Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Master Thesis No. 0838-009

Goal-driven Context-sensitive Production Processes using BPMN - A Case Study

Debasis Kar

Course of Study:	INFOTECH
Examiner:	Prof. Dr. Dr. h. c. Frank Leymann
Supervisor:	M.Sc. C. Timurhan Sungur
Commenced:	September 09, 2015
Completed:	March 10, 2016
CR-Classification:	H.4.1, J.7, K.1

Abstract

Industry 4.0, a brainchild of the German Federal Government, often referred as the Fourth Industrial Revolution, predicts that Smart Factories driven by Internet of Things (IoT) and Cyber-Physical Systems (CPS), will reinvent the traditional manufacturing industry into a digitalized, context-aware and automated manufacturing that will flourish with contemporary Information and Communication Technology (ICT). As the IoT are being deployed across production cites of the manufacturing companies, the need of decision making inside a business process based upon the received contextual data from the execution environment has transpired. Context-sensitive Execution Step (CES) is such an adept concept that illustrates how a business process can be Context-sensitive and keep itself aligned with the abstract business goals of the company.

Realization of the CES based business processes, however, requires a model-driven approach. Being Business Process Model and Notation (BPMN), the de-facto standard for business processes modeling, business analysts of manufacturing companies use it to model and execute business processes in a model-driven approach. In this work, first we analyze the properties of intelligent production processes that discern them from traditional production processes. Following that we scrutinize the properties of BPMN that need to be preserved to keep the consensus that BPMN proposes. Subsequently, we derive properties that are most relevant for modeling CES integrated processes in BPMN.

From the deduced properties, we infer the requirements for modeling CES integrated business processes and propose extensions to standard BPMN. Developed extensions include a new type of service task and a new type of process definition that are technology agnostic. Finally, we discuss the existing state of the art alternatives to compare our approach with those and show why our approach provides a more comprehensive and suitable solution that makes production processes Context-sensitive as well as Goal-driven in unison.

Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

1. Introduction

Since manufacturing processes involve the processes to transform raw materials into finished products exhaustively; economists around the globe assert manufacturing to be a wealth-producing sector of an economy as most developed economies are predominantly dependent upon manufacturing sector. Westkämper [Wes06] suggests that the focus of supply and demand is shifting due to modern innovations in Information and Communication Technology (ICT). Thus manufacturing industry is experiencing more complicated supply chains than before. Customers demanding high levels of individualized products are driving fierce competitions in pricing and compelling manufacturers to endeavor highest levels of efficiencies. The kind of turbulences a manufacturer can expect to be adapted easily using automated processes are shown in Figure 1.1. Furthermore manufacturers can develop effective survival strategies in midst of all these turbulences, if they are able to continuously adapt their organizational structures, technical changes and learn from their previous mistakes [Wes06].

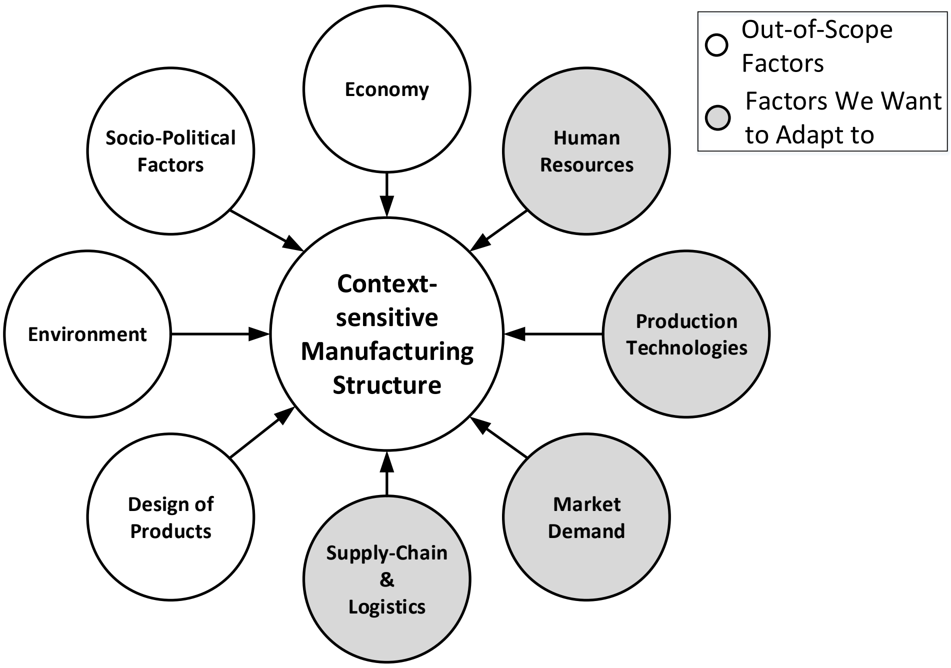

Figure 1.1.: Origins of Turbulences in Manufacturing (Adapted from [Wes06])

According to Graybill [Gra14], adaptive smart manufacturing thrives upon many challenges, e.g., accessing all available information when it is needed, where it is needed, and how it is needed to deduce optimal actions and responses. Adaptive manufacturing facilitates manufacturers to generate and apply data-driven manufacturing intelligence throughout the life-cycle of design, procurement, planning, production, and logistics [RLG$^+$15, Erl12].

The advent of a new wave of technological changes has already started driving a paradigm shift in manufacturing. Manufacturing sector is at the verge of a new industrial revolution which assures smarter industrial production process flow, optimized new business models and highly tailor-made products. The new technological wave builds on the concept of interaction between the real and virtual worlds which in turn becomes the core of the manufacturing processes [RLG+15, HPO15].

According to Erlach [Erl12], the ultimate intent of business process is known as the *"Holy Trinity"* of cost, quality, and time. Basu [Bas14] refers this trio as *"Iron Triangle"* whereas Erlach [Erl12] further adds variability or changeability to the aforementioned three intents. Achievement of production goals are dependent upon factors such as low production costs, high quality of products, short lead times in production and order processing, and moreover the product variety on demand. Products within the lowest price segment can only be distinguished from competition by innumerable individualized (customized) products [Erl12].

Erlach [Erl12] further explains the interrelation between the four goal dimensions and the relevant goal conflicts as shown in Figure 1.2. In a nutshell, the possible relationships between the four goal dimensions can be characterized into following four types:

1. The contradictory antagonism of goals is the strongest type of conflict where goal achievement for the one goal deteriorates that of the other goal [Erl12].
2. The contrary antagonism of goals describes that the attainment of the two goals cannot be improved at the same time, though the fulfillment of the one goal can be improved without negatively affecting the fulfillment of the other goal [Erl12].
3. The subordination of goals is possible when attainment of some goals is basically easier to accomplish than others due to their lower implementable requirements [Erl12].
4. The compatibility of goals exists if two goals can be accomplished independently [Erl12].

Though Erlach [Erl12] suggests that in some scenarios improving individual goals does not affect another goal to the same degree. The objective of production optimization is to counterbalance this four goal dimensions in such an order so that business goals are achieved efficiently at specific production sites.

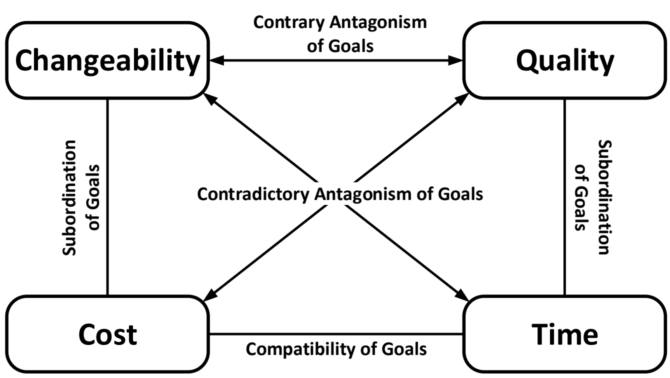

Figure 1.2.: Conflicting Goals of Production Processes (Adapted from [Erl12])

The economic aspect can be foreseen from a recent sector research work carried out by Deutsche Bank [Hen14] where Heng [Hen14] predicts that Germany has a major long-term opportunity to fortify itself as the leader in the global marketplace with its favorable market fundamentals. Industry 4.0 will help Germany to hold the grounds in manufacturing sector even against its fast-growing developing market competitors such as China and India. Thus Germany can remain an industrial heavyweight in the manufacturing sector being the undisputed economic powerhouse of the European Union (EU).

Heng [Hen14] appraises automation, optimization, and dynamic adaptability as the most important requirements in manufacturing sector to increase the efficiencies of the production processes. Since the dawn of sensors and networking technologies, vital information can be gathered beforehand to decide the most suitable and optimized process among the set of available processes. According to the suggestions of Sungur et al. [SBLW16], manually conducted assembly tasks may provide alternatives to the existing automation methods depending on the current demand, status of the machinery, and occupation of the machinery. Similarly Wieland et al. [WKNL07] recommend usage of modern world smart-systems to observe situations that will enable the application of well-adopted business process modeling and execution solutions in the context of manufacturing.

According to Zor et al. [ZGL10], production processes contain both manufacturing and all associated business processes to finish production on time, whereas manufacturing processes only focus upon processes that transform raw materials to final products. Wiendahl et al. [WEN$^+$07] have depicted production process as a macro-level process whereas manufacturing process as micro-level. Both Production and Manufacturing processes can be modeled using business process modeling languages [ZSL11], e.g., Business Process Execution Language (BPEL) [OAS07] or Business Process Model and Notation (BPMN) [OMG11]. After modeling, process models are deployed on compliant workflow engines for an automated execution. But standard BPEL or BPMN don't support adaptive and flexible execution of business processes in manufacturing industry. Manufacturers might gain substantial revenue and edge in market by adapting to structural changes on time [SBLW16, WKNL07].

1.1. Problem Statement

Production processes need to be updated and optimized at regular intervals to stay competitive in the market. With the emergence of new sensor technologies, *Internet of Things (IoT)* (Chapter 2.2), the production processes can be made smarter to leverage the next industrial revolution - commonly referred as *Industry 4.0* (Chapter 2.4).

Sungur et al. [SBLW16] presented a novel approach to support *Context-sensitive Adaptive Production Process* in their research work. They proposed to extend a production process by a sequence of predefined set of sub-processes, i.e., *Context-sensitive Execution Steps (CES)*. For each CES, context-relevant sub-processes are chosen and desired processes are elected, optimized, deployed and executed in runtime [SBLW16]. This approach dictates a way in which processes can possibly adapt themselves to the changing context. All details relevant to context-sensitive workflows have been discussed thoroughly later in Chapter 3.

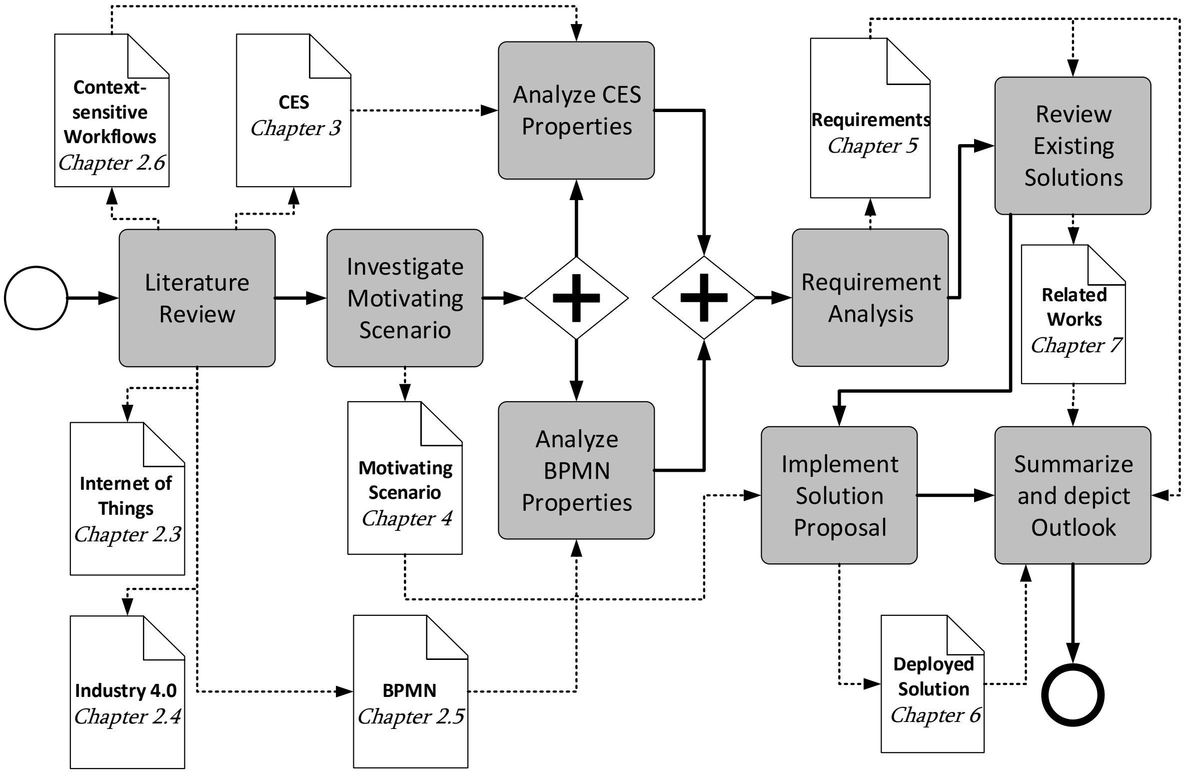

Figure 1.3.: Thesis Methodology and Structure

In this thesis work, we implement the CES flow in a process language agnostic way such that it can be seamlessly integrated with any BPMN engine. We might extend the BPMN keeping its semantics intact if and only if it's needed at all and add the necessary details to make manufacturing process models executable. To create this implementation, we have analyzed the properties that make CES unique and also scrutinized relevant BPMN properties that might be vital during implementation of the CES. These properties are later on used to derive our requirements from which we model our implementation. A summary of the thesis work can be found below in Figure 1.3 that is discussed in the section.

1.2. Methodology and Outline

The remaining master thesis document is structured in the following way as shown in the Figure 1.3.

- *Chapter 2 - Fundamentals:* The literature review of this thesis is carried out in three steps as suggested by Levy et al. [LE06]. We analyze literature related with Industry 4.0, Internet of Things (IoT), Business Process Model and Notation (BPMN), and Context-sensitive Workflows. Main focus was to understand the current development in Industry 4.0 and its implications in midst of the innovations in IoT technologies. Chapter 2.5 is focused upon analyzing properties of CES from the point of view of Business Process Management (BPM), and the efficacies of BPMN. In Chapter 2.6 we analyze literature related with context and context-sensitive workflows.

- *Chapter 3 - Context-sensitive Execution Step:* Properties and operational semantics of Context-sensitive Execution Step (CES) are discussed in this chapter. After subsequent description, formal algorithm of our intended approach has been discussed.
- *Chapter 4 - Motivating Scenario:* For the sake of analysis and for applying the conceptual workflow modeling construct, we have described a motivating scenario depicting a real-world manufacturing scenario which is a mix of both manual- and automated tasks.
- *Chapter 5 - Requirement Analysis:* In this chapter, we derive our requirements from the properties that we have found related. By defining our requirements, we conclude the task requirement analysis in the methodology model Figure 1.3. All the relevant properties and requirements for the CES have been described in this chapter.
- *Chapter 6 - Architecture and Implementation:* During the implementation of our conceptual construct, we use the BPMN extension methodology and we preserve the semantics of the existing BPMN properties. Architecture for the execution of modeled process is touched upon in this chapter.
- *Chapter 7 - Related Works:* In the next task, we select and analyze few already existing extensions of BPMN or any ongoing work in same direction. We propose our solution which will satisfy the requirements that we have previously defined to make sure that our approach proposed by Sungur et al. [SBLW16] can cater the best to the manufacturing sector.
- *Chapter 8 - Summary and Outlook:* In the last chapter, we give a summary and an outlook about our contribution to the manufacturing world.

The thesis document also contains an appendix for the further look-up:

- *Appendix A - List of Acronyms:* The list containing all the abbreviations or acronyms which are used in this document is added in this appendix.

1.3. Summary

In this chapter, we have introduced the need of adaptability in manufacturing industry by discussing the challenges the production and manufacturing companies going to face in next decades. Even though some sections discussed here are not relevant for our main goals, they are relevant for the better understanding of the reader. Finally we have highlighted briefly the thesis methodology that we are going to follow for our research work.

2. Fundamentals

The productivity growth was barely perspicuous for much of the human history and the living standards improved at a snail's pace. Then in the late eighteenth century, a disruptive innovation took place: the *Industrial Revolution* treated as *Industry 1.0*, in which the muscle power of all living beings was replaced by mechanical power which was led by introduction of steam engines and internal combustion engines to the mechanical production facilities. From the early part of the twentieth century, electrification and the division of labor led to the second industrial revolution which is referred as *Industry 2.0* now. The third industrial revolution referred as *Industry 3.0*, also known as the *Digital Revolution*, was set in around the late twentieth century, when Information and Communication Technology (ICT) developed further the automation of production processes. [EA12, HPO15, RLG^{+}15].

Innovations in IT world accelerated sharply the productivity and economic growth that led to sharp rise in demand for new skill-sets, i.e., ICT. At present, another workforce changeover is on the horizon as manufacturing industry is on the threshold of another technological advancement where digital virtual intelligence will augment physical machines. The conditions are full grown and early evidences manifest that this new wave of innovation is already upon us [EA12, LRS^{+}15].

In the next few sections, we discuss about how the next industrial revolution will evolve in the following decade and how it will affect the manufacturing industry and the world economy.

2.1. Cyber-Physical Systems (CPS)

Cyber-Physical Systems (CPS) are "integrations of computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa." [Lee08]. CPS are being used almost everywhere, e.g., medical equipments, driver safety systems for automobiles, industrial automation systems, etc. [Jaz14].

According to Jazdi [Jaz14], a typical CPS consists of a control unit and usually one or more μController(s). The μController is responsible for controlling the sensors and actuators to communicate with the physical world and further it processes the data gathered by the control unit as shown in Figure 2.1. A CPS may need to communicate with other CPS or a Cloud over a communication interface. Jazdi [Jaz14] goes on to define CPS as an embedded system capable of exchanging data over a network. Thus sometimes CPS connected to the Internet and IoT are regarded as same things.

Ultimately CPS requires three components as pointed out by Drath et al. [DH14] similar to the explanations of Jazdi [Jaz14]: "(i) the physical objects, (ii) data models of the mentioned physical objects in a network infrastructure, and (iii) services based on the available data."

[DH14]. Drath et al. [DH14] opine that CPS will ease out the interconnection, integration, testing and simulation components and products in industrial production. This will ascertain CPS as one of the prime technology drivers of the forthcoming industrial revolution [RLG+15].

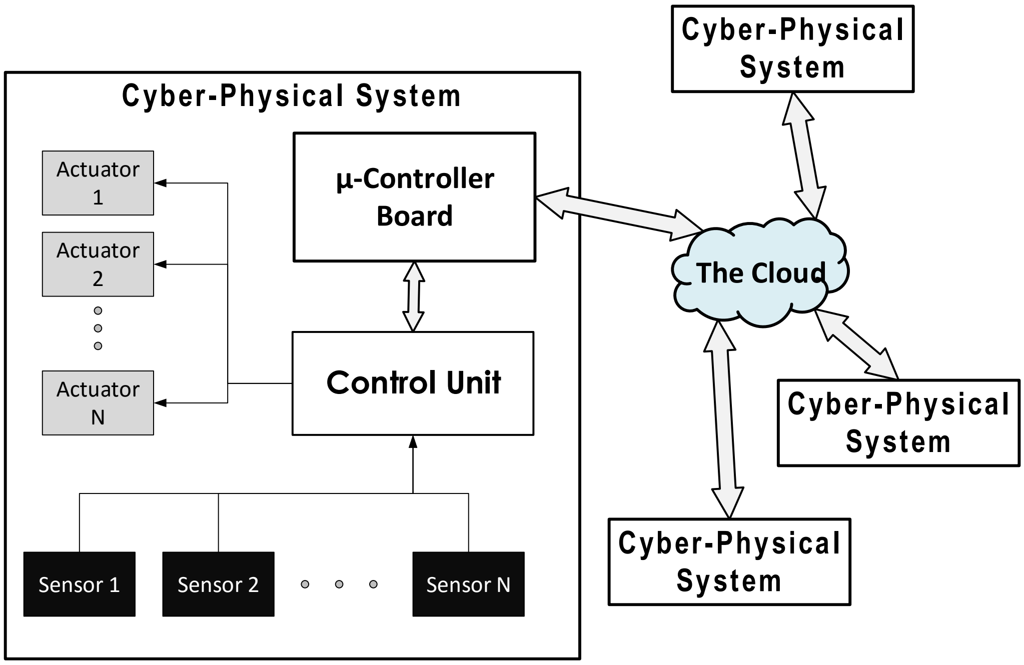

Figure 2.1.: Conceptual Architecture of a CPS (Adapted from [Jaz14])

2.2. Internet of Things (IoT)

The Internet revolution in the twenty-first century led to the interconnection between people at an exceptional scale and pace. The number of interconnected devices is expected to reach 24 billion devices by 2020 as per the estimates by [GBMP13]. The forthcoming industrial revolution is leveraging the creation of a smart environment by connecting objects with objects. The future is thriving upon a new era of ubiquity, i.e., Internet of Things (IoT) era in which human-thing communications and thing-thing communication themselves will be of prime importance [TW10].

Kevin Ashton [Ash09] had coined the term *"Internet of Things (IoT)"* in the context of supply chain management in 1999. Since then, the definition of IoT has covered wide range of applications, e.g., transport, health-care, utilities, etc. Gubbi et al. [GBMP13] predict that *Sensor–Actuator–Internet* framework will be the core technology behind IoT based smart environment. Gubbi et al. [GBMP13] further propose that only Cloud Computing can fulfill the requirements of the next generation IoT applications since it can provide reliable ubiquitous access, dynamic resource discovery and flexibility of choosing different service levels.

2.2.1. Definitions of IoT and Its Trends

IoT is a very broad vision and the research into the IoT is still in its infancy. Therefore, there are not any standard definitions for IoT. The following definitions were provided by different researchers.

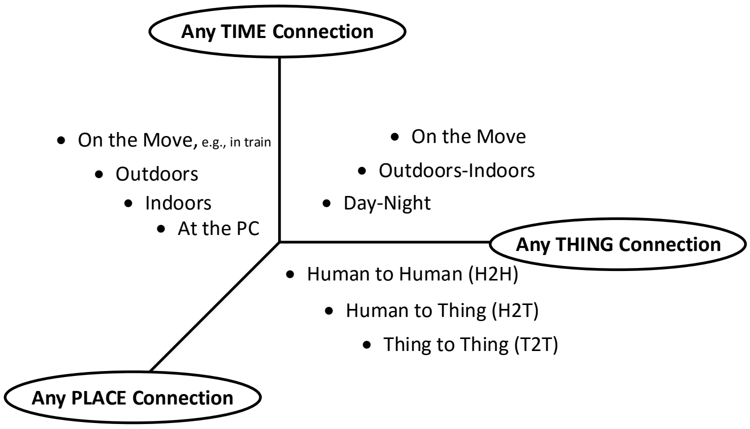

Figure 2.2.: IoT Dimensions (Adapted from [TW10])

Xia et al. [XYWV12] put forward a general IoT definition in their editorial - "*IoT* refers to the networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence. IoT will increase the ubiquity of the Internet by integrating every object for interaction via embedded systems, which leads to a highly distributed network of devices communicating with human beings as well as other devices." [XYWV12].

Gubbi et al. [GBMP13] explain IoT from the point of view of the Cloud applications - "*IoT* means interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a Common Operating Picture (COP) for enabling innovative applications, that is achieved by seamless ubiquitous sensing, data analytics and information representation with Cloud computing as the unifying framework." [GBMP13].

Tan and Wang [TW10] define IoT focusing mainly upon its functionality and identifies IoT as a new dimension that has been added to the world of ICT, i.e., from any *Time*, any *Place* connectivity for anyone to now connectivity for any *Thing* as shown in Figure 2.2 - "*IoT*s have identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environment, and user contexts." [TW10].

Similarly the Cluster of European Research Projects [SGFW10] explains Internet of Things in a more abstract manner - "In the *IoT*, *Things* are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real / physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention." [SGFW10].

We accept the definition provided by Xia et al. [XYWV12] for our research work, because we believe, this definition encapsulates the broader vision of IoT.

Gartner [RvdM15] defines a *Hype Cycle* as "a way to represent the emergence, adoption, maturity, and impact on applications of specific technologies." [RvdM15]. IoT has been identified as one of the emerging technologies in IT as noted in Gartner's IT Hype-cycle - 2015 already shown in Figure 2.3. As per its estimation IoT will take five to ten years for market adoption to reach the plateau of technology. In recent years, IoT has gained much attention from researchers, academia and industries from all over the globe [RvdM15, XYWV12].

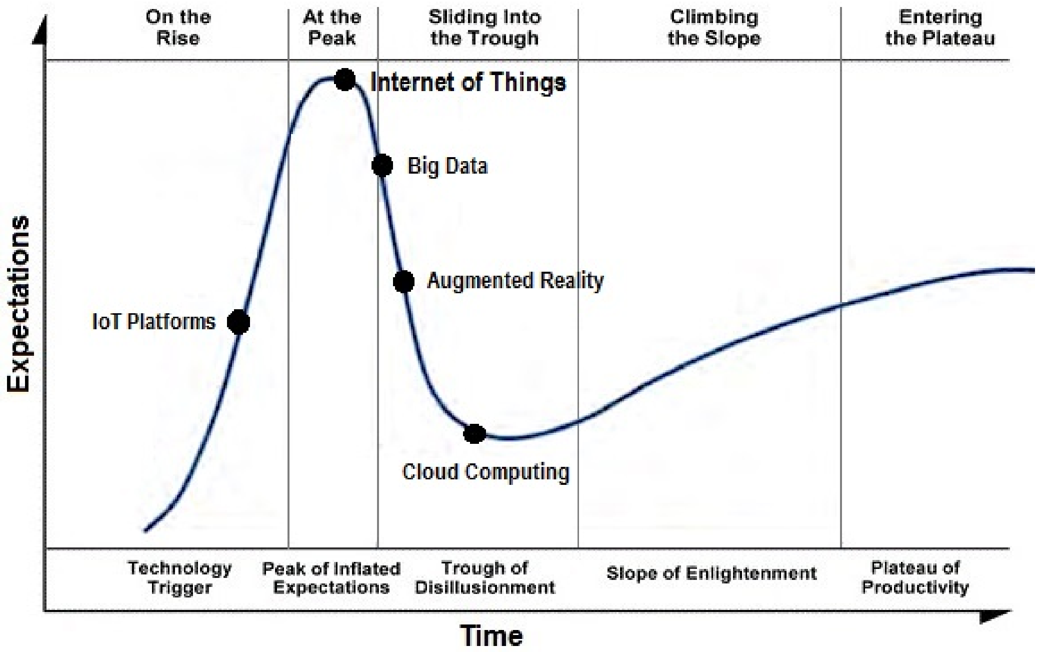

Figure 2.3.: IoT in Gartner Hype-cycle - 2015 (Adapted from [Bro11, RvdM15])

2.2.2. Elements of IoT

IoT is a technological revolution that represents the future of ICT [TW10]. Ubiquitous Computing (Ubicomp) is the method of enhancing computer usage such that it will be omnipresent in the physical environment, yet mostly invisible to the user [Wei93]. According to Gubbi et al. [GBMP13], three IoT components enable the scene behind seamless Ubicomp: (i) *Hardware* made up of sensors, actuators and embedded communication hardware, (ii) *Middleware* [ICG07] like on demand storage and computing tools for data analytics, and (iii) *Presentation* for user-friendly intuitive visualization and interpretation tools that can be widely accessed on different platforms. Things can be connected wired or wireless [GBMP13]. Base on the existing ICT infrastructure, there are many ways to connect a things, e.g., Radio Frequency Identification (RFID), Wireless Sensor Network (WSN), WiFi, 3G Universal Mobile Telecommunications System (UMTS), 4G Long-Term Evolution (LTE), etc. Here we discuss two few enabling technologies, i.e., RFID and WSN in brief for the sake of our research work.

- *Radio Frequency Identification (RFID)* is a non-contact technology that identifies objects attached with tags that help in the automatic identification of anything they are attached to [MWZ+07, TW10]. The passive RFID tags without own power source are pretty common in supply-chain, bank cards, inventory management and road toll tags. Active RFID readers with own power source are mostly used in port containers for monitoring cargo [GBMP13, MWZ+07].
- *Wireless Sensor Network (WSN)* is a dynamic, ad-hoc sensor network that comprises tiny, low-cost, and low-power sensor nodes communicating between themselves using only wireless technologies. Some of the application areas of WSN are health, military, and security, e.g., a node in a WSN might measure temperature values in a room while another node controls the air conditioning according to the sensed values and desired overall room temperature [SSOK13, ASSC02, GBMP13].

2.2.2.1. Addressing Schemes

According to Gubbi et al. [GBMP13], unique identification of *Things* is critical to uniquely identify and control billions of devices remotely through the Internet. Uniform Resource Identifier (URI) gives the most convenient approach to uniquely address each and every sensor nodes. Internet Protocol version 6 (IPv6) can work as an URI to access the resources uniquely and remotely since IPv4 addresses might not be sufficient [GBMP13].

2.2.2.2. Storage, Analytics and Visualization

According to Gubbi et al. [GBMP13], the data accumulated by IoT devices have to be stored, analyzed and visualized using Cloud Computing. Service providers are capable of using state-of-the-art techniques, e.g., genetic algorithms, neural networks, and other artificial intelligence (AI) techniques to make business decisions. Lastly, visualization enables business experts to convert raw data into business knowledge, which is most important in fast decision making [GBMP13].

2.2.3. IoT Architecture

According to Tan and Wang [TW10], IoT is a ubiquitous application technology which can't be designed and deployed over a five-layered TCP/IP architecture. In the IoT billions of nodes are connected which will create much larger traffic and need much more data storages. Gubbi et al. [GBMP13] view IoT from two perspectives, i.e., *Internet-centric* and *Thing-centric*. The *Internet-centric* architecture will revolve around internet services while data is contributed by *Things*. In the *Thing-centric* architecture, smart objects take the center stage [GBMP13]. Tan and Wang [TW10] propose of an Internet-centric approach in their research work.

A simpler conceptual framework shown in Figure 2.4 is inspired from architecture proposed by Tan and Wang [TW10] that integrates the ubiquitous sensor nodes and the applications is of our interest. It also satisfies the proposition laid out by Gubbi et al. [GBMP13] that realization of the full potential of Cloud Computing and Ubicomp is most viable by combining two approaches with a cloud at the center.

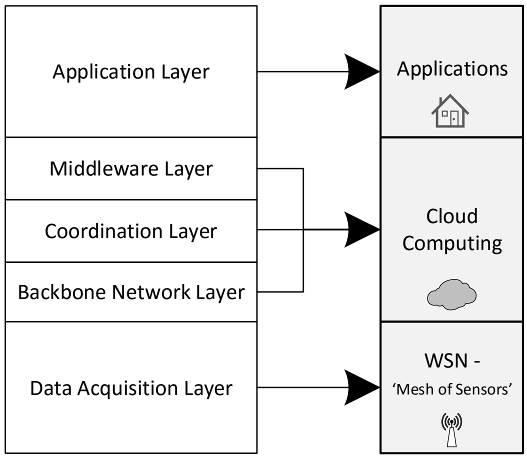

Figure 2.4.: Conceptual IoT Architecture Stack (Adapted from [TW10, GBMP13])

The *Backbone Network Layer* can be thought as the Internet of present day or its expansion. The *Coordination Layer* processes the data received out of different application systems in an unified structure so that the application system becomes inter-operable among the already existing systems and the newly deployed systems [TW10]. Middleware is a software layer that provides reusable solutions to application layer such that gathered data can be reused seamlessly [ICG07, TW10]. As per our evaluation of the model, the three layers in the middle (Middleware, Coordination and Backbone Network) of Tan and Wang [TW10] can be integrated and realized as a single layer of Cloud Computing as proposed by Gubbi et al. [GBMP13]. Gubbi et al. [GBMP13] further mention that sensor providers can offer their data using a storage cloud; analytic tool developers can provide their software tools; AI experts can provide their data mining tools and computer graphics designers can offer their visualization tools as Infrastructures, Platforms, or Software over a cloud.

2.3. Smart Factory

Production processes and the supply-chain in the manufacturing industry can be further improvised by CPS and IoT technologies. As suggested by Dais in a conversation with Löffler and Tschiesner [LT13], innovating truly new technologies and finding competent human resource for robust algorithm design can together leverage the rise of another industrial revolution. More and more industries will continue to have separate design and production processes where supply-chain integration will play a decisive role [LT13].

Lean Production principles discussed by Shah and Ward [SW07] are popular in industries that are intended for removal of waste out of production processes by continuous improvement and emphasis on value adding activities. If a plant implements lean manufacturing, CPS monitor physical processes and make decentralized decisions by communicating and cooperating with each other and human over the IoT. Internet of Services (IoS) provides services that are utilized by participants of the value chain [HPO15]. As a result, a factory becomes *"Smart Factory"* [KZ15, LT13].

Lucke et al. [LCW08] define Smart Factory as "a context-sensitive manufacturing environment that can handle turbulences in real-time production using decentralized information and communication structures for an optimum management of production processes." [LCW08].

The department of Innovative Factory Systems (IFS) at the German Research Center for Artificial Intelligence (DFKI) identified four enablers as shown in Figure 2.5 for the *Smart Factory* [KZ15]:

- *Smart Operator* could administer ongoing activities in ease equipped with smart sophisticated tools, e.g., CPS can auto-detect failures and trigger repair steps [KZ15].
- *Smart Product* could collect precise process data for the analysis during and after its production which is less labor-intensive [KZ15].
- *Smart Machine* like CPS could be integrated into error-prone production process to make it modular and flexible [KZ15].
- *Smart Planner* could optimize processes in real-time, e.g., CPS can optimize a production process by different business objectives, like time, cost or efficiency [KZ15].

As suggested by Derenbach in a conversation with Löffler and Tschiesner [LT13], it's impossible to separate the physical world from business processes; hence translation of physical world to an IT understandable format will require deep insights of mathematical, domain and market know-how. Interdependencies among the manufacturing components and manufacturing environment will be dominated by the usage of ICT [LT13]. In the context of forthcoming industrial revolution, the combination of industrial automation and Lean Production can be instrumental and Smart Factory is such a case in point [KZ15].

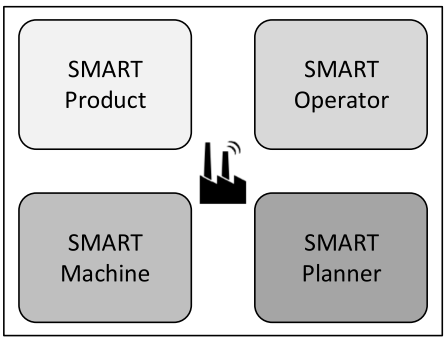

Figure 2.5.: Enablers of Smart Factory (Adapted from [KZ15])

According to Landherr and Constantinescu [LC12], planning and optimization of manufacturing processes with IoT enabled CPS is of very interdisciplinary nature because manufacturing experts have very limited interest in ICT and ICT experts don't have adequate knowledge of the manufacturing processes [LC12].

2.4. Industry 4.0

According to Hermann et al. [HPO15], the term *"Industry 4.0"* - the forthcoming industrial revolution, became publicly familiar in 2011 at Hanover Fair, where Kagermann et al. [KLW11] - a group of industrialists, politicians, and academicians had advertised *"Industrie 4.0"* as an approach to enhance the competitiveness of the German manufacturing industry. The German Federal Government supported their idea by announcing that Industry 4.0 will be an integral part of its "High-Tech Strategy 2020 for Germany[1]" initiative. A group termed as "Industrie 4.0 Working Group" was formed later and it developed and published the first draft of recommendations for its implementation in April 2013 [HPO15].

In North America, General Electric [EA12] has brought similar ideas under the name *Industrial Internet*. Though the technical background is very similar to Industry 4.0, but the application is broader than just industrial production. These different names have caused confusion rather than increasing transparency [DH14]. Hereafter, Industrie 4.0 or Industrial Internet or Integrated Industry will be referred interchangeably with Industry 4.0.

Hermann et al. [HPO15] have provided rationale for the fascination behind Industry 4.0. Firstly, Industry 4.0 is more theoretical [DH14] than empirical like the previous ones which provides possibilities to the industry and academia to shape the bright future ahead. Secondly, Industry 4.0 promises substantial increase in operational effectiveness along with revenue as well as the development of entirely new business process models and products.

2.4.1. Definitions of Industry 4.0

From the literature review of Hermann et al. [HPO15] and Kagermann et al. [KLW11], *Industry 4.0* is a collective term for contemporary automation, data exchange, and manufacturing technologies and concepts of value chain organization which draws together Cyber-Physical Systems (CPS), the Internet of Things (IoT), Smart Factories and the Internet of Services (IoS) together.

Rüßmann et al. [RLG$^+$15] explain it in a similar way as a new industrial revolution - "*Industry 4.0* is a new digital industrial technology that will connect sensors, machines, work-pieces, and IT systems along the value chain beyond the enterprise which in turn will interact with another using standard Internet-based protocols and adapt to changes." [RLG$^+$15].

We will accept the first definition for our research work as it is clear, unambiguous and more consistent than the latter. This definition by Herman et al. [HPO15] portrays Industry 4.0 as the facilitator of Smart Factory.

2.4.2. Components of Industry 4.0

According to Rüßmann et al. [RLG$^+$15], with Industry 4.0, modern ICT will transform relationship among suppliers, producers, and customers - as well as between machine and

[1] http://www.hightech-strategie.de/

human because production processes will no longer remain isolated from each other. Industry 4.0 will change isolated flows into an integrated, optimized, and automated production flow. Rüßmann et al. [RLG+15] and Hermann et al. [HPO15] have discussed many major factors that propels this next industrial revolution among which we have listed the most important ones in our opinion.

- *Smart Factory* is a system that assists people and machines in execution of their tasks. It is like a Calm-system that keeps on working in the background and it is aware of its environment, i.e., the system can consider information coming from physical and virtual world like the position of an object [HPO15].
- *CPS* are integrated network of computation, networking and physical processes that monitor and control the physical processes with input coming from physical world and output going back to physical world, e.g., autonomous automotive systems [HPO15].
- *IoT* allows sensors to interact with each other with limited intelligence. Decentralization of business analytics and decision making is made possible in production by IoT technology that act reactively to the changes in the environment [RLG+15, HPO15].
- *Internet of Services (IoS)* enables service vendors to offer their services over the internet which consists of participants, an infrastructure for services, business models, and the services themselves. Services are created using Enterprise Service-Oriented Architecture (SOA) and accessed as value-added services by consumers via various channels. In the context of Industry 4.0, IoS will offer production technologies and can either be used as manufacturing process or compensation process [OH12, HPO15].
- *Big Data and Analytics* based on large data sets can be used to optimize production quality, save energy, and improve Quality of Service (QoS). As per Gartner [BL12], Big Data is "high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization." [BL12]. In the context of Industry 4.0, it can facilitate the collection and comprehensive evaluation of data from production equipments as well as management information systems which can augment and make enterprise decision making more robust and consistent [RLG+15, HPO15].
- *Cloud Computing* is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources, e.g., networks, servers, storage, applications, and services that can be rapidly provisioned and released with minimal management effort or service provider interaction." as per definition of NIST [MG11]. In the context of Industry 4.0, it will make increased data sharing among the devices across various production sites happen such that reaction times to changes gets reduced. More the machines data and functionality will be deployed to the cloud, more data-driven services for production systems would be available [RLG+15, HPO15].
- *Augmented Reality* is a real or simulated environment in which a perceiver experiences telepresence [SBL+95]. Manufacturers can use augmented reality to provide real-time information to workers at production sites to improve decision making and in some cases workers can be trained using augmented reality technologies [RLG+15].

2.4.3. Enablers of Industry 4.0

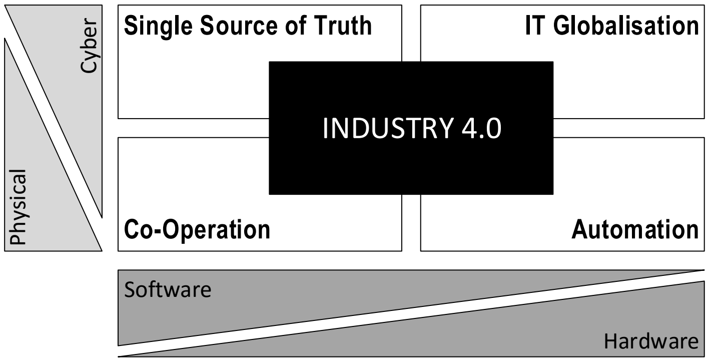

Figure 2.6.: Four Enablers of Industry 4.0 (Adapted from [SRHD15])

Industry 4.0 is a-priori industrial revolution [DH14] and therefore manufacturers have to take empirical measures by their own to introduce its enablers into their production premises to gain maximum profit [KLW11]. According to Schuh et al. [SRHD15], following four enables can be categorized with respect to two dimensions as shown in Figure 2.6. The first dimension describes whether a precondition is physical or virtual, whereas the second dimension describes whether the precondition is hardware or software components. Industry 4.0 can be viewed as a collaborated production phenomenon by the inter-working of machine, human and production system.

- *Single Source of Truth* dictates to embed all product life-cycle related data along the value chain within a single central database using cloud storage and accesses to make all changes to product and production visible and avoid ambiguity during production and simulations [SRHD15].
- *IT-Globalization* made computers achieve exponential growth in speed and cheap storage capacity. This will allow faster extensive simulations of different business processes as well as faster processing of huge amounts of data gathered by companies to get relevant information for improving the existing processes [SRHD15].
- *Automation* leads CPS to be decentralized processes which are able to adapt to dynamic requirements. Thus CPS become self-optimizing [SRHD15].
- *Cooperation* focuses upon connecting all technologies and business activities to empower decision making of decentralized CPS, e.g., efficient sharing of business process data within a network of business analysts [SRHD15, KLW11].

2.4.4. Mechanisms of Industry 4.0 for Next-gen Manufacturing

The work of Schuh et al. [SRHD15] and Schläpfer et al. [SKM15] can be synopsized to know how enablers of Industry 4.0 facilitate promise dramatic increase in the productivity of manufacturers significantly. We have briefly discussed five such mechanisms in this section.

- *Revolutionary Product Life-cycles through Horizontal Integration*: Existing rapid prototyping technologies facilitate companies to produce testable prototypes of the product so that business partner's and customer's feedback can be looped back into the system immediately across countries. Such an iterative model of process development is not as cost intensive as before and therefore leads to higher profit and shorter product life-cycle [SRHD15, SKM15].

- *Virtual Engineering of Complete Value Chains*: Sophisticated software tools are available to simulate the whole production processes to discover possible problems such as capacity problems within it. To get a valuable decision making capability based on simulations, it is necessary to execute a reasonable number of simulations [SRHD15].

- *Revolutionary Short Value Chains*: Taylorism proposes machines for accomplishing a specific task. In order to manufacture individualized and customer-oriented products, the integration of process steps within production systems can't be avoided which leads to a reversion of Taylorism causing price inflation. Yet there exists an optimal threshold for number of process steps in a production system which have to be incorporated to achieve minimal costs for the final product [SRHD15].

- *Vertical Networking*: Smart factories make manufacturing highly customer-specific and individualized using CPS to adapt to dynamic changes occurring in the market as well as production environment. IoT enabled networked resources and workers makes maintenance and logging of activities easier [SKM15].

- *Thorough Engineering and Exponential Technologies*: New innovations in technology can realize into self-optimizing systems which do exist theoretically. Such a production system would be able to reach a higher productivity level than expected due to cybernetic effects is predicted by Schuh et al. [SRHD15]. An example would be a productivity of 10,000 units whereas the estimated maximum before self-optimization was 8,000 units [SRHD15, SKM15].

2.4.5. Benefits of Industry 4.0 in Manufacturing

Industry 4.0 promises a range of benefits that spans across machines, industries, and societies which will influence the broader industrial economy. Some companies have already embraced it at this early stage realizing the huge benefits it pledges. Companies now need to overcome challenges that are hindering them from adopting this new wave of change. The initial strong impact is likely to be felt in the area of advanced manufacturing as predicted by General Electric [EA12]. Lorenz et al. [LRS+15] have analyzed what Industry 4.0 has in its store for the manufacturing industry by looking at the effects that this will have on Germany's manufacturing sector, which is one among the best in the world.

- Manufacturers will be able to boost up their competitiveness that will ease the expansion of their workforce as well as the increase in productivity [LRS+15, RLG+15].

- Manufacturers won't need to off-shore the factories to developing countries anymore as the labor cost advantages of those locations will be certainly nullified [LRS+15].

- Manufacturers will be able to create new jobs to cope up with the higher demand resulting from the new individualized products and services [RLG+15, EA12, LRS+15].
- Manufacturers can have robot-assisted production units that will reduce most number of jobs on the shop floor [LRS+15].
- The factory workforce will require less training for machine operation and production since production system will require very little manual intervention which will be highly automated and will be fitted with augmented reality measures [LRS+15].
- Manufacturers in developed countries can maintain their productivity despite the aging workforce by the use of automation to assist workers with manual tasks, e.g., a robot could lift a car's interior-finishing elements, such as a roof lining, into the chassis after manual alignment by an aging worker [LRS+15].
- Technology-assisted predictive maintenance would be enabled by Industry 4.0, e.g., a technician can identify defects and order spare parts just by remotely reviewing real-time sensor data on machine performance. Later, while making repairs, the technician can be assisted by augmented-reality technology in addition to the automatically documentation of whole process [LRS+15].
- *Capital costs* can be reduced through value-chain optimization, *energy costs* can be cut by smart control of production facilities, and *personnel costs* can be brought down with highly automated production processes promised by Industry 4.0, which will make manufacturing more cost-efficient than before [Hen14].

According to Heng [Hen14], Industry 4.0 implementation will be tailor-made for each company and cannot be supplied as "off the shelf" package. Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Fraunhofer, and companies such as Bosch Rexroth, Daimler, Siemens, Hewlett-Packard (HP), SAP, etc. have already shown the aforementioned benefits associated with Industry 4.0 developing various projects [Hen14].

According to Lorenz et al. [LRS+15], Industry 4.0 might become cause for job losses for some categories of work, such as assembly and production planning. But categories such as IT and analytics will gain as more people from these fields would be required to oversee the production. To realize the enormous promises of Industry 4.0, Governments need to strengthen their support to their economies and industries, academia need to close the IT skills gap, and early enabling companies need to retrain their workforce to remain updated and support this next wave of revolutions going forward by guiding late adopters [EA12].

2.5. Business Process Model and Notation (BPMN)

Business Process Model and Notation (BPMN) is is a standard visual notation for capturing business processes in a business process model [DDO08]. Business Process Management Initiative (BPMI.org) developed BPMN, which has been maintained by the Object Management Group (OMG) since both merged their respective Business Process Management (BPM) activities in June 2005. Version 2.0 of BPMN was released in January 2011 and the name was adapted to *Business Process Model and Notation* from *Business Process Modeling Notation* as execution semantics were also introduced alongside the notational and diagramming elements. Hereafter, BPMN 2.0 will be used interchangeably with BPMN [OMG11].

2.5.1. Motivation for Choosing BPMN

Kiper et al. [KAA97] observe that it's easier for non-programmers, e.g., business experts to model their business processes in a graphical (pictorial) way. Though Unified Modeling Language (UML) [OMG15] is an already established standard by and its behavioral diagrams such as activity diagrams are suitable for visual depiction, UML can be used for only software systems. Business experts would prefer an executable model for business process modeling which is not provided by UML. For such situations BPMN is suitable, because BPMN has its operational semantics [Sun13, OMG11].

Business Process Execution Language (BPEL) [OAS07] is an executable language for specifying business processes with web services standardized by Organization for the Advancement of Structured Information Standards (OASIS). Though BPEL is thought to have superior execution semantics, BPMN is widely adopted as the standard of business process modeling language to improve collaboration among stakeholders in the process as BPMN offers simplistic usual visual notation that is easy to comprehend than the BPEL in XML [CT12].

Because the business processes reflected in our research work will be executable business processes on process engines, we opt for BPMN to model such processes. BPMN has the characteristics that satisfy our requirements:

- BPMN has a well-known visual representation and is commonly used as it inherits and combines elements from earlier proposed notations for business process modeling such as XML Process Definition Language (XPDL) [WFM12] and the activity diagrams component of UML [DDO08].
- BPMN is executable.
- There are many open-source business process engines which support BPMN are available, e.g., Activiti[2], Stardust[3], Camunda BPM[4], etc.
- BPMN is highly intuitive especially at the level of domain analysis and high-level systems design from business expert's point of view that bridges the gap between business- and ICT experts as mentioned by Dijkman et al. [DDO08] in their work.
- BPMN can be extended in case it is needed [OMG11].
- Direct mapping from BPMN to BPEL is possible to some extent for which execution engines and formalizations exist [WvdAD+06, OAS07].

However, an extension to standard BPMN might be needed to address domain specific properties of CES construct which is going to be discussed in Chapter 2.6.

[2] http://www.activiti.org/
[3] https://www.eclipse.org/stardust/
[4] https://camunda.org/

2.5.2. Properties of BPM and BPMN

BPMN is the de-facto standard for representing in a very expressive graphical way and most business experts model their business processes mostly using BPMN [CT12]. The current BPMN models are treated to be more readable, flexible and expandable than before [Sun13]. In this section, we will give some properties of general BPM concepts and BPMN.

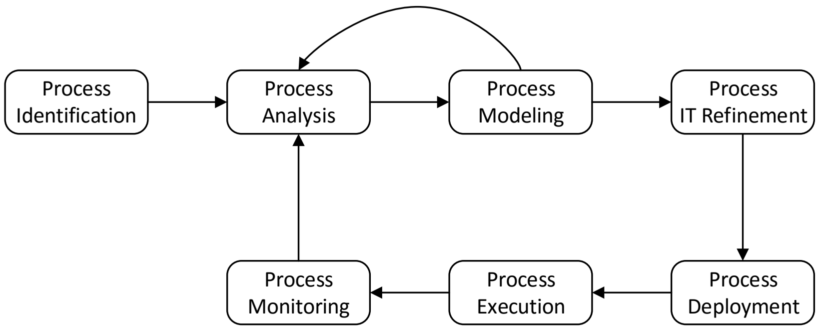

Figure 2.7.: BPM Life-cycle (Adapted from [DLRMR13, LR00])

2.5.2.1. Dimensions of Workflows (BP1)

Hollingsowrth [Hol95] defines *Workflow* as "the computerized facilitation or automation of a business process, in whole or part." [Hol95]. The BPMN users can orchestrate their processes as the new version 2.0 of OMG standard BPMN has its operational semantics and can be executed on its own process engine [Ley10]. The life-cycle of a business process is shown in Figure 2.7. In general, we need three dimensions to define business process items of any process [LR00]:

- *"What"*: What is the work item?, e.g., 'Check Sensor Status'.
- *"With"*: With what should this work item be accomplished?, e.g., 'a Web Service'.
- *"Who"*: Who will work upon this work item?, e.g., 'A Machine or Human'.

2.5.2.2. Hierarchies of BPMN Modeling (BP2)

Conceptually BPMN has 3 levels of modeling, i.e., *Descriptive Modeling (L1)*, *Analytical Modeling (L2)* and *Executable Modeling (L3)*. L1 documents the processes within a process model using basic BPMN shapes by describing the order of activities and the role or organization performing them. L2 documents the process in an unambiguous manner following BPMN semantics and validation rules. In L1, there can be errors in the model whereas in the L2 there is expected to be no errors. L3 targets the developers by adding language based execution details in the meta-model underlying the scheme. The generated BPMN serialized file is executed on a business process engine and it orchestrates defined set of activities [Sil11, OMG11].

2.5.2.3. Flexibility of BPMN Models (BP3)

Weske [Wes12] views business process models as the representations of internal business processes which can be executed by automated execution engines. Such business models need to be refined and optimized after each execution. BPMN provides ways to alter the business models, which will be effortless for a business expert [OMG11, Sun13].

2.5.2.4. Extensibility Mechanism of BPMN (BP4)

For certain domain specific applications, modeling elements of BPMN might not be sufficient. In these circumstances domain experts tend to extend BPMN meta-model to capture a better manifestation of their own application domains [Sun13, OMG11]. Generally, it can be achieved in two ways, i.e.,

- BPMN extensions are done by adding new elements and attributes to existing BPMN elements such that they do not contradict with already existing elements and attributes. This approach guarantees interchangeability of existing BPMN constructs. It's often done by defining desired properties in an external schema and referencing this schema from the internal schema [OMG11, AtAC15].
- There are some BPMN open-source vendors, e.g., Activiti, Stardust, etc. who have their own business process engines. Such BPMN vendors provide certain extensions points that they think will be useful in most of the business scenarios [AtAC15, Com15].

2.5.2.5. Cognitive Potency of BPMN (BP5)

The cognitive efficacy of any visual notation is most important for a better common understanding. Genon et al. [GHA11] analyzed BPMN using principles of the 'Physics of Notation' theory of Moody [Moo09] and how BPMN provides cognitive effectiveness is explained including its drawbacks. BPMN constructs are evaluated to have a one-to-one relation with its semantic description which satisfies the property of 'Semiotic Clarity', one of the prime factors that make a notation more intuitive [Moo09, GHA11].

2.6. Context-sensitive Processes

Business processes can be defined, managed and executed through a diversity of software-integrated systems by employing *Process Management Systems* that is driven by a business logic [Hol95]. Industrial production processes communicate with physical world with IoT nodes so that they can adapt to their environmental changes. Such dynamic processes will facilitate manufacturing to be more intelligent which can adapt to changing situations in runtime [WKNL07]. We will define few terms here which will be used throughout research work.

2.6.1. Definitions of Context and Context-sensitivity

Business processes should be able to handle information about the physical world, referred as *"Context"* [WKNL07]. The most general definitions that is provided by Abowd et al. [ADB+99] can be relevant for our further research work - "*Context* is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [ADB+99].

A process that considers context is called *"Context-sensitive"*. Wieland et al. [WKNL07] refer such processes as *"Context-aware"* workflows. Hereafter, Context-aware will be referred interchangeably with Context-sensitive. Abowd et al. [ADB+99] refer a system as *Context-sensitive* "if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task." [ADB+99].

Henricksen [Hen03] defines *"Context Attribute"* as an element of the context model that describes the context. A context attribute has an identifier, a type and a value, and optionally a collection of properties that can be treated as the building block of a meaningful context, e.g., <sensor-name, sensed-value> key-value pair can be a context attribute for a sensor node. [PZCG14]. The basis for context-sensitive workflows is context information such as the GPS position of workers and components, the state of all factory objects, etc. Context data is sensed via sensor modules mounted to the components [WKNL07].

2.6.2. Context Management Life-cycle

A context life-cycle shows how context move from phase to phase in a context-sensitive software system or business process engine. In this section we discuss about the movement of context in Context-sensitive systems. In simplest terms, Perera et al. [PZCG14] describe the context life-cycle in four phases as shown in Figure 2.8 [PZCG14].

- **Context Acquisition**: Context needs to be acquired from various sources that can be varied based on responsibility, frequency, context source, sensor type, and acquisition process. These five factors need to be considered when developing context-aware middleware solutions in the IoT paradigm [PZCG14].
  - *Based on Responsibility*: Context acquisition can be primarily realized using two methods, i.e., either the sensors can *push* data to the software component which is responsible for acquiring sensor data or the software component can *pull* the sensor data by making a request to the sensor over some medium [PZCG14].
  - *Based on Frequency*: Context can be gathered based on two different event types, i.e., *instant events* which don't span across certain amount of time, e.g., switching a light, and *interval events* that span a certain interval of time, e.g., sensing a wire after every 20 seconds [PZCG14].
  - *Based on Source*: Context can be acquired *directly from sensor* by communicating with the sensor hardware and related device drivers. IoT applications can acquire sensor data by *middleware solutions* such as Nexus [LCW09] where heterogeneous sensors

are deployed. Context can also be acquired from several other *context storages*, e.g., databases via different mechanisms such as web-service calls [PZCG14].

– *Based on Sensor Types*: There are different types of sensors that can be employed to acquire context. *Physical sensors* generate sensor data by themselves. *Virtual sensors* retrieve data from many sources and publish it as sensor data, e.g., Twitter tweets. *Logical (Software) sensors* combine both aforementioned sensors in order to produce more meaningful information, e.g., a web service dedicated to provide weather information [PZCG14].

– *Based on Acquistion Process*: There can be three ways to acquire context: *sense*, *derive* data by computing already sensed data, and *manually provided* data by predefined configurations [PZCG14].

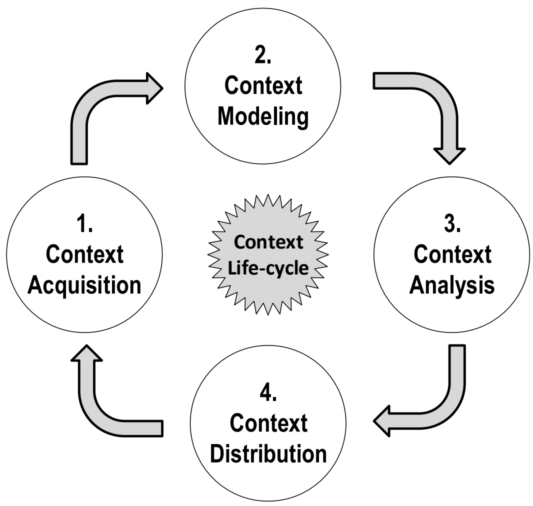

Figure 2.8.: Context Life-cycle (Adapted from [PZCG14])

- **Context Modeling**: The acquired data needs to be modeled and represented in terms of previously specified context, QoC attributes and the queries for context requests. After the validation of initial data, the context data can be pushed to an existing context repository. Perera et al. [PZCG14] have listed existing methods to model context data, e.g., *Key-Value Scheme*, *Markup Scheme (e.g., Extended Markup Language (XML))*, *Graphical* (e.g., databases), *Object-based*, *Logic-based*, *Ontology-based*, etc.

- **Context Analysis**: This phase is primarily meant for the cleaning and consolidation of multiple sensor data and inferring preliminary high level information from using lower-level context such that any remaining uncertainty and imperfection gets removed using different algorithms, e.g., *Decision Tree*, *hidden Markov Models*, *k-Nearest Neighbor*, *Neural Networks*, *Ontology-based*, *Fuzzy logic*, etc [PZCG14].

- **Context Distribution**: Finally, the context data needs to be delivered to the context consuming software agents. Context can be distributed either by a *query* (request) from consumer or consumers can *subscribe* for a specific sensor or to an event, that will make the process real-time [PZCG14].

2.6.3. Definitions of Context- Query, Event, Condition and Decision

Now we have reached the stage where we can define few more definitions proposed by Wieland et al. [WKNL07] which will be used across this research work.

- *Context Query* is a synchronous query designed in a specific query language, e.g., XPath, which supports object selection based on spatial predicates and filtering of the results, to access context data from context repository or database [WKNL07].
- *Context Event* is an asynchronous event triggered by change in context being monitored. The listening for this special environment state is done in parallel to the normal workflow [WKNL07].
- *Context Condition* is a predicate or operator in a specific language, e.g., XML Path Language (XPath) [W3C15] upon which context-decisions are evaluated [WKNL07].
- *Context Decision* is used to route process control flow based on context data using context-conditions [WKNL07].

2.7. Summary

In this chapter, we have provided background on Industry 4.0, IoT, BPMN and general Context-sensitive workflows. Even though some sections discussed here are not relevant for our main goals, they are relevant for future reference and we need to keep them in our minds during the process of the implementation.

- **IoT** is "a networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence. IoT will increase the ubiquity of the Internet by integrating every object for interaction via embedded systems, which leads to a highly distributed network of devices communicating with human beings as well as other devices" [XYWV12].
- **Industry 4.0** is "a collective term for contemporary automation, data exchange, and manufacturing technologies and concepts of value chain organization which draws together Cyber-Physical Systems (CPS), the Internet of Things (IoT), Smart Factories and the Internet of Services (IoS) together." [HPO15, KLW11].
- **BPMN** is the de-facto standard for representing in a very expressive graphical way and most business experts model their business processes mostly using BPMN [CT12].
- **Context** is "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [ADB+99].
- **Context-sensitive** system is "a system that uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task." [ADB+99].

3. Context-sensitive Execution Step

According to Perera et al. [PZCG14], IoT envisions a generation where thousands of sensor nodes would be deployed throughout manufacturing facilities connected over Internet and Context-sensitive applications that can decide which sensor devices to look for in an era of pervasive and ubiquitous computing. This enthralled many researchers and engineers across various disciplines to design prototypes, methods, and systems using Context-sensitive techniques. Likewise Sungur et al. [SBLW16] propose an approach by which manufacturing companies can remain competent in the ever changing global market by having an adaptive production process which is Context-sensitive too.

3.1. Definitions and Facets of Context-sensitive Execution Step

The approach introduced by Sungur at al. [SBLW16] presents the capability of capturing new information and adapting to it with an innovative modeling of manufacturing activities and manufacturing know-how. Before diving into any more conceptual details, we will define the basic terms that governs this approach.

Sungur et al. [SBLW16] propose a new process modeling construct named *"Context-sensitive Execution Step (CES)"*. Though CES constructs are visioned as sub-process structures of BPMN, we visualize it as BPMN task structures that are activated when the incoming flows are activated upon some conditions. Primarily a CES task contains its own input data and output variable to hold its generated output. A meta-model proposed by Sungur at al. [SBLW16] is adapted to our requirements can be seen in Figure 3.1.

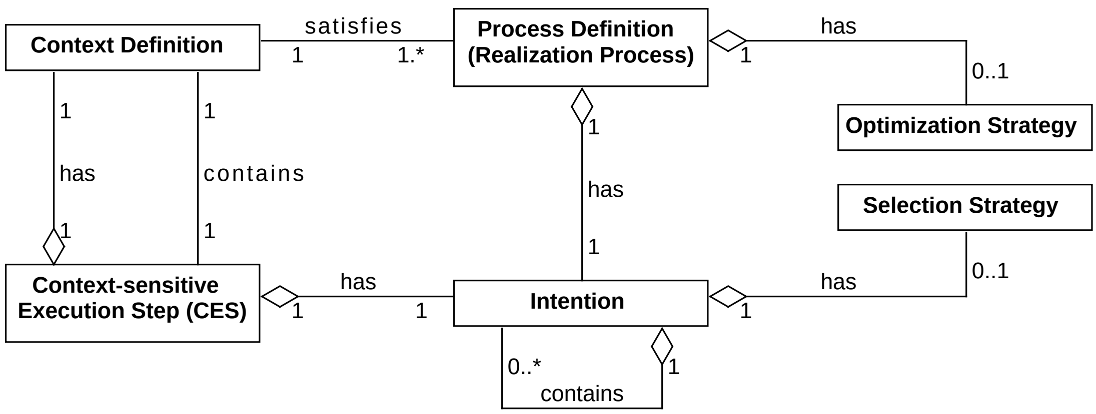

Figure 3.1.: Meta-model of a CES (Adapted from [SBLW16])

As depicted in Figure 3.1, a CES can be associated with following facets related to Context-sensitive modeling, i.e.,

- *Context Definition* specifies the required Contexts ($ConDef_i$) and related Context Conditions ($ConRule$) that will validate an underlying process definition in a certain scenario and thus makes CES adaptive, e.g., repairing a machine if its sensor senses so [SBLW16].
- *Intention* contains the main goal of the process, e.g., performing a specific task which is very specific in nature [SBLW16].
- *Process Definitions* ($ProDef_i$) are the main realization processes to achieve all defined intentions. A Context Definition can satisfy multiple process definitions by satisfying a set of correlated intentions (*Goals*), e.g., different process models to achieve same target [SBLW16].
- *Selection Strategy* ($SelStrat$) is used for choosing between multiple processes with same goals and Context Definitions that are generally contained by Main-intention, e.g., choosing a process based upon its probabilistic weight determined heuristically [SBLW16].
- *Optimization Strategy* ($OptStrat$) provides means of an automated optimization of the process to be run, e.g., an activity of optimizing resources before the main business process starts [SBLW16]. Providing dynamic flexibility or optimization during the execution of a BPMN model is out of the scope of our research work.

The set of all required contexts for the activation of a CES entity is denoted as *ReqCon*. Similarly all defined intentions of a CES entity that consists of its Main-intention and Sub-intentions are denoted by a set $ReqGoal$. *Sub-Intentions* are the refined intentions, i.e., sub-goals that will be associated with a Main-intention to model a certain business requirement, e.g., high automation, high maintenance activity, etc. [SBLW16]. Likewise, Process Definitions might contain *Complementary Realization Processes* (*ComplePro*) that are emphasized to run during the realization of main process definition for better business value and quality, e.g., a maintenance routine to take care of the machines [SBLW16].

3.2. Operational Semantics

The activation of a CES occurs as soon as its predecessor activity finishes its execution and the control gets transferred to CES task. As per the specifications of Sungur et al. [SBLW16], after the activation of a CES construct, the execution goes along with the flow shown in Figure 3.2. A CES can wait for its initial input data (if it exists) (S1.1), else it can directly reach the step where it gathers Contexts from some source (S1.2). CES must avoid any non-existing data usage as that might not be available in runtime.

If no context definition has been found, the process behaves as if all the underlying process alternatives satisfy the cause and execution proceeds to the Intention matching step (S3). Otherwise, the available Context Definitions are evaluated to the gathered Contexts (S2) and the context-satisfying set of processes (P_{con}) are sent for Intention matching (S3). After the matching process of Intentions of all the processes, a set of intentions-satisfying processes (P_{goal}) are generated, and the processes that are present in both P_{con} and P_{goal} are sent for redundancy check (S4). If there are more than one process definitions for reaching a Main-intention is available, elimination is carried out based upon some Selection Strategy ($SelStrat$). If $SelStrat$ is not available and Optimization Strategy ($OptStrat$) is available, execution goes to optimization phase (S5). Otherwise, selected process definition goes for execution and deployment in the final step without any intermediate optimization (S6).

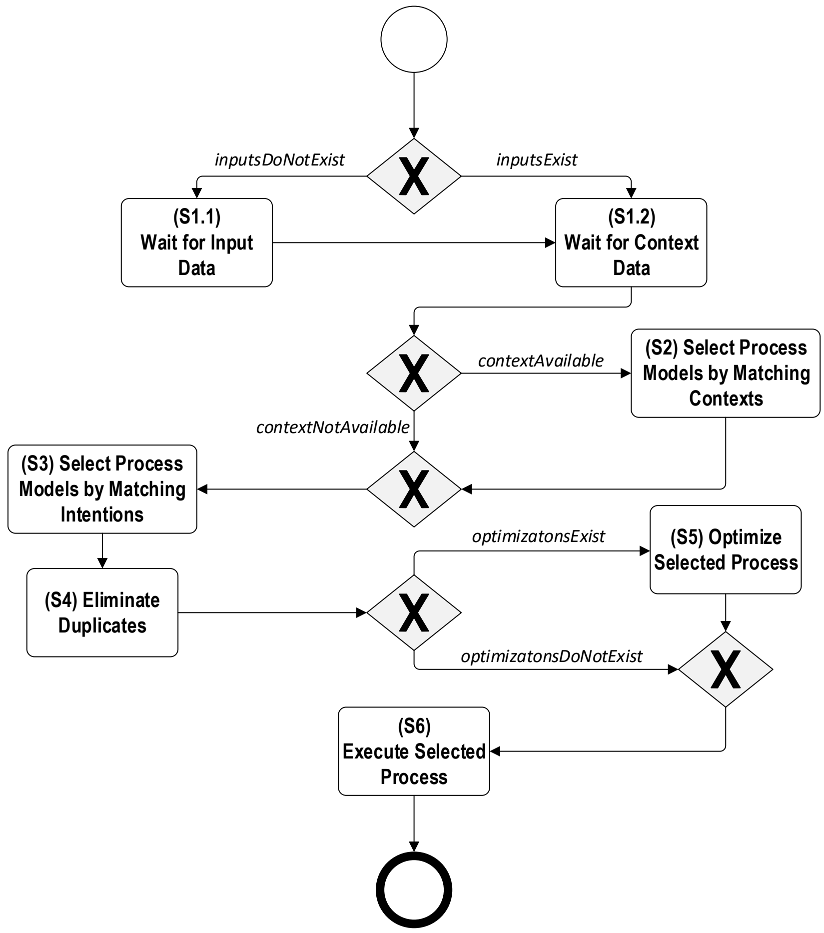

Figure 3.2.: Operational Semantics of a CES (Adapted from [SBLW16])

Sungur et al. [SBLW16] again propose the semantics for the successful execution criteria for a CES task. After the completion of a main realization process, complementary realization processes are enacted by the workflow engine. If the main realization process has succeeded and associated complementary process terminates unexpectedly, the execution is termed successful with a warning [SBLW16].

Prior to designing our algorithm, we will define *Process Definitions Repository (ProcRepo)* that contains process definition descriptor (*Id*), Context Conditions ($ConRule$), Complementary Realization Processes ($ComplePro$), Selection Strategy ($SelStrat$) and Optimization Strategy ($OptStrat$). We can formalize the whole operational semantics in the form of an algorithm for better understanding as shown in Algorithm 3.1 that has a linear worst-case complexity of $O(n)$. *Context Store (CS)* will store Contexts as key(k)-value(v) pairs, i.e., $CS = (\{C_i\})_{i \in [1,n]} = \{(k, v)\}$. Explicit input data set and output variable set will be denoted as IN and OUT respectively. OUT will contain final result to be returned to the callee or workflow engine.

Algorithm 3.1 Pseudocode for Operational Semantics of a CES

Input/Precondition:
Set $ProcRepo = (\{ProDef_i\})_{i \in [1,n]} = \{(Id, ConRule, Goals, ComplePro, OptStrat, SelStrat)\}$
$ProcRepo \neq \phi$, $ReqGoal \neq \phi$, $ReqCon \neq \phi$ and Set $IN = ProcessInput$.
Output/Postcondition: Set $OUT = ProcessResultVariable$

procedure EXECUTECES($ReqGoal$, $ReqCon$, IN, $ProcRepo$) ▷ These are the input data
    $OUT \leftarrow \phi$, $P_{goal} \leftarrow \phi$, $P_{con} \leftarrow \phi$, $initCon \leftarrow true$, $processDef \leftarrow \phi$
    for all $context \in ReqCon$ **do** ▷ Get value of each Required Context
        $value \leftarrow \text{GETCONTEXT}(context)$ ▷ This will query Middleware
        if $value \neq \phi$ **then**
            $CS_i.k \leftarrow context.name$ ▷ Store fetched Data in CS
            $CS_i.v \leftarrow value$
        else
            $initCon \leftarrow false$ ▷ No Context Available
    for all $condition \in ProDef_i.ConRule \wedge initCon = true$ **do** ▷ Context Matching
        $conVal \leftarrow \text{EVALUATE}(condition)$ ▷ Evaluate Conditions
        if $conVal = true$ **then** ▷ Store Descriptors
            $P_{con} \leftarrow P_{con} \cup ProDef_i.Id$
    for all $goal \in ReqGoal \wedge \exists ProDef_i \mid goal \in ProDef_i.Goals$ **do** ▷ Match Intention
        $P_{goal} \leftarrow P_{goal} \cup ProDef_i.Id$ ▷ Store Descriptors
    $P_{goal} \leftarrow P_{goal} \cap P_{con}$ ▷ Filter Mutually-exclusive Processes
    for all $id \in P_{goal} \wedge \exists ProDef_i \mid id = ProDef_i.Id$ **do** ▷ Select Process
        if $ProDef_i.SelStrat \neq \phi$ **then**
            $processDef \leftarrow \text{STRATEGYSELECT}(P_{goal}, ProcRepo)$ ▷ Select based on Strategy
        else
            $processDef \leftarrow \text{RANDOMSELECT}(P_{goal})$ ▷ Select randomly
    if $processDef.OptStrat \neq \phi$ **then** ▷ Check Optimization Strategy Existence
        $\text{OPTIMIZE}(processDef)$ ▷ Execute Optimization Process
    $OUT \leftarrow \text{RUN}(processDef, IN)$ ▷ Execute Realization Process
    $\text{RUN}(processDef.ComplePro)$ ▷ Execute Complementary Realization Process

3.3. Context-sensitive Production Processes

As per Sungur et al. [SBLW16], if we integrate CES with a standard production process, the latter will becomes *Context-sensitive Adaptive Production Process*. One of the important activity of BPM life-cycle is process identification. After the identification and analysis of production processes the variable parts of any production process can be defined as CES along with its input data, main-intention, etc. Realization processes are added to a central repository so that CES can look for these processes in it and deploy one of the filtered processes [SBLW16].

As we have chosen BPMN as our implementation and modeling standard, we need to describe a CES construct graphically using standard BPMN notation not violating the OMG standard. Moody [Moo09] suggests the importance of an icons that quickens cognizance and improve comprehensibility of diagrams to both naive and novice users. Unlike most Software Engineering notations that are visually one-dimensional and less appealing, we wanted to realize more than one of the eight available visual communication channels defined by Moody [Moo09]. Furthermore, we need an icon that will be easy to draw by hand while drawing a BPMN task shown in Figure 3.3 over non-digital medium. Hence, we came up with the icon shown in Figure 3.3 which is a *Data Object* with *"C"* written inside that signifies reception of Contexts from IoT devices. Our icon can be categorized as *Hybrid Symbol* by Moody [Moo09] where the text inside the object expands the meaning and such an amalgamation of textual and graphical representations makes the CES icon more appealing.

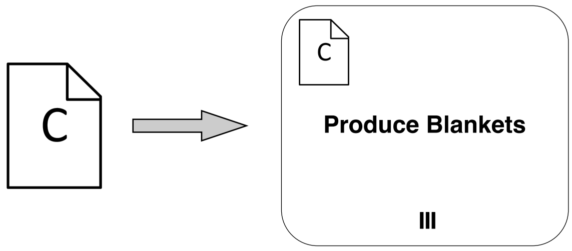

Figure 3.3.: Icon for a CES Construct and a BPMN Task with a CES Construct

A very rudimentary abstract architecture to realize the concept of CES have been proposed by Sungur et al. [SBLW16] that will be adapted by us again in the Chapter 6 while realizing the concepts.

3.4. Summary

In this chapter, we have provided introduced the concepts of CES along with its various components to the reader. The operational semantics of a CES is also discussed briefly in the form of a diagram and formal algorithm. Later we defined the title of our research work *Context-based Production Process* and designed the fancy pictorial representation of a BPMN task with a CES construct.

- **CES** are similar to BPMN task structures that are activated when the incoming flows are activated upon some conditions. Primarily a CES task contains an Intention, Context Definition, Process Definitions, Optimization Strategy (optional) and Selection Strategy (optional). Additionally it can contain its own explicit input data and output variable to hold its generated output [SBLW16].
- **Context Definition** specifies the required Contexts and related Context Conditions that will validate an underlying process definition in a certain scenario and thus makes CES adaptive [SBLW16].

4. Motivating Scenario

To reinforce our concept, we set the stage for our motivating scenario that is based on the case studies introduced by Erlach [Erl12]. The scenario describes a production process that produces electric blankets for the European market. These blankets under observations are primarily meant for Southern European countries where houses with heating are not common [Erl12]. Customers may order blanket of different variations, e.g., for single-bed, for doubled bed, foot warmers, etc.

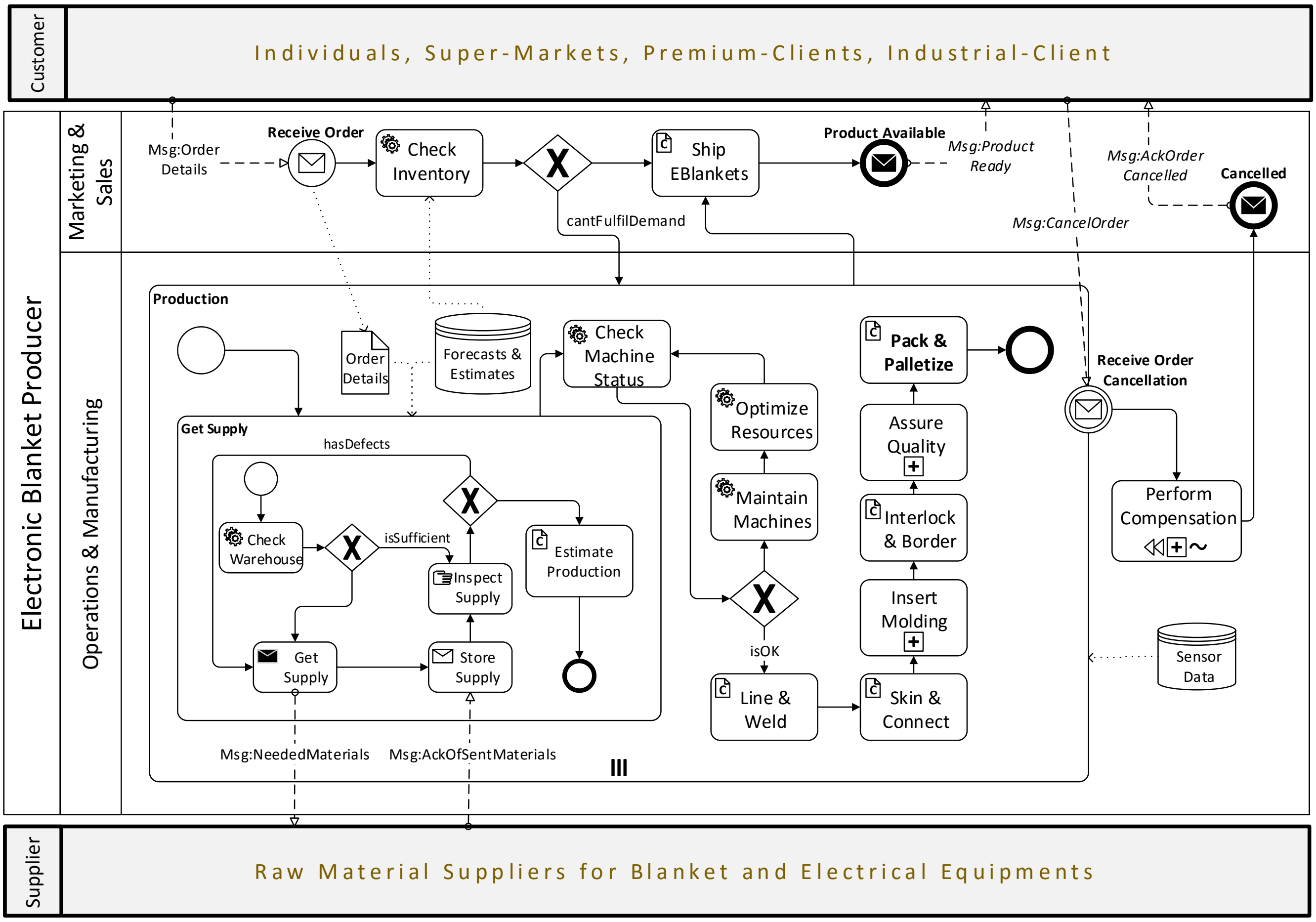

Figure 4.1.: Overall Blanket Production Process in BPMN with CES Tasks

The production process starts with the receipt of the order from the customer and inventory status is inquired so that a manufacturing decision can be taken. Each blanket may have a RFID tag attached so that the exact location and count of the blankets can be tracked remotely. If number of units order can be served from inventory directly, blankets are shipped directly via the *Shipping* task, otherwise manufacturing process begins with the activity of getting required raw-materials from the supplier considering other factors such as warehouse status, order amount and future order forecast. Subsequently the production process shown in Figure 4.1 revolves around six distinct subprocesses, namely, *Lining and Welding*, *Skinning and Connecting*, *Insert Molding*, *Interlocking and Bordering*, *Quality Assurance*, and *Packaging*.

Among all these subprocesses / tasks, only few of them can be modeled using CES task as those processes are dependent on the production environment, e.g., Lining and Welding, Skinning and Connecting, Interlocking and Bordering, Packaging, Shipping, etc. In our research work, we will simulate a CES task that consists of both manual and automated activities such that the paradigm of CES can be validated for both manual and automated tasks. *Packaging* task is well-suited for our purpose and in the following discussion we will analyze it in detail.

4.1. Process Variants

Packaging of blankets involves three activities: (i) Packing blankets manually or using machine by placing a blankets with an instruction leaflet into a cardboard box, (ii) Sealing cardboard boxes and wrapping them as per safety rules, and (iii) Sorting blankets on pallets and transferring them to the staging area for shipping. The whole *Packaging and Palletizing* process of blankets is shown in Figure 4.2.

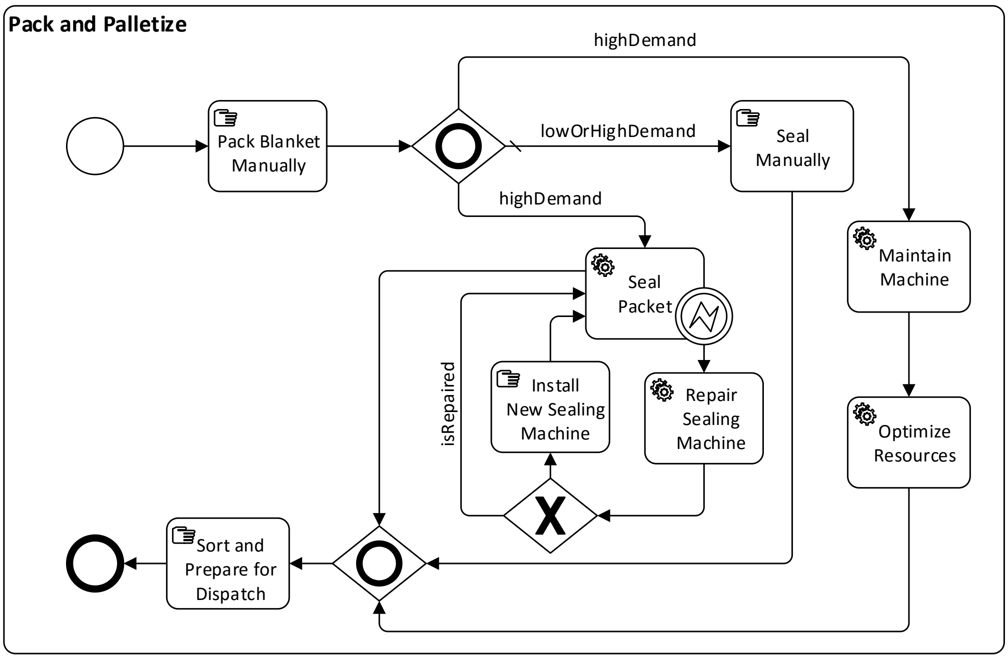

Figure 4.2.: Packing and Palletization of Blankets in Standard BPMN

The packaging process also contains two optional activities as shown in Figure 4.2: (i) Maintenance activity that only runs if the sealing is carried out by machine, and (ii) Resource optimization activity that is a supplementary task carried out (if required) to ensure the best operational environment for the production systems. The depicted scenario where IoT elements like RFID takes center stage of production is inspired from the research of Scholz-Reiter et al. [SRTÖS09] where logistics processes of jeans manufacturers are governed by Smart-labels, i.e., RFID.

4.1.1. Main Realization Process Models

Our motivating scenario starts when the product passes the quality check and packaging process begins. The process ends when the blankets are palletized for shipping. Depending on the context, i.e., sensor statuses, availability of resources and business forecasts, four different process variants of packing and palletizing can be assumed out of the consolidated process shown in Figure 4.2. The process of remodeling process variants out of a composite process model can't be thought as *Process Fragmentation* introduced by Schumm et al. [SKK$^+$11] because process fragments are extracted from a process analogous to extraction of sub-graph out of a graph where the objective of fragment and parent process differ in most cases.

- **Manual Variant** (P1) shown in Figure 4.3 assumes the unit of blankets ordered by customers, or forecasted to be produced to be too low, i.e., less than 1000 units per day. It also assumes abundance of workforce in the factory premises during the production, i.e., more than 10 workers. For keeping the whole workforce occupied in production, every task is carried out manually without the usage of any automation or machine. Manual variant has an assumed frequency of execution of 20% in our scenario.

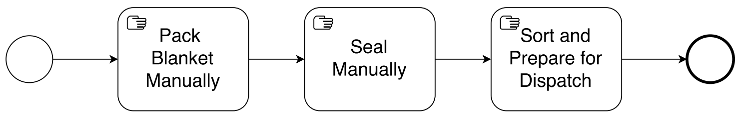

Figure 4.3.: Manual Packing and Palletization of Blankets in BPMN

- **Semi-manual Variant** (P2) shown in Figure 4.4 assumes the unit of blankets to be sealed to be very high, i.e., greater than 1000 units per day. It also assumes that all the machines

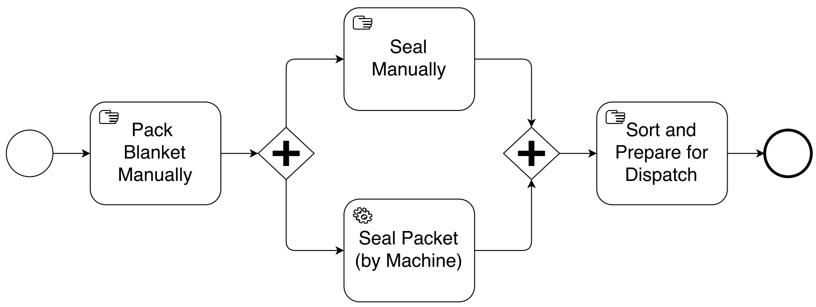

Figure 4.4.: Semi-manual Packing and Palletization of Blankets in BPMN

required for the sealing task are functioning properly, i.e., attached IoT nodes have not signaled any malfunction. Thus, sealing can be carried out both manually (by 10 workers) and automatically using the machines, which lead to high throughput. This Semi-manual variant has an assumed frequency of execution of 74% in our scenario.

- **Semi-manual Adaptive Variant with Repairing Activity** (P3) shown in Figure 4.5 is the extension of aforementioned process P2, where the malfunctioned sealing machine is repaired automatically by adaptive monitoring and maintenance nodes, and semi-automatic sealing is restored back to the normalcy. Such an adaptive Semi-manual variant has an assumed frequency of execution of 5% in our scenario.

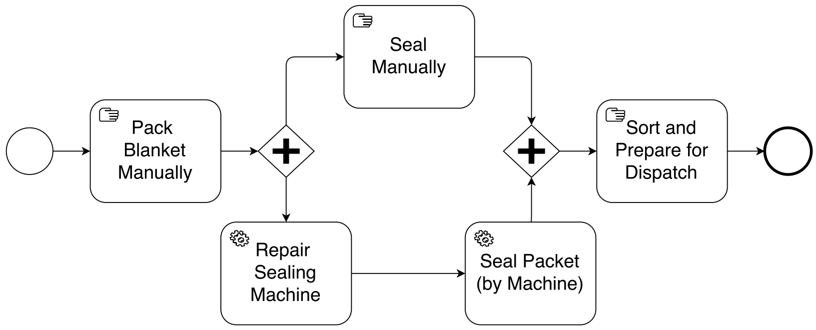

Figure 4.5.: Packing and Palletization of Blankets in BPMN with Repair Activity

- **Semi-manual Adaptive Variant with Re-installation Activity** (P4) shown in Figure 4.6 is the extension of aforementioned process P3, where the malfunctioned sealing machine can't be repaired anymore due to some irreversible damage occurred during the production. Therefore a new machine has to be installed and commissioned as soon as possible. Such a variant has an assumed frequency of execution of 1% in our scenario.

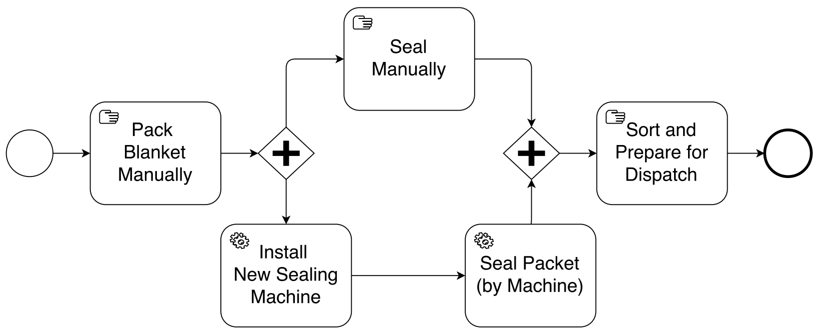

Figure 4.6.: Packing and Palletization of Blankets in BPMN with Re-installation Activity

4.1.2. Complementary Realization Process Model

Any other realization processes defined in the execution environment that are not meant to be the main business process, rather a supporting process of the main business process is referred as complementary process. In our scenario, the maintenance of machines shown in Figure 4.7 can be thought as a naive complementary process (P5) which is executed whenever we choose one of the semi-manual process variants, i.e., P2, P3, or P4 as our main business process so that the machines can be taken care of after the execution has been finished.

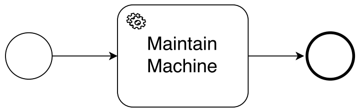

Figure 4.7.: Complementary Realization Process of Maintenance in BPMN

4.1.3. Optimizing Process Model

To remain competitive in market, processes need to be improved regularly to adapt new challenges, e.g., a naive optimization task (P6) shown in Figure 4.8 would improve resource utilization and deallocate extra workers from one production facility to another that requires more workers. Optimization is not the core interest of our research work as we are more interested upon realizing a context-sensitive execution rather than dynamic optimization of process models. Generally, processes can be remodeled by discovering the pitfalls in the earlier model in the final step of BPM life-cycle, i.e., process monitoring and auditing.

Figure 4.8.: A Naive Optimization of Resources in BPMN

4.2. Process Modeling Considerations and Assumptions

Manufacturing processes thrive upon the execution context collected from the physical world for the successful completion of the process. In this section, we have briefly explained the context gathering techniques used for our motivating scenario along with relevant assumptions and constraints considered throughout the context modeling.

4.2.1. Context Acquisition

In our motivating scenario, we will model our business process depending upon four simple contexts mentioned below that are gathered from different sources.

- **Amount of Blankets** to be produced or forecasted to be produced will be referred as `unitsOrdered` henceforth. If `unitsOrdered` is small, it's advisable to deploy more manual workforce which will be economic for the company rather than deploying high-end machines with lots of surveillance and maintenance measures. If `unitsOrdered` is large, deploying both machines and manual workforce seems logical to gain edge in market. `unitsOrdered` is a primary context received from the customer.

- **Availability of Workers** will be referred as `availableWorkers` henceforth. The GPS locations of the factory workers can be tracked so that workers of certain capability can be allotted to a specific task. Each machine unit equipped with a RFID chip inside can be aware of its location and find the nearest worker who can oversee it. For simplicity we will use 3 worker profiles, namely, (a) Supervisors, (b) Machine Operators and (c) Manual Workers. `availableWorkers` is a secondary context calculated from the primary contexts like GPS locations.

- **Sensor Statuses**: Various sensors in conjugation with the machines will form a WSN of their own. In our scenario, packet sealing machines augmented with passive infrared sensors can track the movement of objects, i.e., unsealed cardboard boxes. Shock detectors embedded in the sealing machine ensure that the the blanket and its packet don't get ruptured due to any malfunction in sealing machine. `infraredSensorStatus` and `shockDetectorStatus` will be referred as `sensorStatus` for simplicity.

In the end, we define the corresponding Context-conditions for each context such that it can be validated to realize a satisfying process variant.

Variant	Context Condition
P1	`unitsOrdered` $\leq$ `1000 AND availableWorkers` $\geq$ `10`
P2	`sensorStatus = "Okay" AND (unitsOrdered > 1000 OR availableWorkers < 10)`
P3	`sensorStatus` $\neq$ `"Okay" AND (unitsOrdered > 1000 OR availableWorkers < 10)`
P4	`sensorStatus` $\neq$ `"Okay" AND (unitsOrdered > 1000 OR availableWorkers < 10)`

Table 4.1.: Process Variants with Context Conditions

4.2.2. Types of Intentions (Goals)

Manufacturing industry now captures cross-functional interdependencies and proposes objectives that will improve the business both in quality and cost. In our packing and palletizing of blankets scenario, we try to find such business objectives that will impact the modeling process, e.g., *High Throughput*, *High Human-Resource (HR) Utilization*, *High Automation*, *Low Maintenance*, *Complement*, *Optimize*, etc. Table 4.2 lists the intentions we have assumed for the variants of our process model.

Variant	Main-Intention	Sub-Intentions
P1	Pack & Palletize	High HR Utilization
P2	Pack & Palletize	High Automation, High HR Utilization, High Throughput
P3	Pack & Palletize	High Automation, Low Maintenance, High Throughput
P4	Pack & Palletize	High Automation, Low Maintenance, High Throughput
P5	Complement	-
P6	Optimize	-

Table 4.2.: Process Variants with their Intentions

4.3. Summary

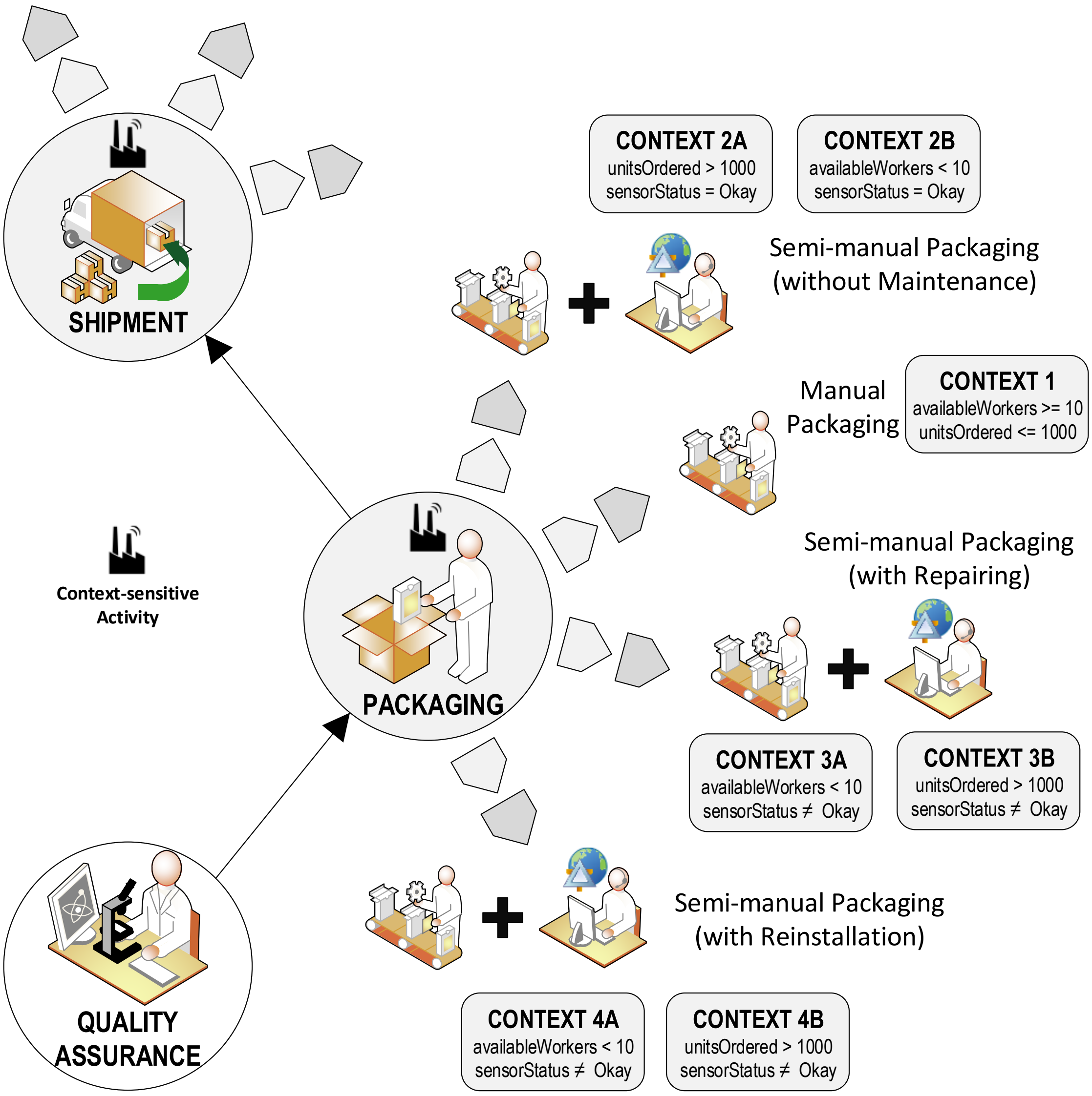

Figure 4.9.: Comprehensive Visual Graph of Motivating Scenario

Finally, our motivating scenario can be recapitulated using a visual graph shown in Figure 4.9. The assumed Context-rules for each context-set are shown in shaded boxes. This chapter will allow us to rethink and analyze the requirements of our CES task. As shown in Figure 4.9, our scenario of packaging of blankets can have different process variants with different objectives. Quality assurance task is a non context-sensitive task that is carried out before the control gets transferred to our context-sensitive packaging task. Furthermore, we explained all our assumptions, i.e., context gathering techniques, context modeling, etc. that will be required during the implementation of our concepts later.

5. Requirements for Context-sensitive Execution Step

In this chapter, we analyze the properties of CES tasks relevant for business process modeling which in conjugation with the BPMN properties will guide our requirements for the same. There are some properties that need to be conformed during BPMN modeling keeping critical properties of CES tasks intact.

5.1. Drivers and Properties of Production Processes

To appraise CES task properties, we need to consider the driving forces behind production processes that are important during BPMN modeling. Such drivers become relevant property if and only if it's critical for the realization of CES task.

5.1.1. Cutting Edge Support for Industry 4.0 and IoT (DP1)

IoT introduced technologies that revolutionized the manufacturing industry by pushing it to the brink of Industry 4.0. Manufacturing industry is now facing a global trend to manufacture customer-oriented products in an efficient and intelligent way such that any change to the production environment is adapted by CPS operating inside a smart factory [SKM15, WKNL07, DH14].

5.1.2. Volatile due to Changes in Business Goals (DP2)

Production processes are tightly coupled with the business world. We have already discussed the turbulences the manufacturing world can face in Chapter 1 [Wes06]. Such disturbances lead to the change of business objectives that are tailor-made for specific scenarios, e.g., if a factory wants to reduce energy consumption, it should utilize its HR extensively for doing tasks manually than automated machines, since the objective of business shifts here from *High Automation* to *High HR Utilization* which is pretty straightforward to apprehend.

5.1.3. Susceptible to Instability in Execution Environment (DP3)

Process context changes due to changes occurring in production environment. Existing IoT elements are capable of monitoring the execution context in real-time which assists processes to become context aware. As real-world production systems are prone to more and more runtime instabilities, processes must be robust enough to adapt such changes [WKNL07], e.g., if a machine fails, manual alternative must be looked for until machine is repaired or replaced.

5.1.4. Optimal Process Execution (DP4)

Production processes focus primarily upon error-free execution, customizable product portfolio and high profitability. The optimization of each alternative path is significant for the overall efficiency of the process, e.g., if chances are high enough that a certain sub-process is needed again and again, it need not be de-provisioned [SBLW16, VHGSH+15].

5.1.5. Parallel Execution of Business Logic (DP5)

Parallel execution of the same manufacturing process is a common thing in standard business processes, e.g., multiple machines sealing packets at the same time to gain higher productivity. Concurrent manufacturing processes must remain independent of each-other and shared resources must be provisioned in a consistent and fail-safe manner.

5.1.6. Event-based or Interval-based Process Execution (DP6)

The difference in timings of business operations might result in different contexts and different behavior which makes this property relevant.

5.1.7. Data Propagation in Runtime (DP7)

Manufacturers need to study sales patterns, competitor performance, etc. to be able to make well-judged, efficient decisions. Such data can come from different sources, i.e., audit-data, serial data received from the previous activity, processing data generated while executing context. Such data should be stored persistently as context-sensitive systems take decision upon some well-analyzed information.

5.2. Requirements of CES Task

In this section, we will derive our requirements from the relevant properties discussed in the previous section. We will also consider the properties of BPMN discussed in Chapter 2.5.2 to preserve semantics of BPMN standard.

5.2.1. IoT Incorporated Production Processes (R1)

As an outcome of property DP1, process definitions inside the process repository should contain context informations, such as, required context names, context validation rules, etc. This requirement is an abstract requirement to make business process engines context-aware.

5.2.2. Goal-driven Production Process (R2)

As a consequence of property DP2, a CES task must look for satisfying business objectives, i.e., intentions of a process definitions so that the processes can be aligned with high-level business goals of the company.

5.2.3. Context-sensitive Production Process (R3)

As a consequence of property DP3, a CES must be able to sense the environment context whenever required and execute accordingly. It should be able to perform simultaneous execution of CES tasks too for context-driven adaptability [SBLW16].

5.2.4. Optimizable Production Process (R4)

As a consequence of property DP4, a CES task should be able to incorporate process optimization strategies to provide an edge to the company's business process management.

5.2.5. Event-driven or Periodic Context Acquisition (R5)

Event-driven operations are unlike periodic operations because they are triggered whereas periodic operations are executed in certain time intervals. Though CES task can provide both event-driven and periodic operations, we prefer to have event-driven CES task derived from the property DP6, since event-driven task consumes less energy as it requires lesser communication efforts to collect contexts from the IoT nodes.

5.2.6. Distribution of Data Across Production Processes (R6)

In CES tasks, data are produced which can be used by forthcoming tasks and subprocesses to make decisions in business process seamlessly. Similarly, CES task should be able to consume the contextual data received from IoT nodes (from the property DP1) along with the data received from the previously executed tasks and subprocesses. As a consequence of property DP7, CES task needs to preserve the semantics of the task construct defined in OMG standard of BPMN.

5.2.7. Prioritization of Goals (R7)

In a business model, there may be situation specific conflicting business goals, e.g., high automation and lower energy consumption simultaneously. To accommodate property DP2 during BPMN modeling, we should have a mechanism to prioritize business goals such that the corresponding operations get aligned with the performance goals defined, e.g., in a scenario, where energy costs are higher than normal, process model with the lower energy consumption should get the priority instead of highly automated one.

5.2.8. Resilience to Minor Changes (R8)

CES task should be flexible as standard BPMN tasks. Changes in the deployment of a particular process model such as switching to different process engine, optimizing process or data paths inside a process model, etc. should not affect the model itself. Thus, BPMN process model with CES task preserves the property BP3 of standard BPMN.

5.2.9. Multiple Instantiation Capability (R9)

Similar to standard business processes, multiple instances of the same CES task can exist concurrently to support parallel execution discussed as a property DP5. A visual representation of parallel CES task also needed to be defined to satisfy the property BP5. Multiple instantiation can increase the productivity and competitiveness of the company sharply.

5.2.10. Near Real-time Processing (R10)

Near real-time implies an event with no consequential delays, i.e., a tiny scenario-specific time delay is introduced by CES task between the event occurrence and use of processed data. Though the definitions of "Near real-time" and "Real-time" are not clear and distinct, "Near real-time" is suited in our case as our delay in processing can vary from scenario to scenario that is practical and can't be avoided [NCS96].

5.3. Summary

In this section, we investigated the properties of production processes and created our requirements using the relevant properties. A table of relationships between relevant properties and requirements is shown in Table 6.1. In this table, we have shown which property has been a basis for which requirement.

Requirement Name	Mapped Properties
IoT Incorporated Production Processes (R1)	DP1
Goal-driven Production Process (R2)	DP2
Context-sensitive Production Process (R3)	DP3, BP4
Optimizable Production Process (R4)	DP4, BP1
Event-Driven or Periodic Context Acquisition (R5)	DP6, BP5
Distribution of Data Across Production Processes (R6)	DP1, DP7
Prioritization of Goals (R7)	DP2
Resilience to Minor Changes (R8)	BP3
Multiple Instantiation Capability (R9)	DP5
Near Real-time Processing (R10)	DP1, DP3

Table 5.1.: Mapping of Requirements to Properties of Production Processes and BPMN

6. Case Study: Realization of Context-sensitive Execution Step using BPMN

To satisfy the requirements presented in Chapter 5.2, we propose extensions to standard BPMN service task, i.e., a CES Task is an extension of Service Task of standard BPMN and a Process Definition entity that extends standard Process of standard BPMN. These solutions are built upon the work by Sungur et al. [SBLW16]. In the following sections, we will define the aforementioned extensions in detail. In this chapter, we will give details about the archetypal implementation of the motivating scenario using the extensions that we proposed. In the following sections, we will also describe the architecture of our proposed solution and how we placed our extensions inside a BPMN engine along with the brief details of our implementation.

6.1. Context Modeling

All context data related to our motivating scenario can be modeled and represented as a non-relational database schema as shown in Figure 6.1. Such a schema can be validated later through Context-Conditions. Database storage makes the context retrieval moderately easier and large volume of context can be stored persistently this way by the Middleware. We have assumed `sensorStatus` can have 3 statuses as shown in Figure 6.1, i.e., `Stopped`, `Malfunctioned`, or `Okay`.

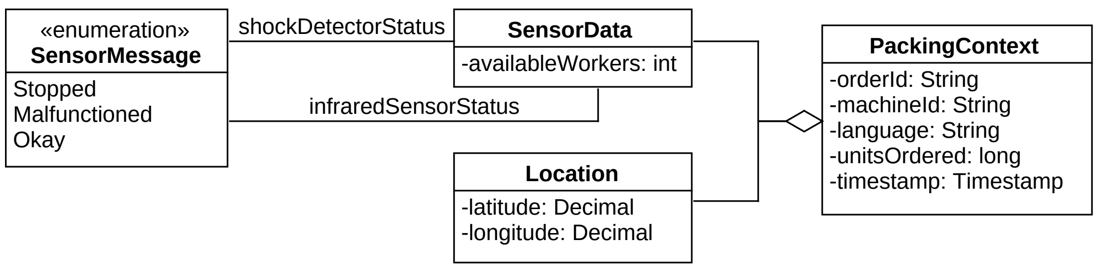

Figure 6.1.: A Schema of Context Data for a Non-relational Database

Cloud computing is making the non-relational model more desirable these days as these can scale quickly for better performance. IoT and Middleware developers do not have to settle for the relational model when data requires a different way of storage and retrieval for performance issues [PPS11].

6.2. Implementation Proposal

A CES task corresponds to a task executed in a BPMN process engine. A class diagram of a CES task is depicted in Figure 6.2. CES Task extends service based task of BPMN to make itself backward compatible with standard web services. The instances of Process Definition are realized by the CES Task by the matching target intentions and initial context specified in the process definition. Each element of this class diagram is discussed in detail in the following section.

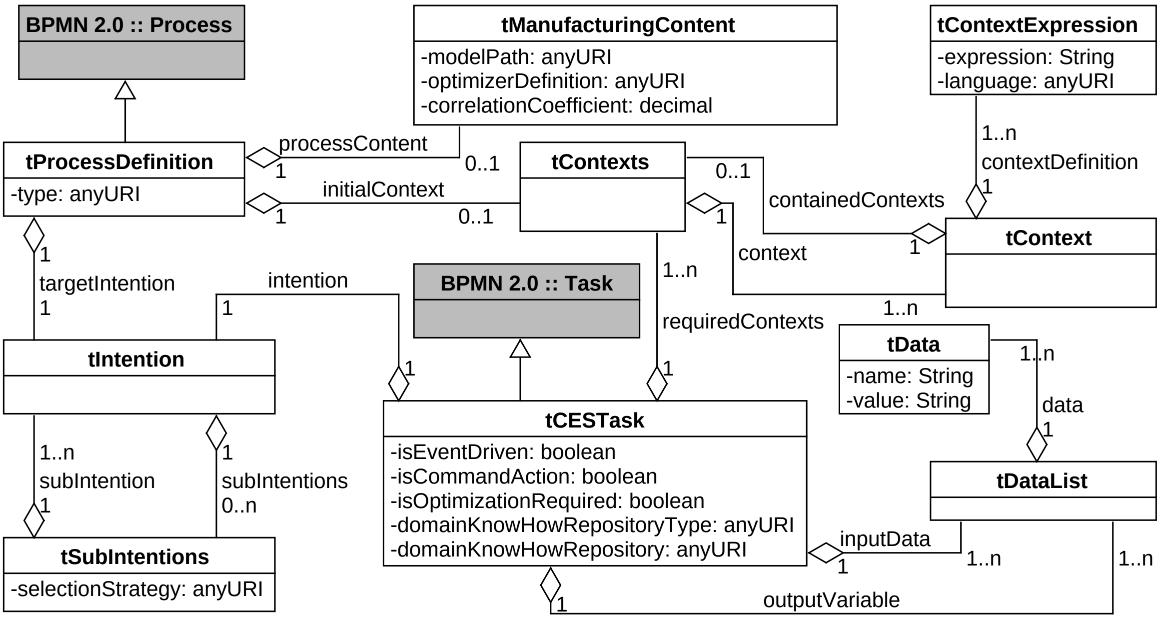

Figure 6.2.: Class Diagram of Proposed CES Task Extension

6.2.1. CES Task

As we just mentioned, `tCESTask` extends *Service Task* of BPMN to make itself backward compatible with standard web services. The instance based on `tIntention` is used to describe business goals or objectives dynamically which satisfies the requirement R2. The instance `tContexts` stands for the set of required contexts that needs to be gathered in runtime for an intelligent and efficient decision making, thus satisfies requirement R3. The `tDataList` instances are used to define the inputs and outputs of the `tCESTask` that satisfies the requirement R6.

By introducing the 5-tuple, i.e., `isEventDriven`, `isCommandAction`, `isOptimizationRequired`, `domainKnowHowRepositoryType`, `domainKnowHowRepository`, we meet requirements R4, R5 and R9. The `isEventDriven` and `isCommandAction` attributes can make `tCESTask` either periodic or event-driven and usage of this satisfies our requirement R5. To satisfy requirement

R4, `domainKnowHowRepository` stores the location of process definitions whereas `domainKnowHowRepository` checks the type of storage used, i.e., XML, Database based, etc. Listing 6.1 shows the XML Schema Definition (XSD) of a CES Task. Generally speaking, CES Task also satisfies the requirement of R1, R8 and R10 as it makes the BPMN service task pluggable to IoT based applications with near real-time processing capability.

Listing 6.1: XSD Definition of CES Task Definition

```
<complexType name="tBaseType" abstract="true">
	<sequence>
		<element name="Documentation" type="string"></element>
	</sequence>
	<attribute name="name" type="string"></attribute>
	<attribute name="targetNamespace" type="anyURI"></attribute>
</complexType>

<complexType name="tCESTask">
	<complexContent>
	<extension base="tBaseType">
		<sequence>
			<element name="Intention" type="tIntention"></element>
			<element name="RequiredContexts" type="tContexts"></element>
			<element name="InputData" type="tDataList" maxOccurs="1"
			    minOccurs="0"></element>
			<element name="OutputVariable" type="tDataList" maxOccurs="1"
			    minOccurs="0"></element>
			<element name="OptimizationRequired" type="boolean" maxOccurs="1"
			    minOccurs="0"></element>
			<element name="DomainKnowHowRepositoryType" type="anyURI"></element>
			<element name="DomainKnowHowRepository" type="anyURI"></element>
		</sequence>
		<attribute name="isEventDriven" type="boolean"></attribute>
		<attribute name="isCommandAction" type="boolean"></attribute>
	</extension>
	</complexContent>
</complexType>
```

6.2.2. Intention

The `tIntention` type is used to bind a CES Task to business goals of the company. The corresponding CES task is expected to match the goals of process definitions during the execution. To define the type of definition language used to define `tIntention`, we have used `tDefinition` instance. An intention element can contain itself as `tSubIntentions`. Listing 6.2 shows the XSD definition of Intention. This element satisfies the requirement R2. `tSubIntentions` contains `selectionStrategy` that can be used during the execution to choose among multiple processes satisfying same business objectives. Therefore, it also satisfies the requirement R7. `tIntention` element is generally defined during the modeling of a CES Task.

Listing 6.2: XSD Definition of Intention

```
<complexType name="tIntention">
        <complexContent>
        <extension base="tBaseType">
                <sequence>
                        <element name="Definition" type="tDefinition" maxOccurs="unbounded"
                            minOccurs="1"></element>
                        <element name="SubIntentions" type="tSubIntentions"
                            maxOccurs="unbounded" minOccurs="0"></element>
                </sequence>
        </extension>
        </complexContent>
</complexType>

<complexType name="tDefinition">
        <sequence>
                <element name="DefinitionContent" type="tContent"></element>
        </sequence>
        <attribute name="definitionLanguage" type="anyURI"></attribute>
</complexType>

<complexType name="tSubIntentions">
        <sequence>
                <element name="selectionStrategy" type="anyURI"
                minOccurs="1" maxOccurs="1"></element>
                <element name="SubIntention" type="tIntention" maxOccurs="unbounded"
                    minOccurs="1"></element>
        </sequence>
</complexType>
```

6.2.3. Contexts

The `tContexts` type is used to bind a CES Task to context data received from the IoT present in the execution environment. The corresponding CES task is expected to search for the specified contexts and fetch the present status of these context from the Middleware or directly from the WSN. Listing 6.3 shows the XSD definition of Contexts. This element satisfies the requirement R1 and R3 mostly. `tContext` contains an instance of `tContextExpression` to evaluate the context data based on some criterion and take business decision dynamically during the runtime.

Listing 6.3: XSD Definition of Context

```
<complexType name="tContext">
        <complexContent>
        <extension base="tBaseType">
                <sequence>
                        <element name="ContextDefinition" type="tContextExpression"
                            maxOccurs="unbounded" minOccurs="1">
                        </element>
                        <element name="ContainedContexts" type="tContexts" maxOccurs="1"
                            minOccurs="0"></element>
```

```
            </sequence>
        </extension>
        </complexContent>
</complexType>

<complexType name="tContexts">
        <sequence>
                <element name="Context" type="tContext" maxOccurs="unbounded"
                minOccurs="1"></element>
        </sequence>
</complexType>
```

6.2.4. Input and Output Data

The `tDataList` type is used to bind input and output data to the CES Task. The corresponding CES task is expected to store the final output in `OutputVariable` element. Listing 6.4 shows the XSD definition of Data. This element satisfies the requirement R6 straightforward. Key-Value scheme defined in `tData` makes it analogous to the data format being used in BPEL processes which makes it unifiable with other process modeling languages seamlessly.

Listing 6.4: XSD Definition of Input/Output Data

```
<complexType name="tData">
        <attribute name="name" type="string"></attribute>
        <attribute name="value" type="string"></attribute>
</complexType>

<complexType name="tDataList">
        <sequence>
                <element name="data" type="tData" maxOccurs="unbounded"
                    minOccurs="1"></element>
        </sequence>
</complexType>
```

6.2.5. Process Definition

`tProcessDefinition` extends process structure of BPMN to make itself backward compatible with standard processes. The instances based on `tIntention` are used to describe business goals or objectives that a business process confirms to.

Listing 6.5: XSD Definition of Process Definition

```
<complexType name="tProcessDefinition">
        <complexContent>
                <extension base="tBaseType">
                <sequence>
                        <element name="ProcessContent" type="tManufacturingContent"
                            maxOccurs="1" minOccurs="0"></element>
```

```
                    <element name="TargetIntention" type="tIntention"></element>
                    <element name="Definition" type="tDefinition" maxOccurs="unbounded"
                        minOccurs="1"></element>
                    <element name="InitialContexts" type="tContexts" maxOccurs="1"
                        minOccurs="0">
                    </element>
                </sequence>
                    <attribute name="processType" type="anyURI"></attribute>
                    <attribute name="id" type="ID"></attribute>
                </extension>
        </complexContent>
</complexType>
```

Each process can contain initial contexts defined as `tContexts` which generally contains the `tContextExpression` to look for the most eligible process model in runtime. Listing 6.5 shows the XSD definition of Process Definition. `tProcessDefinition` is augmented with instances of `tManufacturingContent` to specify the model repository paths and process specific attributes during the modeling time.

6.2.6. Manufacturing Content and Context Expression

`tManufacturingContent` is a 3-tuple, i.e., `modelPath`, `optimizerDefinition`, `correlation-Coefficient`. `modelPath` and `optimizerDefinition` persists the location of process model and it's optimizing process model respectively. `correlationCoefficient` can be used to assign priority to any process model so that in case of any ambiguity during the execution, higher priory process gets dispatched, which satisfies the requirement R4 and R7.

The `tContextExpression` type is used to specify context validation rules in a certain query language, i.e., XPath, so that the CES Task will be able to find the most eligible process model at a certain scenario. Listing 6.6 contains the XSD definition of Context Expression and Manufacturing Content just discussed previously.

Listing 6.6: XSD Definition of Manufacturing Content and Context Expression

```
<complexType name="tManufacturingContent">
        <sequence>
                <element name="ModelPath" type="anyURI" maxOccurs="1"
                minOccurs="0"></element>
                <element name="OptimizerDefinition" type="anyURI" maxOccurs="1"
                minOccurs="0"></element>
                <element name="CorrelationCoefficient" type="decimal" maxOccurs="1"
                minOccurs="0"></element>
        </sequence>
</complexType>

<complexType name="tContextExpression">
<sequence>
        <element name="Expression" type="string" maxOccurs="1" minOccurs="0"></element>
</sequence><attribute name="language" type="anyURI"></attribute>
</complexType>
```

6.3. Architecture of Realization

The architecture of Context-sensitive execution of a Goal-driven CES task is somewhat inspired by the work of Hirmer et al. [HWS+15] in SitOPT project. The conceptualized architecture of ours can be found in Figure 6.3. Our architecture consists of three distinct

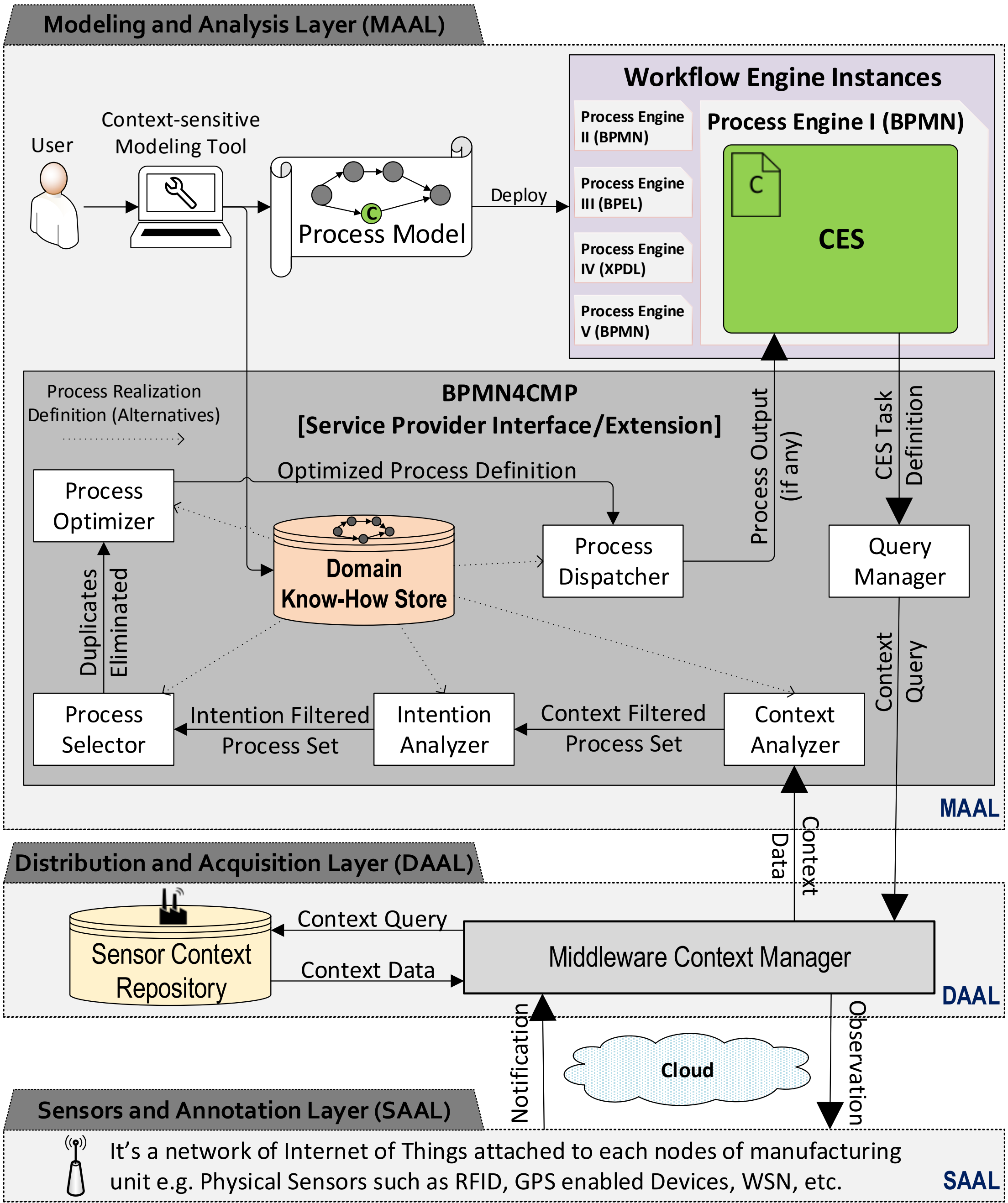

Figure 6.3.: Conceptual Architecture of CES Task Execution

layers, namely, *Sensors and Annotation Layer (SAAL)*, *Distribution and Acquisiton Layer (DAAL)*, and *Modeling and Analysis Layer (MAAL)*, which are detailed in the following subsections. Our work revolves around the development of components in MAAL. A deployment model of the CES Task can be found in Figure 6.4. The process repository contains process models, i.e., BPMN , BPEL, or XPDL files and corresponding model meta-data. The location of the repository defined in the `tCESTask` used during the run time. The extension - BPMN for Context-sensitive Manufacturing Process (BPMN4CMP), as we know it, contains different components of the MAAL.

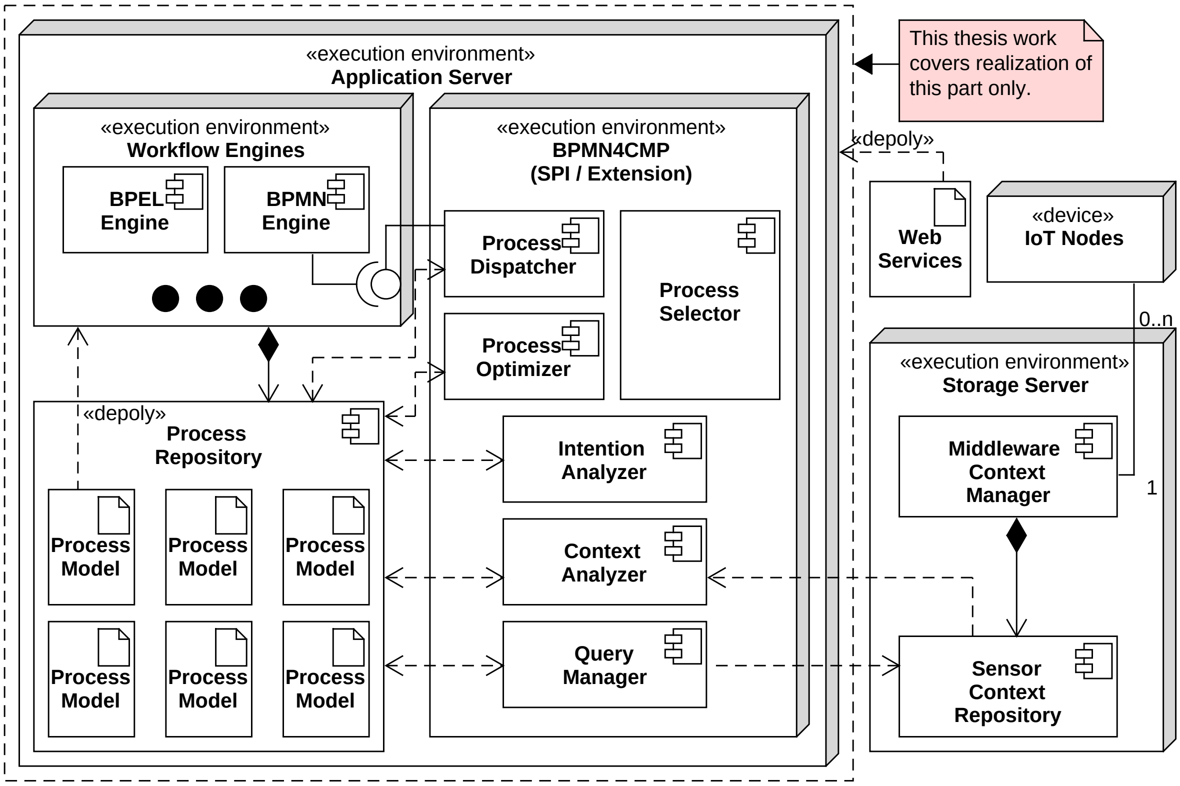

Figure 6.4.: Deployment Diagram of CES Task Extentsion to Process Engine

6.3.1. Sensors and Annotation Layer (SAAL)

SAAL refers to physical smart environments in manufacturing facilities with a massive deployment of sensors and actuators along with the Internet framework, i.e., Cloud Computing. As Gubbi et al. [GBMP13] evaluate Cloud Computing to be the most reliable service provider for next generation smart factory and smart environments, we placed Cloud Computing as the core communication channel between layers of SAAL and DAAL which can facilitate infrastructure scaling, effective monitoring, cost reduction and horizontal integration of the existing enterprise model.

6.3.2. Distribution and Acquisition Layer (DAAL)

As we have already mentioned earlier, Middleware is a software layer that provides reusable solutions to application layer, i.e., MAAL, such that persisting data can be reused seamlessly [ICG07]. Whenever anything anomalous happens in the production environment, an event is triggered by the sensor with all the meta-data and the event is logged to a Middleware. The Middleware can also be an silent observer without subscribing to the underlying Cloud based Sensor service provider. In such a case, Middleware can pull the status of progress of the production, the status of machines, and cross-disciplinary side effects of a assembly line delay due to shop-floor level disturbances, e.g., machine malfunctions, demand changes, changes in business goals, etc. On the whole, raw context can be modeled and managed using Middleware, e.g., Nexus by Lucke et al.[LCW09], CoBrA by Chen et al. [CFJ03], Context Toolkit by Dey et al. [DAS01], etc.

Generally, Middleware manager is augmented with a persistent storage medium - *"Sensor Context Repository"*, as we perceive it. All relevant data received from or sent are logged in this repository. In our work, development of SAAL along with Middleware solutions is beyond the scope. Therefore, we have implemented our solution using a Non-relational database solution provider which is capable of consuming Context-queries and producing Context-data for the MAAL.

6.3.3. Modeling and Analysis Layer (MAAL)

MAAL predominantly consists of components of three categories, i.e., Process Modeler, *Process Engine*, and *Process Engine Extension / Service Provider Interface (SPI)*. Production

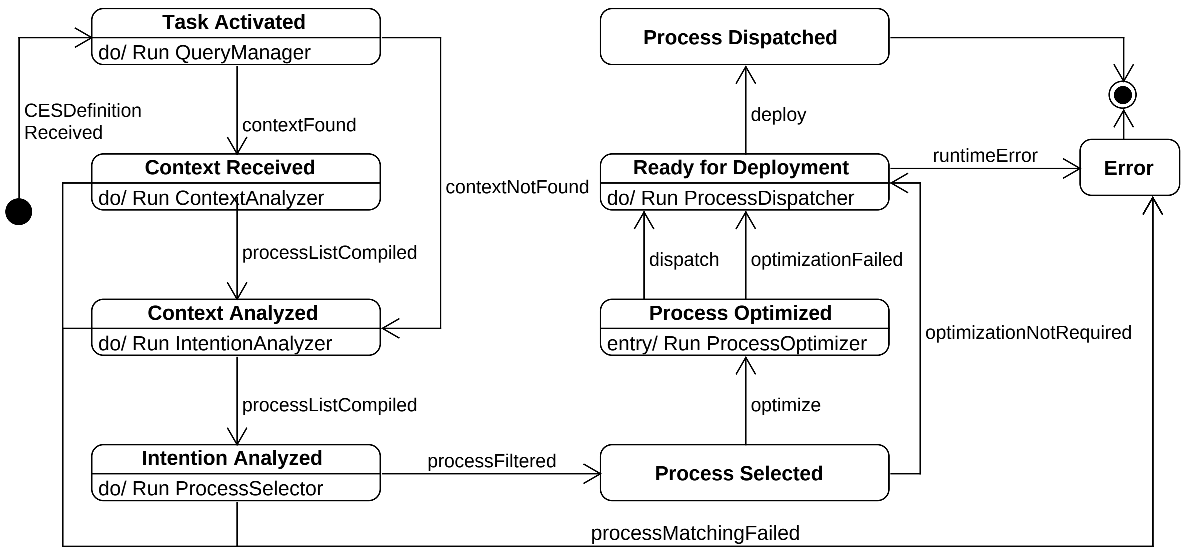

Figure 6.5.: State Machine Diagram Depicting SPI States during Execution of CES Task

processes with the CES tasks are modeled using a custom *Context-sensitive Production Process Modeling Tool* component which can edit and add CES tasks. Production Engine contains the instances of *Context-sensitive Production Process Execution Engine* which can execute standard processes along with the CES tasks developed by us with the help of SPI components.

Figure 6.5 shows the possible state transition between the sub-components defined inside the SPI as a part of BPMN4CMP extension. The transition events along with the detailed behavior each sub-component is explained in the following subsections.

6.3.3.1. Domain Know-how Repository

Every process execution requires domain knowledge, i.e., domain-specific process models and associated meta-data. We have used a centralized XML based knowledge base - "Domain Know-how Repository", as we call it, which will hold all the related process models specific to the business scenario. Listing 6.7 shows a `tProcessDefinition` stored inside the repository.

Listing 6.7: A Process Definition for a Manual Sealing Task and Associated Tasks

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessDefinitions xmlns="http://www.uni-stuttgart.de/iaas/ipsm/v0.2/"
    xmlns:ns2="http://docs.oasis-open.org/tosca/ns/2011/12">
       <ProcessDefinition processType="http://www.omg.org/spec/BPMN/2.0/" id="PMX001"
           name="ManualPacking" targetNamespace="http://www.activiti.org/">
       <Documentation>This is a dummy Manual Model.</Documentation>
       <ProcessContent>
              <ManufacturingContent>
                     <ModelPath>\\domain-know-how\\PMX001.bpmn</ModelPath>
                     <OptimizerDefinition>POPT01</OptimizerDefinition>
                     <CorrelationCoefficient>0.20</CorrelationCoefficient>
              </ManufacturingContent>
       </ProcessContent>
       <TargetIntention name="PackAndPalletize" targetNamespace="http://./././packaging">
              <SubIntentions>
                     <SubIntention name="highHRUtilization"
                         targetNamespace="http://./././packaging">
                     </SubIntention>
              </SubIntentions>
       </TargetIntention>
       <InitialContexts>
              <Context name="CON1" targetNamespace="http://./././packaging">
                     <ContextDefinition language="http://www.w3.org/TR/xpath">
                            <ContextExpression>
                            //Context[@name='availableWorkers']/ContextDefinition/
                            ManufacturingContext[SenseValue&gt;=10]/SenseValue/text() |
                                //Context[@name='unitsOrdered']/ContextDefinition/
                            ManufacturingContext[SenseValue&lt;=1000]/SenseValue/text()
                            </ContextExpression>
                     </ContextDefinition>
              </Context>
       </InitialContexts>
       </ProcessDefinition>
</ProcessDefinitions>
```

6.3.3.2. Query Manager

Query Manager prepares Context Query for the underlying Middleware and pushes the query to the Middleware. The implementation will look for the Sensor Context Repository of DAAL layer where the required contexts can be found. Context-query can be designed in any querying language, e.g., XPath, Structured Query Language (SQL), etc. As shown in Figure 6.5, if no context-data is available, the forthcoming Context Analysis is suspended and the control shifts directly to the *Intention Analyzer* instead of *Context Analyzer*.

6.3.3.3. Context Analyzer

Upon activation *Context Analyzer* receives the serialized Context-data from the underlying Middleware. In the next step, it fetches the available process definition inside Domain Know-how Repository to validate the received contexts. It filters out the initial pool of process definitions by a set of Context Conditions defined for each process definition. Finally it forwards the list of process definitions that satisfied the conditions and thought to be the best for the certain scenario to the *Intention Analyzer* as shown in Figure 6.5.

6.3.3.4. Intention Analyzer

Upon activation *Intention Analyzer* analyzes the received process definitions by filtering them with required main- and subgoals defined for the specific business scenario. If it receives the control directly from the Query Manager, then it fetches the available process definition inside Domain Know-how Repository like Context Analyzer to find the processes that satisfy the business objectives specified. Finally, it forwards the list of filtered process definitions to the *Process Selector* as shown in Figure 6.5.

6.3.3.5. Process Selector

Upon activation *Process Selector* filters the received process definitions from *Intention Analyzer* with some predefined selection strategy. It tries to select the best among the process definition available. In such a case when no selection strategy is available, then a process definition is chosen randomly. Upon the completion of process selection, the state of SPI goes to 'Process Selected' state from where SPI becomes partially dependent upon the process engine instances residing inside MAAL. Finally the selected process is forwarded to the *Process Optimizer* if optimization is required, else directly to the *Process Dispatcher*.

In our proposed implementation, we would like to have a Selection Strategy based upon *Weights* ($SelStrat_W$) (statistical probabilities) associated with each of the process alternatives. These weights can be assigned to a process heuristically or by auditing the past logs of processes in execution and thus arriving at a value statistically. Sometimes it's better to have a CES system with a naive selection strategy shown in Algorithm 6.1 than nothing at all.

Algorithm 6.1 Pseudocode for a Naive Selection Strategy based on Weights

Input/Precondition:
Set $ProcRepo = (\{ProDef_i\})_{i \in [1,n]} = \{(Id, ConRule, Goals, ComplePro, OptStrat, SelStrat)\}$
$ProcRepo \neq \phi$, Set $P = ProcessDefinitionSet$
Output/Postcondition: $processDef$, Process definition of the selected process.

```
procedure WEIGHTSTRATEGYSELECT(P, ProcRepo)          ▷ These are the input data
    maxWeight ← 0
    for all id ∈ P ∧ ∃ProDef_i | id = ProDef_i.Id do  ▷ Process Selection Step
        if ProDef_i.SelStrat_W ≥ maxWeight then
            maxWeight ← ProDef_i.SelStrat_W           ▷ Change Maximum Weight
            processDef ← ProDef_i                     ▷ Assign Process Descriptor
```

6.3.3.6. Process Optimizer

Process Optimizer executes the optimization process related to the chosen main business process by the Process Selector. This sub-component looks for any defined optimization model for the process, deploys and executes the optimization model with the required process engine instance. After the execution, it returns its run status to the callee. Finally it forwards the chosen process definition to the *Process Dispatcher* for the execution of main business process model.

6.3.3.7. Process Dispatcher

6.4. Implementation

The back-end is developed in Java and the graphical user interface (GUI) of the CES Task for Activiti BPMN Engine is developed in Java. The application depends on Eclipse environment such that a CES Task can be modeled using the palette developed by us. For the communication between components calls have been used. which can be realized using database solutions like MongoDB . These extensions are implemented on an existing eclipse-based BPMN modeling tool which is called *"Activiti Designer"*. The build is done using Maven.

1 2 3 4 5 6 7 8 9 10 11 12 13 14

The web services approach implements SOA. A major focus of web services is to make functional building blocks accessible over standard Internet protocols that are independent from platforms and programming languages. These services can be new applications or just wrapped around existing legacy systems to make them network-enabled. A service can rely on another service to achieve its goals.

A WSDL document defines services as collections of network endpoints, or ports [CCMW11]. WSDL is in practice the standard for XML-based service description. It is the minimum standard service description necessary to support interoperable web services.

E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1, March 2011

6.4.1. Use Case Diagram

6.4.2. Flow of Execution

6.4.2.1. Process Filters

6.5. Summary

[1] https://www.oracle.com/java/
[2] http://www.activiti.org/
[3] https://www.mongodb.org/
[4] https://www.eclipse.org/mars/
[5] https://maven.apache.org/
[6] https://spring.io/
[7] http://camel.apache.org/
[8] https://www.rabbitmq.com/
[9] http://www.sonarqube.org/
[10] https://cxf.apache.org/
[11] http://junit.org/
[12] http://www.slf4j.org/
[13] http://www.w3.org/TR/wsdl/
[14] http://www.w3.org/TR/soap/

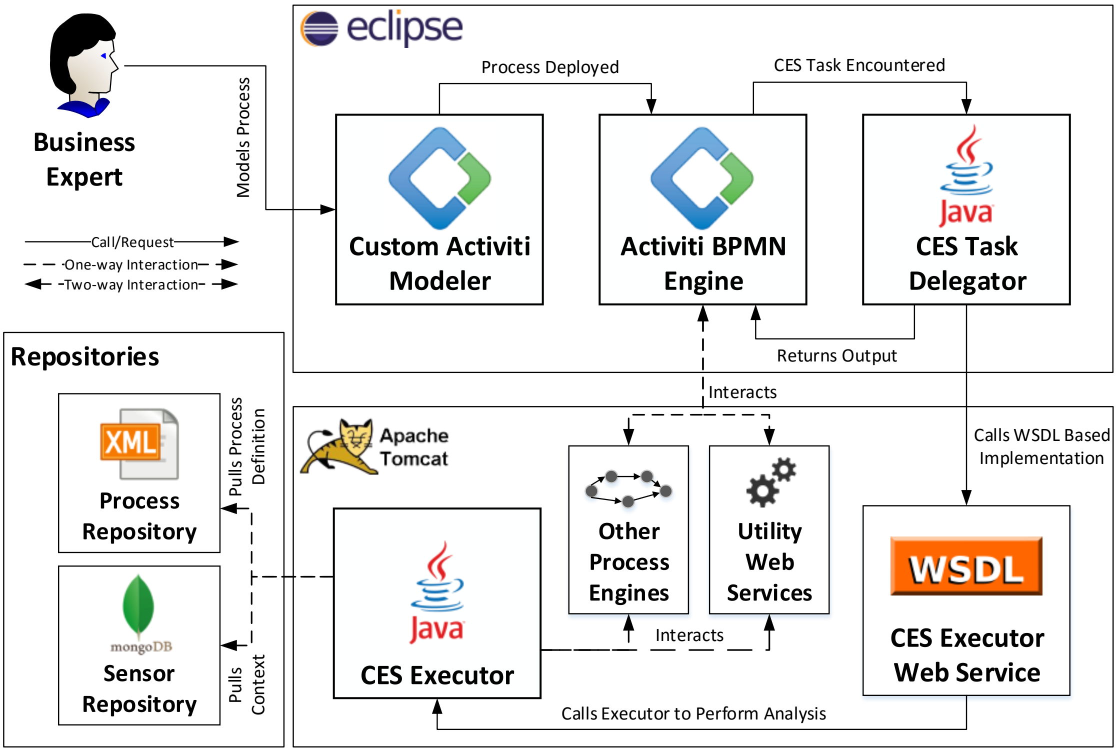

Figure 6.6.: A Holistic Overview of the Implementation

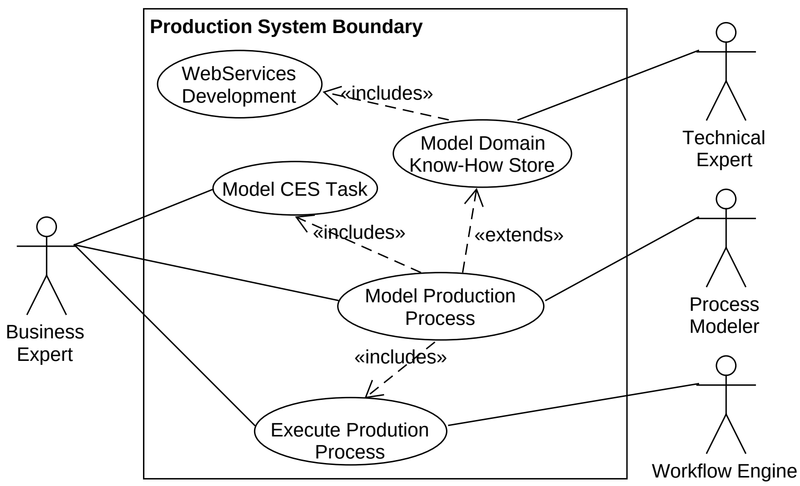

Figure 6.7.: Use Case Diagram of CES Task Executor Extentsion to BPMN

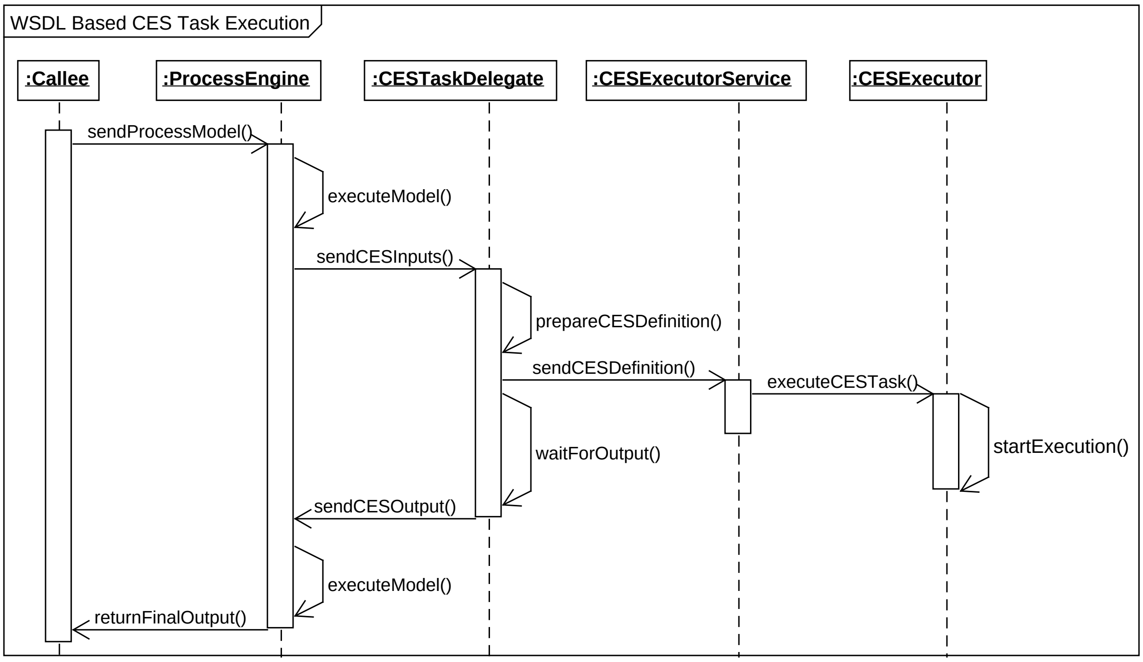

Figure 6.8.: Sequence Diagram of WSDL based CES Task Execution

Requirement Name	Mapped Proposal
IoT Incorporated Production Processes (R1)	`tCESTask`
Goal-driven Production Process (R2)	`tCESTask`, `tIntention`
Context-sensitive Production Process (R3)	`tCESTask`, `tContexts`, `tContextExpression`
Optimizable Production Process (R4)	`tCESTask`, `tManufacturingContent`
Event-Driven or Periodic Context Acquisition (R5)	`tCESTask`
Distribution of Data Across Production Processes (R6)	`tCESTask`, `tDataList`
Prioritization of Goals (R7)	`tCESTask`, `tIntention`
Resilience to Minor Changes (R8)	All
Multiple Instantiation Capability (R9)	`tCESTask`
Near Real-time Processing (R10)	`tCESTask`

Table 6.1.: Mapping of Requirements to Implementation Proposals

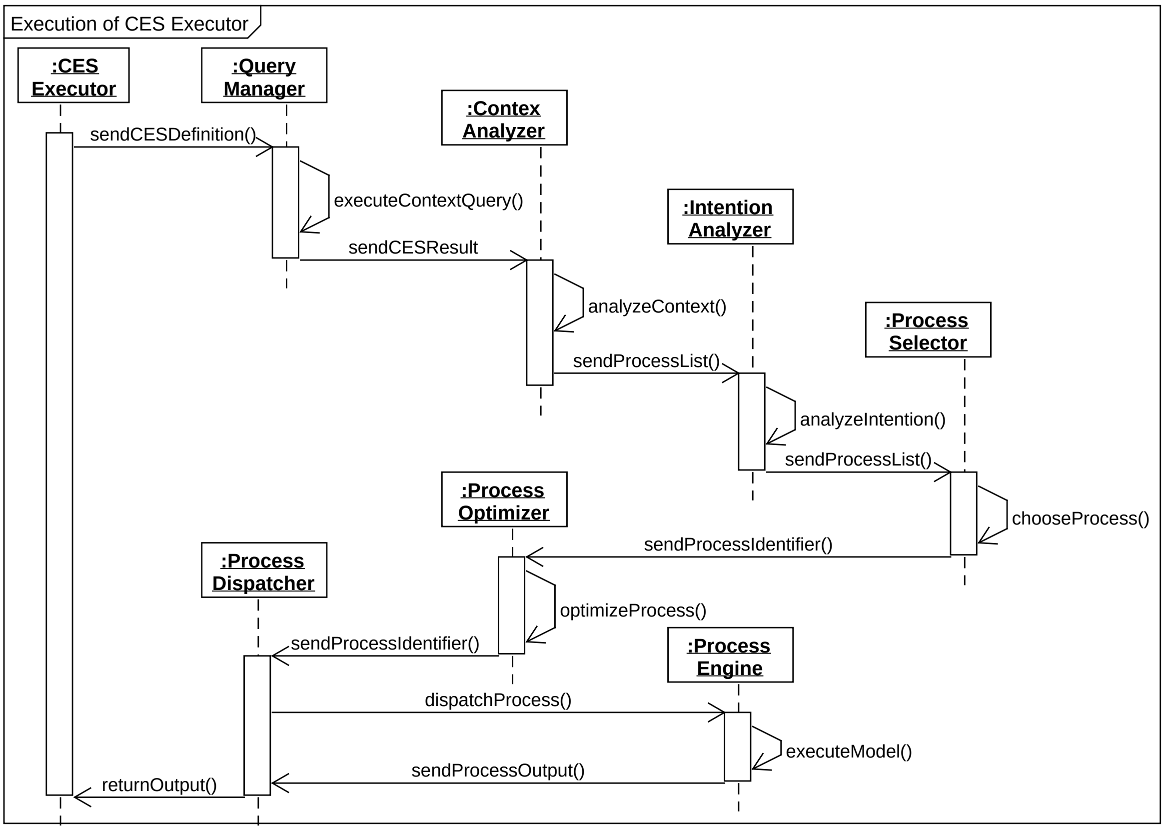

Figure 6.9.: Sequence Diagram of a CES Executor

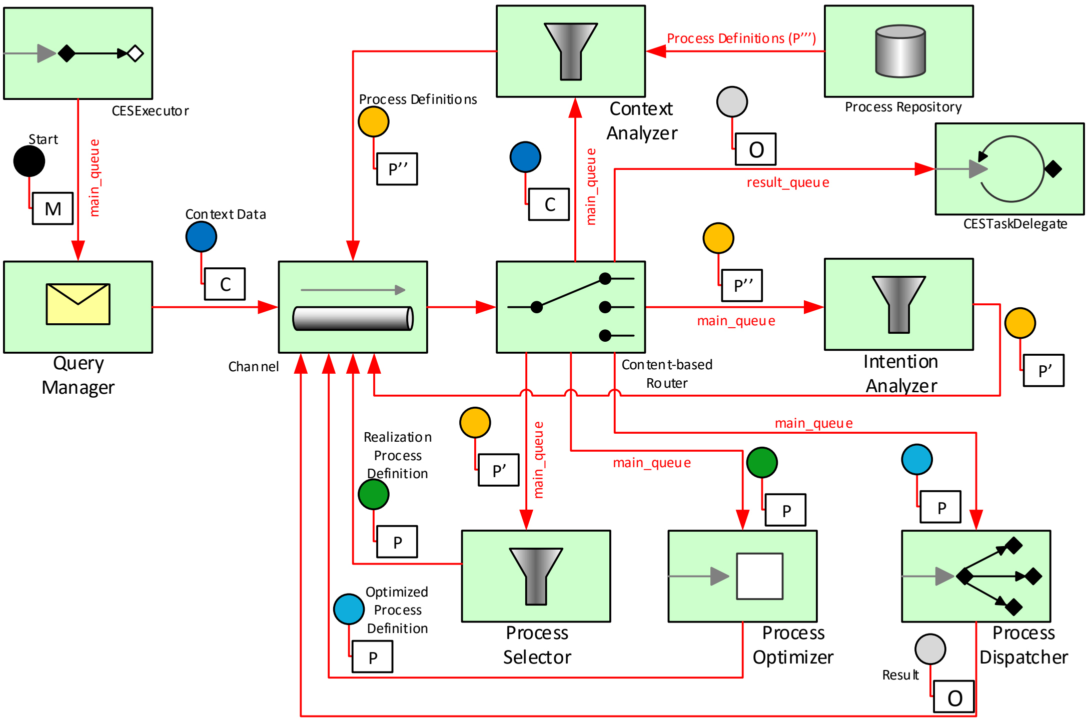

Figure 6.10.: Filtering Process among Components of BPMN4CMP Extension

7. Related Works

In this chapter, we discuss related works that are relevant to our research work whilst using the foundation and implementation we have discussed in Chapter 2 and Chapter 6 respectively. Brief comparison among the approaches towards the design and implementation of context-sensitive processes for the manufacturing and related industries is the main concern to this section.

Standard BPMN has a construct called *"Ad-Hoc Sub-Process"* that can embed multiple inner activities inside it. Such a construct provides flexible ordering compared to ordinary flow of processes [OMG11]. Thus, few modelers tend to use it as a container for a context-sensitive process modeling. Furthermore, Wolf et al. [WHR09] present extensions for process modeling language to gather relevant context data from the local environment using sensors, mobile devices, etc. Pfeffer et al. [PLS08] have presented the modifiability of abstract service composition plans using genetic programming techniques which attempts to address business goals and gains. Reichert et al. [RRHB09] and Hallerbach et al. [HBR09] present an extension to a BPMN based business modeling framework that supports process variants for a base process associated with adjustment points - which refers to a specific model fragment by few well-defined change patterns.

Similarly, Andrikopoulos et al. [ABS$^+$13], Adams et al. [ATHRvdA09] and Bucchiarone et al. [BMPR12] in their respective works have proposed to define 'What' dimension of the process without explicitly defining the 'With' and 'Who' dimensions. The actual process models are chosen during the execution based upon context data. Likewise van Der Aalst et al. [vDAPS09] have presented a constraint-based declarative framework that attempts to make process models more flexible during runtime with more support to context-sensitivity. Proposal of Marconi et al. [MPS$^+$09] includes conditional branches within flows with context conditions as guard conditions. Each flow consists of context handling activities and error handlers to manage all possibilities arising out of a execution scenario. Moreover, Wieland et al.[WKNL07] presented concepts for modeling context-sensitive workflows by deploying smart machines and collecting context information for the execution of business processes.

7.1. Evaluation

The integration of IoT enabled smart factories and Middleware by our proposed approach confirms to the requirement R1. Similarly, the approaches of Wieland et al.[WKNL07], Wolf et al. [WHR09], Andrikopoulos et al. [ABS$^+$13], Adams et al. [ATHRvdA09] and Bucchiarone et al. [BMPR12] include context acquisition from the physical devices similarly. The other described works do not address R1.

By supplementing each Intention with a Selection Strategy, we satisfy the requirement R2. The approaches presented by Solution presented by Andrikopoulos et al. [ABS$^+$13], Pfeffer et al.

[PLS08], Adams et al. [ATHRvdA09], Wolf et al. [WHR09] and Bucchiarone et al. [BMPR12] partially conform this requirement, as they haven't considered the possibility of selecting one among different process in a case where all of them have same goals.

The Context Definition of a CES task satisfies the requirement R3 clearly as our approach is context-driven upon enactment. Except Wieland et al.[WKNL07], Wolf et al. [WHR09], Marconi et al. [MPS+09], Andrikopoulos et al. [ABS+13], Adams et al. [ATHRvdA09] and Bucchiarone et al. [BMPR12], no other work address R3 completely.

By including Optimization Strategy with each Process Definition, each variant of a process model can be optimized during execution. Such an optimization functionality is not considered by anyone so only our work conforms to requirement R4. Furthermore, requirement R5 of event-driven context mechanism is supported by the works that meet the requirement R3. Only our CES approach considers the multiple parallel execution of relevant processes along with the prioritization of business objectives during modeling time. Thus, the requirements R7 and R9 are met only by us. Similarly the requirements R6, R9 and R10 are supported limitedly by these aforementioned works.

Dynamic adaptability on the basis of the current execution environment and business objectives are the prime focus of our research work. Most of the earlier approaches have a limited scope and deal with local forms of adaptation [BMPR12, ATHRvdA09, PLS08, RRHB09].

7.2. Summary

In this chapter, we briefly compared our extension with few extensions of standard BPMN and BPEL which might be a relevant extension to model context-sensitive production processes. We also mentioned the conformance to requirements of ours by each of these earlier research works. In Table 7.1, we have shown which related work conforms to the requirements led out by us.

Related Work	Requirement Name
Ad-hoc Subprocesses of BPMN [OMG11]	R9, Partially (R2, R3, R6)
Wolf et al. [WHR09]	R1, R3, R5, R6
Pfeffer et al. [PLS08]	R1, R2
Reichert et al. [RRHB09]	R1, R6, R8, R10
Hallerbach et al. [HBR09]	R1, R8, R10
Andrikopoulos et al. [ABS+13]	R1, R2, R3, R5, R6, R8
Adams et al. [ATHRvdA09]	R1, R2, R3, R5, R6, R8, R10
Bucchiarone et al. [BMPR12]	R1, R2, R3, R5, R6, R8
van Der Aalst et al. [vDAPS09]	R1, R6, R8
Marconi et al. [MPS+09]	R1, R3, R5, R6, R8
Wieland et al. [WKNL07]	R1, R3, R5, R6, R8, R10

Table 7.1.: Mapping of Requirements to Related Works

8. Conclusion and Future Work

In this chapter, we will provide a comprehensive summary of our research work and present information about the future work related to the CES construct.

8.1. Conclusion

This thesis work presents a case-study based on the concept of a new process modeling construct - the Context-sensitive Execution Step (CES) [SBLW16] in which we have integrated both automated and manual processes. Furthermore, we have proposed an abstract system architecture which can be extended or implemented by other researchers or companies based on their preferences for a context-sensitive execution of production processes. Our approach is not only context-sensitive, it also considers the business objectives of a business expert.

8.2. Future Work

This thesis work is focused on the BPMN modeling side of CES execution. As our implementation is generic and pluggable in nature, the current solution can be extended to other platforms such as BPEL, XPDL etc. The extension of ours can be validated against more complex process models which are amalgamation of manual and automated processes.

We have demonstrated a very naive optimization strategy in our case study. As optimization of business process in runtime is beyond the scope of our research focus, later dynamic process optimization feature can be integrated into our solution for an efficient and optimal process execution inside process engine.

In our case-study, we have used a XML based Domain Know-how Repository for storing process definitions persistently. Such a storage needs a GUI based tool to facilitate process definition modeling for the technical expert without any error and hard work. Furthermore, other ways of storing process definitions such as XML based database systems can be explored in future.

Due to the time-constraint, our solution is focused upon only event-driven CES execution of CES task. In future, periodic CES tasks can be designed which can be useful in certain scenarios. Similarly current intention analysis assumes one-stage goal or business objective matching process, which can be upgraded to a recursive goal-matching one without much effort.

Communication overhead among the modules inside an application server can further be optimized so that the response time and turn-around time of the CES execution will improve.

Appendix A.

List of Acronyms

The following list contains all the acronyms which are used in this document.

AI Artificial Intelligence

BPEL Business Process Execution Language

BPM Business Process Management

BPMI Business Process Management Initiative

BPMN Business Process Model and Notation

BPMN4CMP BPMN for Context-sensitive Manufacturing Process

CES Context-sensitive Execution Steps

COP Common Operating Picture

CES Context-sensitive Execution Step

CPS Cyber-Physical Systems

DAAL Distribution and Acquisition Layer

DFKI Deutsches Forschungszentrum für Künstliche Intelligenz

EU European Union

GPS Global Positioning System

HP Hewlett-Packard

HR Human Resource

ICT Information and Communication Technology

IFS Innovative Factory Systems

IoS Internet of Services

IoT Internet of Things

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IT Information Technology

LTE Long-Term Evolution

MAAL Modeling and Analysis Layer

NIST National Institute of Standards and Technology

OASIS Organization for the Advancement of Structured Information Standards

OMG Object Management Group

QoC Quality of Context

QoS Quality of Services

RFID Radio Frequency Identification

SAAL Sensors and Annotation Layer

SOA Service Oriented Architecture

SPI Service Provider Interface

SQL Structured Query Language

TCP Transmission Control Protocol

Ubicomp Ubiquitous Computing

GUI Graphical User Interface

UML Unified Modeling Language

UMTS Universal Mobile Telecommunications System

URI Uniform Resource Identifier

WSDL Web Services Description Language

WSN Wireless Sensor Network

XML Extended Markup Language

XSD XML Schema Definition

XPath XML Path Language

XPDL XML Process Definition Language

Bibliography

[ABS+13] Vasilios Andrikopoulos, Antonio Bucchiarone, Santiago Gómez Sáez, Dimka Karastoyanova, and Claudio Antares Mezzina. Towards Modeling and Execution of Collective Adaptive Systems. In *Service-Oriented Computing - ICSOC 2013 Workshops*, pages 69–81. Springer, 2013.

[ADB+99] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-awareness. In *Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[Ash09] Kevin Ashton. That 'Internet of Things' Thing. *RFiD Journal*, 22(7):97–114, 2009.

[ASSC02] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):398, 2002.

[AtAC15] Alfresco and the Activiti Community. Activiti v5.19.0 User Guide. `http://activiti.org/userguide/index.html#bpmnCustomExtensions`, 2015. [Online].

[ATHRvdA09] Michael Adams, Arthur Ter Hofstede, Nick Russell, and Wil van der Aalst. *Dynamic and Context-aware Process Adaptation, Chapter 5 - Handbook of Research on Complex Dynamic Process Management: Techniques for an Adaptability in Turbulent Rnvironments*. Business Science Reference, 2009.

[Bas14] Ron Basu. Managing Quality in Projects: An Empirical Study. *International Journal of Project Management*, 32(1):178–187, 2014.

[BL12] Mark A Beyer and Douglas Laney. *The Importance of 'Big Data': A Definition*. Stamford, CT: Gartner, 2012.

[BMPR12] Antonio Bucchiarone, Annapaola Marconi, Marco Pistore, and Heorhi Raik. Dynamic Adaptation of Fragment-based and Context-aware Business Processes. In *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pages 33–41. IEEE, 2012.

[Bro11] Joe Brockmeier. Gartner adds Big Data, Gamification, and Internet of Things to its Hype Cycle. `http://readwrite.com/2011/08/11/gartner-adds-big-data-gamifica`, 2011. [Online].

[CFJ03] Harry Chen, Tim Finin, and Anupam Joshi. An Ontology for Context-aware Pervasive Computing Environments. *The Knowledge Engineering Review*, 18(03):197–207, 2003.

[Com15] Eclipse Process Manager (Stardust) Community. Stardust Documentation Suite for Release 2.1.1. http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.stardust.docs.dev%2Fhtml%2Ftoc.html, 2015. [Online].

[CT12] Michele Chinosi and Alberto Trombetta. Bpmn: An Introduction to the Standard. *Computer Standards & Interfaces*, 34(1):124 – 134, 2012.

[DAS01] Anind K Dey, Gregory D Abowd, and Daniel Salber. A Conceptual Framework and A Toolkit for Supporting the Rapid Prototyping of Context-aware Applications. *Human-computer interaction*, 16(2):97–166, 2001.

[DDO08] Remco M Dijkman, Marlon Dumas, and Chun Ouyang. Semantics and Analysis of Business Process Models in BPMN. *Information and Software Technology*, 50(12):1281–1294, 2008.

[DH14] Rainer Drath and Alexander Horch. Industrie 4.0: Hit or Hype? [Industry Forum]. *Industrial Electronics Magazine, IEEE*, 8(2):56–58, 2014.

[DLRMR13] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A Reijers. *Fundamentals of Business Process Management*. Springer, 2013.

[EA12] PC Evans and M Annunziata. Industrial Internet: Pushing the Boundaries of Minds and Machines, General Electric, 2012.

[Erl12] Klaus Erlach. *Value Stream Design: The Way Towards a Lean Factory*. Springer Science & Business Media, 2012.

[GBMP13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[GHA11] Nicolas Genon, Patrick Heymans, and Daniel Amyot. Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In *Software Language Engineering*, pages 377–396. Springer, 2011.

[Gra14] Robert Graybill. Wolfram data summit: Data Aggregation and Analysis Challenges for Intelligent Manufacturing. http://wac.36f4.edgecastcdn.net/0036F4/small/WDS/2014/B_Graybill_Wolfram%20Data%20Summit%20-%20Nimbis.pptx, 2014. [Online].

[HBR09] Alena Hallerbach, Thomas Bauer, and Manfred Reichert. Guaranteeing Soundness of Configurable Process Variants in Provop. In *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*, pages 98–105. IEEE, 2009.

[Hen03] Karen Henricksen. *A Framework for Context-aware Pervasive Computing Applications*. University of Queensland, 2003.

[Hen14] Stefan Heng. *Industry 4.0: Upgrading of Germany's Industrial Capabilities on the Horizon*. Available at SSRN 2656608, 2014.

[Hol95] David Hollingsworth. Workflow Management Coalition - WFMC: The Workflow Reference Model v1.1. ftp://www.ufv.br/dpi/mestrado/Wkflow-BPM/The%20Workflow%20Reference%20Model.pdf, 1995. [Online].

[HPO15] Mario Hermann, Tobias Pentek, and Boris Otto. *Design Principles for Industrie 4.0 Scenarios: A Literature Review*. Technische Universität Dortmund - Audi Stiftungslehrstuhl Supply Net Order Management, 2015.

[HWS+15] Pascal Hirmer, Matthias Wieland, Holger Schwarz, B Mitschang, U Breitenbücher, and F Leymann. Sitrs - A Situation Recognition Service based on Modeling and Executing Situation Templates. In *Proceedings of the 9th Symposium and Summer School on Service-Oriented Computing (SUMMERSOC 2015)*, pages 247–258, 2015.

[ICG07] Valerie Issarny, Mauro Caporuscio, and Nikolaos Georgantas. A Perspective on the Future of Middleware-based Software Engineering. In *2007 Future of Software Engineering*, pages 244–258. IEEE Computer Society, 2007.

[Jaz14] Nasser Jazdi. Cyber Physical Systems in the Context of Industry 4.0. In *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*, pages 1–4. IEEE, 2014.

[KAA97] James D Kiper, Brent Auernheimer, and Charles K Ames. Visual Depiction of Decision Statements: What is Best for Programmers and Non-programmers? *Empirical Software Engineering*, 2(4):361–379, 1997.

[KLW11] Henning Kagermann, Wolf-Dieter Lukas, and Wolfgang Wahlster. Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. Industriellen Revolution. http://www.vdi-nachrichten.com/Technik-Gesellschaft/Industrie-40-Mit-Internet-Dinge-Weg-4-industriellen-Revolution, 2011. [Online].

[KZ15] Dennis Kolberg and Detlef Zühlke. Lean Automation Enabled by Industry 4.0 Technologies. *IFAC-PapersOnLine*, 48(3):1870–1875, 2015.

[LC12] Martin Landherr and Carmen Constantinescu. Intelligent Management of Manufacturing Knowledge: Foundations, Motivation Scenario and Roadmap. *Procedia CIRP*, 3:269–274, 2012.

[LCW08] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. Smart Factory - A Step Towards the Next Generation of Manufacturing. In *Manufacturing Systems and Technologies for the New Frontier*, pages 115–118. Springer, 2008.

[LCW09] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. Context Data Model, the Backbone of a Smart Factory. In *CIRP Manufacturing System*, 2009.

[LE06] Yair Levy and Timothy J Ellis. A Systems Spproach to Conduct an Effective Literature Review in Support of Information Systems Research. *Informing Science: International Journal of an Emerging Transdiscipline*, 9(1):181–212, 2006.

[Lee08] Edward A. Lee. Cyber Physical Systems: Design Challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, Jan 2008.

[Ley10] Frank Leymann. BPEL vs. BPMN 2.0: Should You Care? In *Business Process Modeling Notation*, pages 8–13. Springer, 2010.

[LR00] Frank Leymann and Dieter Roller. *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[LRS+15] Markus Lorenz, Michael Rüßmann, Rainer Strack, Knud Lasse Lueth, and Moritz Bolle. *Man and Machine in Industry 4.0: How Will Technology Transform the Industrial Workforce Through 2025?* Boston Consulting Group Perspectives, BCG, 2015.

[LT13] Markus Löffler and Andreas Tschiesner. The Internet of Things and the Future of Manufacturing. McKinsey&Company, 2013.

[MG11] Peter Mell and Tim Grance. *The NIST Definition of Cloud Computing*. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, 2011.

[Moo09] Daniel L Moody. The 'Physics of Notations': Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.

[MPS+09] Annapaola Marconi, Marco Pistore, Adina Sirbu, Hanna Eberle, Frank Leymann, and Tobias Unger. Enabling Adaptation of Pervasive Flows: Built-in Contextual Adaptation. In *Service-Oriented Computing*, pages 445–454. Springer, 2009.

[MWZ+07] Zhang Min, Li Wenfeng, Wang Zhongyun, LI Bin, and Ran Xia. A RFID-based Material Tracking Information System. In *2007 IEEE International Conference on Automation and Logistics*, pages 2922–2926. IEEE, 2007.

[NCS96] USA National Communications System. Federal Standard 1037c: Glossary of Telecommunications Terms. http://www.its.bldrdoc.gov/fs-1037/dir-024/_3492.htm, 1996. [Online].

[OAS07] OASIS. Web Services Business Process Execution Language v2.0. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf, 2007. [Online].

[OH12] Jennifer O'Rourke and Anika Huegle. Internet of Services. http://wiki.scn.sap.com/wiki/display/Research/Internet+of+Services, 2012. [Online].

[OMG11] Object Managment Group OMG. Business Process Model and Notation (BPMN) v2.0 Specification. http://www.omg.org/spec/BPMN/2.0/PDF/, 2011. [Online].

[OMG15] Object Management Group OMG. Unified Modeling Language (UML) Specification v2.5. http://www.omg.org/spec/UML/2.5/PDF, 2015. [Online].

[PLS08] Heiko Pfeffer, David Linner, and Stephan Steglich. Dynamic Adaptation of Workflow based Service Compositions. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pages 763–774. Springer, 2008.

[PPS11] Rabi Prasad Padhy, Manas Ranjan Patra, and Suresh Chandra Satapathy. Rdbms to Nosql: Reviewing Some Next-generation Non-relational Databases. *International Journal of Advanced Engineering Science and Technologies*, 11(1):15–30, 2011.

[PZCG14] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context-aware Computing for the Internet of Things: A Survey. *Communications Surveys & Tutorials, IEEE*, 16(1):414–454, 2014.

[RLG+15] Michael Rüßmann, Markus Lorenz, Philipp Gerbert, Manuela Waldner, Jan Justus, Pascal Engel, and Michael Harnisch. *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*. Boston Consulting Group Perspectives, BCG, 2015.

[RRHB09] Manfred Reichert, Steve Rechtenbach, Alena Hallerbach, and Thomas Bauer. Extending a Business Process Modeling Tool with Process Configuration Facilities: The Provop Demonstrator. In *Proc. BPM'09 Demonstration Track*, 2009.

[RvdM15] Janessa Rivera and Rob van der Meulen. Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations that Organizations Should Monitor. http://www.gartner.com/newsroom/id/3114217, 2015. [Online].

[SBL+95] Jonathan Steuer, Frank Biocca, Mark R Levy, et al. Defining Virtual Reality: Dimensions Determining Telepresence. *Communication in the Age Of Virtual Reality*, pages 33–56, 1995.

[SBLW16] C Timurhan Sungur, Uwe Breitenbücher, Frank Leymann, and Matthias Wieland. Context-sensitive Adaptive Production Process. volume 41, pages 147–152. Elsevier, 2016.

[SGFW10] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. *Vision and Challenges for Realising the Internet of Things*. Publications Office of the European Union, 2010.

[Sil11] Bruce Silver. *BPMN Method and Style, with BPMN Implementer's Guide: A Structured Approach for Business Process Modeling and Implementation using BPMN 2.0*, volume 450. Cody-Cassidy Press, Aptos, CA, 2011.

[SKK+11] David Schumm, Dimka Karastoyanova, Oliver Kopp, Frank Leymann, Mirko Sonntag, and Steve Strauch. Process Fragment Libraries for Easier and Faster Development of Process-based Applications. *Journal of Systems Integration*, 2(1):39, 2011.

[SKM15] Ralf C. Schläpfer, Markus Koch, and Philipp Merkofer. *Industry 4.0: Challenges and Solutions for the Digital Transformation and Use of Exponential Technologies*. Deloitte Consulting AG, 2015.

[SRHD15] Günther Schuh, Christina Reuter, Annika Hauptvogel, and Christian Dölle. Hypotheses for a Theory of Production in the Context of Industrie 4.0. In *Advances in Production Technology*, pages 11–23. Springer, 2015.

[SRTÖS09] B Scholz-Reiter, M Teucke, ME Özsahin, and Steffen Sowade. Smart Label-supported Autonomous Supply Chain Control in the Apparel Industry. In *5th International Congress on Logistics and SCM Systems (ICLS 2009)*, pages 44–52, 2009.

[SSOK13] C. Timurhan Sungur, Patrik Spiess, Nina Oertel, and Oliver Kopp. Extending BPMN for Wireless Sensor Networks. In *2013 IEEE International Conference on Business Informatics*, pages 109–116. IEEE Computer Society, 2013.

[Sun13] C. Timurhan Sungur. Extending BPMN for Wireless Sensor Networks. Diplomarbeit, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany, März 2013.

[SW07] Rachna Shah and Peter T Ward. Defining and Developing Measures of Lean Production. *Journal of Operations Management*, 25(4):785–805, 2007.

[TW10] Lu Tan and Neng Wang. Future Internet: The Internet of Things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 5, pages V5–376. IEEE, 2010.

[vDAPS09] Wil MP van Der Aalst, Maja Pesic, and Helen Schonenberg. Declarative Workflows: Balancing between Flexibility and Support. *Computer Science-Research and Development*, 23(2):99–113, 2009.

[VHGSH+15] Karolina Vukojevic-Haupt, Santiago Gomez-Saez, Florian Haupt, Dimka Karastoyanova, and Frank Leymann. A Middleware-centric Optimization Approach for the Automated Provisioning of Services in the Cloud. In *Proceedings of the 2015 IEEE 7th International Conference on Cloud Computing Technology and Science, CloudCom 2015, Vancouver, Canada*, pages 174–179. IEEE, 2015.

[W3C15] World Wide Web Consortium W3C. W3C: XML Path Language (XPath), v1.0. http://www.w3.org/TR/xpath/, 2015. [Online].

[Wei93] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36(7):75–84, 1993.

[WEN+07] H-P Wiendahl, Hoda A ElMaraghy, Peter Nyhuis, Michael F Zäh, H-H Wiendahl, Neil Duffie, and Michael Brieke. Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals-Manufacturing Technology*, 56(2):783–809, 2007.

[Wes06] E Westkämper. Factory Transformability: Adapting the Structures of Manufacturing. In *Reconfigurable Manufacturing Systems and Transformable Factories*, pages 371–381. Springer, 2006.

[Wes12] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer Science & Business Media, 2012.

[WFM12] Workflow Management Coalition WFMC. Workflow Standard: Process Definition Interface - XML Process Definition Language(XPDL) Specification v2.2. http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20(2012-08-30).pdf, 2012. [Online].

[WHR09] Hannes Wolf, Klaus Herrmann, and Kurt Rothermel. Modeling dynamic context awareness for situated workflows. In *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, pages 98–107. Springer, 2009.

[WKNL07] Matthias Wieland, Oliver Kopp, Daniela Nicklas, and Frank Leymann. Towards Context-aware Workflows. In *CAiSE07 Proc. of the Workshops and Doctoral Consortium*, volume 2, page 25, 2007.

[WvdAD+06] Petia Wohed, Wil MP van der Aalst, Marlon Dumas, Arthur HM ter Hofstede, and Nick Russell. *On the Suitability of BPMN for Business Process Modelling*. Springer, 2006.

[XYWV12] Feng Xia, Laurence T Yang, Lizhe Wang, and Alexey Vinel. Internet of Things. *International Journal of Communication Systems*, 25(9):1101, 2012.

[ZGL10] Sema Zor, Katharina Görlach, and Frank Leymann. *Using BPMN for Modeling Manufacturing Processes*. na, 2010.

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

All links were last followed on February 26, 2016

Acknowledgement

I am sincerely thankful to my mentor and supervisor C. Timurhan Sungur from the University of Stuttgart for his help, guidance, motivation and support during all the phases of my master thesis. I would also like to thank Prof. Dr. Frank Leymann for giving me this wonderful opportunity to do my master thesis at the Institute of Architecture of Application Systems.

I am also thankful to my family and friends for their help and moral support during the tenure of my master thesis.

Debasis Kar

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources than those named and all those passages quoted from other works either literally or in the general sense are marked as such. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Stuttgart, February 26, 2016 ———————————————-

(Signature)