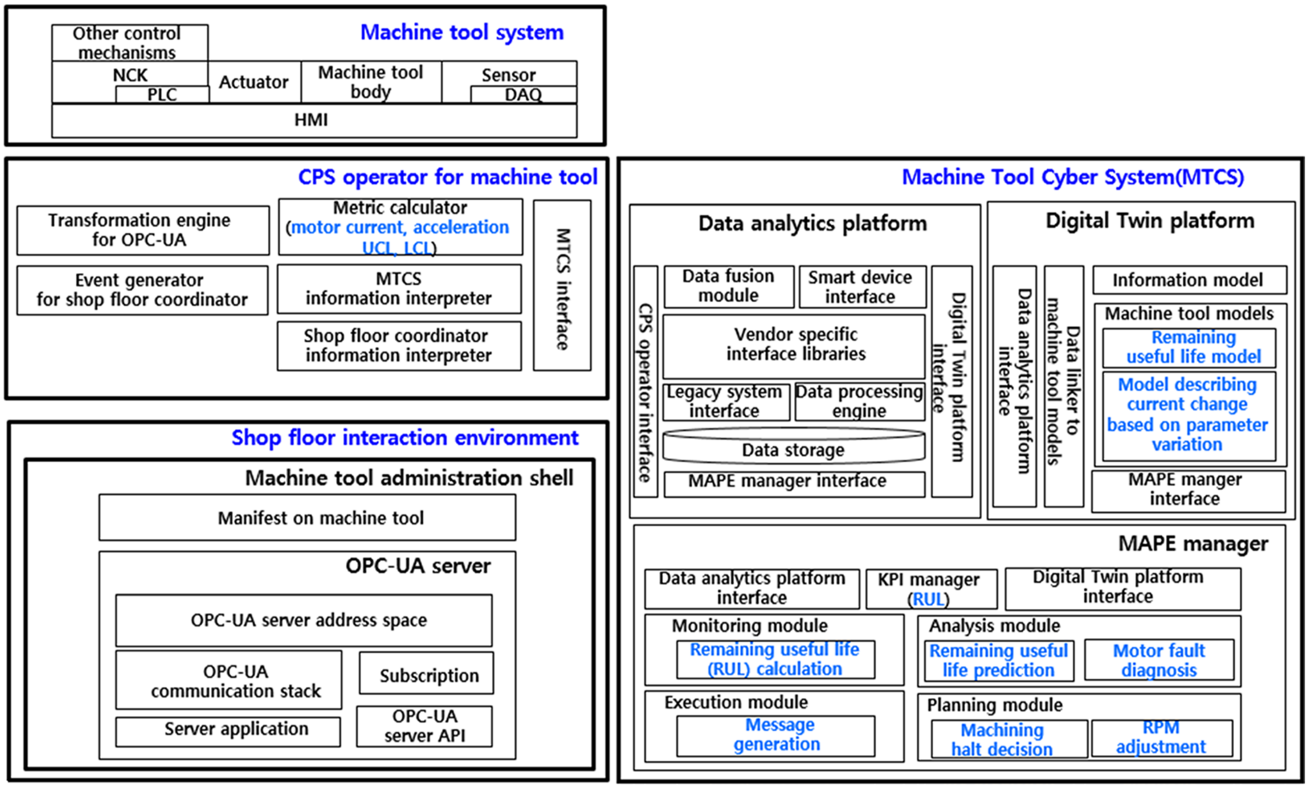


Fig. 12 SMTS functional architecture for the case manufacturer

necessity for requesting MTCS execution when motor axis abnormality is detected in the metric calculation, the event generation for MTCS is removed. The simulation function is also removed in MTCS because there is no need to do a simulation in planning. Instead, we can see that the MAPE function, the corresponding KPI, and the metric in the CPS operator are embodied as things related to tool axis health. Here, the remaining useful life model in digital twin is used in the remaining useful life calculation and prediction function in the analysis. The model describing current change based on parameter variation is used in motor fault diagnosis and planning by inputting various variations of parameter sets.

6. *Make operation scenario based on SMTS operation perspective* In steps 2 and 4, Activity flow and information elements are derived from the manufacturing site's point of view. In this step, the operation scenario from the functional perspective of SMTS is described. In the operation scenario, two perspectives exist. Figure 13a refers to the machine tool in which the failure occurs and Fig. 13b refers to the replacement machine tool. In practice, the throughput in the die casting process is greater than that of the machine tool, which means that there exist more machine tools. However, many machine tools are represented by only two machine tools because they have the same characteristics in terms of receiving the production plan modified by the shop floor coordinator. Table 4 shows a description of each step in the operation scenario. In this operation scenario, we assume that modifying the operation schedule involves identifying an alternative machine tool.

Conclusions

In this paper, we derive a machine tool architecture conforming to Industry 4.0 starting from stakeholder requirements derived through interviews. These requirements are translated into design considerations, largely in terms of Industry 4.0 components, machine tool self-intelligence and contribution to autonomous operation. These design considerations have been the basis for the derivation of the vision and functional modeling of the SMTS, and mapping work among them is conducted to make sure that all of the things are closely related. The functional architecture for SMTS is derived by specifying the four main functions in the SMTS conceptual model with functional modeling. To apply SMTS in the real industry, this paper shows an implementation procedure and applies this to the rotor manufacturing industry. The application of this case shows how the SMTS is applied to industry and, in this case, the

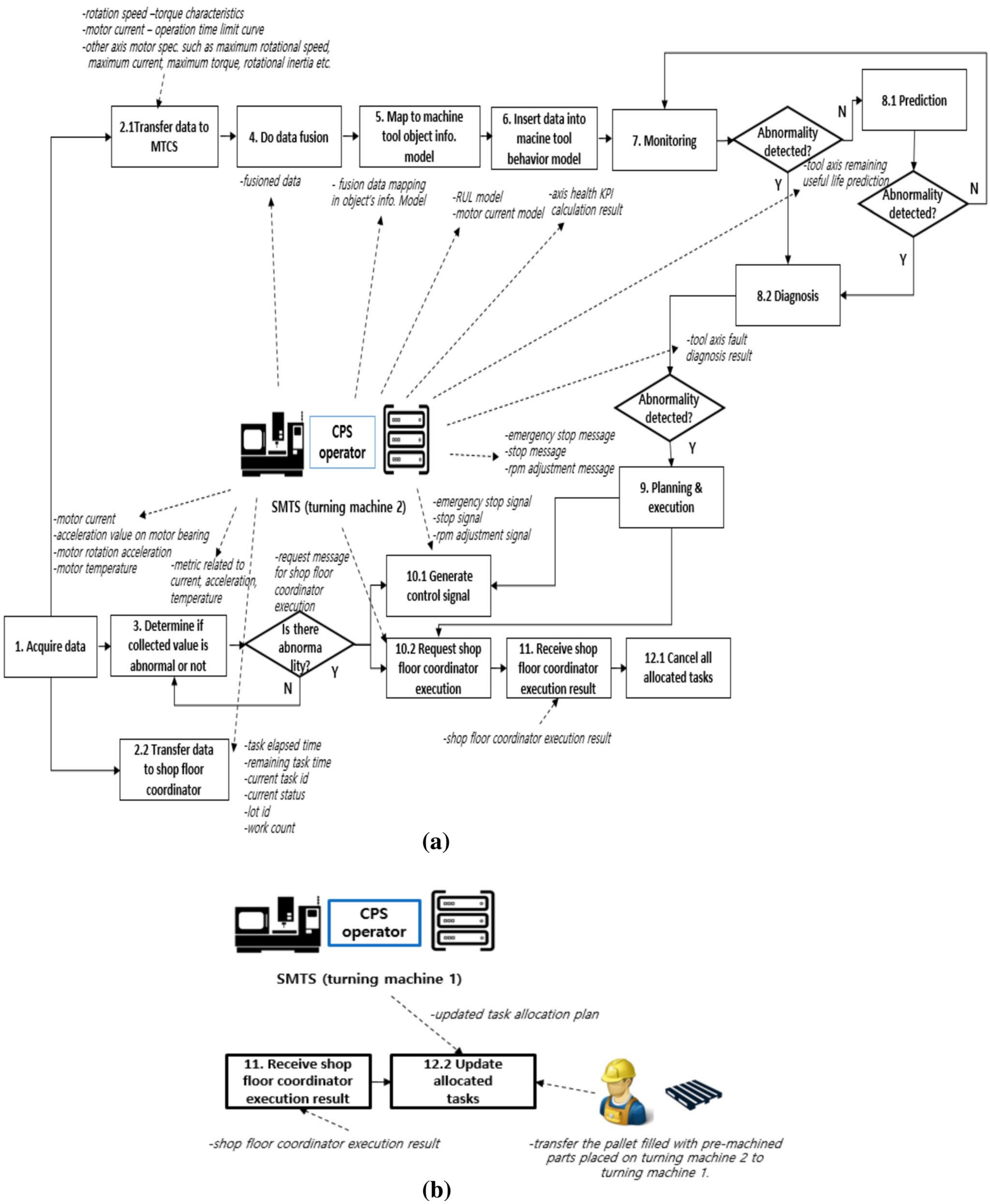


Fig. 13 **a** Operation scenario for failed SMTS and **b** operation scenario for replaced SMTS

Table 4 Operation scenario description

Machine tool	No.	operation	Description
Turning machine 2	1	Acquire data	Acquire motor current, acceleration value of bearing and rotation, motor temperature, task elapsed time, remaining task time, current task id, status, lot id, work count, etc. from CNC, sensor, HMI
	2.1	Transfer data to MTCS	The collected motor current, acceleration, and temperature values are communicated to the MTCS. In addition, the MTCS receives tool axis motor specification data such as motor speed-torque characteristics, motor current—operation time limit curve, etc.
	2.2	Transfer data to shop floor coordinator	The collected task elapsed time, remaining task time, current task id, status, lot id, work count, etc., are sent to the shop floor coordinator using the CPS operator
	3	Determine if collected value is abnormal or not	Measure whether the collected signal meets a predefined criterion and check rapidly if there exists abnormality
	4	Data fusion	Perform pre-processing, formatting, and grouping of collected machine tool and engineering data
	5	Map to machine tool object information model	Mapping fusion data to machine tool object model based on information modeling in digital twin
	6	Insert data into machine tool behavior model	Periodically update the behavior model output by inserting data from the mapped machine tool object model into the behavior model
	7	Monitoring	Calculate remaining useful life to determine if the current state is good/potentially dangerous/dangerous
	8.1	Prediction	Predict the remaining life of the tool axis motor
	8.2	Diagnosis	Identify the cause of the current failure or sign of degradation
	9	Planning and execution	If the remaining service life is less than the reference value, stop message after work is done is generated, and rpm adjustment instruction is provided. If failure is detected, the emergency stop message is issued
	10.1	Generate control signal	If the collected signal exceeds a predefined criterion, the machining operation is stopped and the alarm is generated. When the MTCS response is received, the CPS operator interprets it and then generates control signals
	10.2	Request shop floor coordinator execution	If the collected signal exceeds a predefined criterion or if the stop command is delivered from MTCS, request shop floor coordinator execution
	11	Receive shop floor coordinator execution result	Receive the results of the shop floor coordinator's execution and convert them into a machine-understandable form
	12.1	Cancel all allocated tasks.	Cancel all tasks allocated in turning machine 2
Turning machine 1	11	Receive shop floor coordinator execution result	Receive the results of the shop floor coordinator's execution and convert them into a machine-understandable form
	12.2	Update allocated tasks	The work plan placed on turning machine 2 is transferred to turning machine 1

effect of the shortening the response time and reduction of the deviation among the workers can be confirmed.

To realize SMTS, an environment that can interact with an existing machine tool and MTCS, machine tool data fusion infrastructure in MTCS, machine tool-specific digital twin model, interface to shop floor coordinator and adjacent material handler should be developed. Especially, the development of the digital twin model in MTCS is the most important subsystem because it is a key subsystem that greatly influences MAPE function, information model and interaction with the machine tool unit. Furthermore, research on shop floor coordinators and their interaction with SMTS will be conducted.

Based on the functional architecture of this paper we are working on the realization of the machine tool digital twin, which aims to contribute to expanding the usage scope and depth of SMTS. We are also developing a prototype that can represent the validity of the architecture on the case study in Sect. 6. Furthermore, research for developing a functional architecture and prototype for other shop floor devices compatible with Industry 4.0 is actively pursued.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (NRF-2019R1A2C1004388)

References

Bitkom, VDMA, ZWEI. (2016). *Implementation Strategy Industrie 4.0.*

Brennan, R. W., Balasubramanian, S., & Norrie, D. H. (1997). Dynamic control architecture for metamorphic control of advanced manufacturing systems. In *Architectures, networks, and intelligent systems for manufacturing integration* (Vol. 3203, pp. 213–224). International Society for Optics and Photonics.

Çaydaş, U., et al. (2012). Support vector machines models for surface roughness prediction in CNC turning of AISI 304 austenitic stainless steel. *Journal of Intelligent Manufacturing, 23*(3), 639–650.

Chandrasekaran, M., Muralidhar, M., Krishna, C. M., & Dixit, U. S. (2010). Application of soft computing techniques in machining performance prediction and optimization: A literature review. *The International Journal of Advanced Manufacturing Technology, 46*(5–8), 445–464.

Colombo, A. W. (1998). *Development and implementation of hierarchical control structures of flexible production systems using high level petri nets*. Bamberg: Meisenbach.

Dong, X., Jian-qu, Z., & Feng, W. (2012). Fuzzy PID control to feed servo system of CNC machine tool. *Procedia Engineering, 29,* 2853–2858.

Gajate, A., et al. (2012). Tool wear monitoring using neuro-fuzzy techniques: A comparative study in a turning process. *Journal of Intelligent Manufacturing, 23*(3), 869–882.

Ganguli, A. B. H. I. J. I. T., Deraemaeker, A., Horodinca, M. I. H. A. I. T. A., & Preumont, A. (2005). Active damping of chatter in machine tools-demonstration with a 'Hardware-in-the-Loop'simulator. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 219*(5), 359–369.

Gou, L., Luh, P. B., & Kyoya, Y. (1998). Holonic manufacturing scheduling: Architecture, cooperation mechanism, and implementation. *Computers in Industry, 37*(3), 213–231.

Goyal, D., & Pabla, B. S. (2015). Condition based maintenance of machine tools: A review. *CIRP Journal of Manufacturing Science and Technology, 10,* 24–35.

Grangel-Gonzalez, I., Halilaj, L., Coskun, G., Auer, S., Collarana, D., & Hoffmeister, M. (2016). Towards a semantic administrative shell for industry 4.0 components. In *IEEE tenth international conference on semantic computing (ICSC)* (pp. 230–237). IEEE.

IBM. (2005). *An architectural blueprint for autonomic computing*. White paper.

IEC TC65/SC65E. (2016). *OPC unified architecture—Part 1: Overview and concepts.*

IEC TC65/SC65E. (2018). *Engineering data exchange format for use in industrial automation systems engineering—Automation markup language—Part 1: Architecture and general requirements*.

Khorasani, A. M., Yazdi, M. R. S., & Safizadeh, M. S. (2012). Analysis of machining parameters effects on surface roughness: A review. *International Journal of Computational Materials Science and Surface Engineering, 5*(1), 68–84.

Kouiss, K., et al. (1997). Using multi-agent architecture in FMS for dynamic scheduling. *Journal of Intelligent Manufacturing, 8*(1), 41–47.

Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence, 22*(7), 979–991.

Li, B. (2012). A review of tool wear estimation using theoretical analysis and numerical simulation technologies. *International Journal of Refractory Metals & Hard Materials, 35,* 143–151.

Li, Y., Zhao, W., Wu, W., Lu, B., & Chen, Y. (2014). Thermal error modeling of the spindle based on multiple variables for the precision machine tool. *The International Journal of Advanced Manufacturing Technology, 72*(9–12), 1415–1427.

Lin, G. Y. J., & Solberg, J. J. (1992). Integrated shop floor control using autonomous agents. *IIE Transactions, 24*(3), 57–71.

Liu, C., & Xu, X. (2017). Cyber-physical machine tool-the era of machine tool 4.0. *Procedia CIRP, 63,* 70–75.

Mellal, A., et al. (2016). Parameter optimization of advanced machining processes using cuckoo optimization algorithm and hoopoe heuristic. *Journal of Intelligent Manufacturing, 27*(5), 927–942.

Quintana, G., & Ciurana, J. (2011). Chatter in machining processes: A review. *International Journal of Machine Tools and Manufacture, 51*(5), 363–376.

Ruijun, L., Wenhua, Y., Zhang, H. H., & Qifan, Y. (2012). The thermal error optimization models for CNC machine tools. *The International Journal of Advanced Manufacturing Technology, 63*(9–12), 1167–1176.

Schild, K., & Bussmann, S. (2007). Self-organization in manufacturing operations. *Communications of the ACM, 50*(12), 74–79.

Shen, W., et al. (2000). MetaMorph II: An agent-based architecture for distributed intelligent design and manufacturing. *Journal of Intelligent Manufacturing, 11*(3), 237–251.

Siddhpura, M., & Paurobally, R. (2012). A review of chatter vibration research in turning. *International Journal of Machine Tools and Manufacture, 61,* 27–47.

Siddhpura, A., & Paurobally, R. (2013). A review of flank wear prediction methods for tool condition monitoring in a turning process. *The International Journal of Advanced Manufacturing Technology, 65*(1–4), 371–393.

Singh, K., Vasant, P., Elamvazuthi, I., & Kannan, R. (2015). PID tuning of servo motor using bat algorithm. *Procedia Computer Science, 60,* 1798–1808.

Sousa, P., & Ramos, C. (1999). A distributed architecture and negotiation protocol for scheduling in manufacturing systems. *Computers in Industry, 38*(2), 103–113.

Suh, S.-H., et al. (2008). *Theory and design of CNC systems*. Berlin: Springer.

Suh, S.-H. (2017). Insight of smart factory from CNC to CPSoS. *Int'l symposium on smart factory, POSTECH*.

Suh, S. H., & Cheon, S. U. (2002). A framework for an intelligent CNC and data model. *The International Journal of Advanced Manufacturing Technology, 19*(10), 727–735.

Tangjitsitcharoen, S., et al. (2015). Advance in chatter detection in ball end milling process by utilizing wavelet transform. *Journal of Intelligent Manufacturing, 26*(3), 485–499.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry, 37*(3), 255–274.

Wagner, C., et al. (2017). The role of the Industry 4.0 asset administration shell and the Digital Twin during the life cycle of a plant. In 22nd *IEEE international conference on emerging technologies and factory automation (ETFA)* (pp. 1–8). IEEE.