



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

ANANDA S CHAKRABORTI  
A FRAMEWORK FOR ADOPTION OF OPEN KNOWLEDGE  
DRIVEN-MANUFACTURING EXECUTION SYSTEM

Master of Science thesis

Examiners: Dr. Andrei Lobov, Prof.  
Jose L. Martinez Lastra.  
Examiners and topic approved by the  
Dean for the Faculty of Engineering Sci-  
ences on 01.02.2017

**ANANDA S CHAKRABORTI:** A Framework for adoption of Open Knowledge Driven-Manufacturing Execution System.

Tampere University of technology

Master of Science Thesis, 67 pages, 01 Appendix pages

January 2017

Master's Degree Program in Automation Engineering

Major: Factory Automation and Industrial Informatics

Examiners: Dr. Andrei Lobov, Prof. Jose L. Martinez Lastra

Keywords: OKD-MES, Software Adoption, Layer-by-Layer Training

## **ABSTRACT**

Open Knowledge Driven – Manufacturing Execution System (OKD-MES) is a comprehensive software solution for all activities in shop floor level. Like any other software system, it comes with packages of code. Adoption of OKD-MES means adoption of technologies like web services as a standard of communication, controllers with web capabilities, Service Oriented Architecture (SOA), Ontologies in manufacturing etc. OKD-MES adoption in whole and part inspires the modern industries to move towards a digital world. Wide spread adoption of OKD-MES is possible when the user can identify the value in the system with clear metrics. The actors of this system come from different sections of the industry hence, information delivery plays a crucial role in the adoption process. OKD-MES has a layered architecture hence, a layer-by-layer adoption methodology is proposed. For every layer, guidelines, manuals and organized training contents are prepared to provide the user a working knowledge of the architecture, communication between layers, devices, etc. It is believed that training of every aspect of the system is essential for the adoption process. A use case approach is followed to provide developers, engineers and students a chance to understand working of the system. Tutorials are prepared from the obstacle faced in developing such a system. Pedagogical techniques are identified to facilitate the adoption process in both technical and commercial groups.

The guideline and training materials are validated with participants from different groups and their concerns are addressed in the training materials. An attempt has been made to capture the users verbatim with the help of a feedback procedure. This framework strives to initiate and maintain the process of adoption of OKD-MES.

## PREFACE

To understand the challenges of digitalization faced by manufacturing industry, one must be closely associated with people at the shop floor. Working in a semi-automated manufacturing plant for 2 years, I realized the importance of a robust Manufacturing Execution System. In Edward. A Murphy's words, "If something can go wrong, it will" and it is extremely valid for manufacturing. But I firmly believe that the "wrong" can be substantially minimized if a reliable information system is in place. In this thesis work an attempt has been made to understand the obstacles and challenges of adopting these information systems in factory floors and proposing a methodology to facilitate the adoption process.

This work would not have been possible without many people. First and foremost, I would like to thank my supervisor, Dr. Andrei Lobov for his guidance, ideas and support. I learnt a lot about handling research problems and academic writing from him.

I would like to express my heartfelt thanks to Sergii Iarovyj for his help and support. I remember the many hours of discussions and planning we had which made this work possible. I would also like to thank Johanna Rytönen and Anne Korhonen for helping me with the official matters and creating a supportive environment.

I would like to thank my friends and colleagues Balaji, Wael, Amir, Naveen, Samuel and Anisha for making a delightful environment to work as well as have fun.

Finally, I would like to thank my mother, Dr. Mridula Chakravarty for her belief in me.

Tampere, 25.01.2017

Ananda Chakraborti

## CONTENTS

1.	INTRODUCTION.....	1
	1.1 BACKGROUND.....	1
	1.2 PROBLEM DEFINITION .....	3
	1.3 HYPOTHESES.....	3
	1.4 LIMITATION .....	4
2.	STATE OF THE ART.....	5
	2.1 ADOPTION OF OKD-MES .....	5
	2.2 FESTO MPS .....	12
	2.3 DEPLOYMENT OF OKD-MES .....	13
	2.4 TECHNOLOGY ACCEPTENCE MODEL .....	18
	2.5 LIFELONG LEARNING .....	21
	2.6 SUMMARY.....	22
3.	METHODOLOGY .....	23
	3.1 TRAINING NEED ASSESMENT.....	24
	3.2 EMPLOYEES READINESS FOR TRAINING.....	28
	3.3 CREATING A LEARNING ENVIRONMENT.....	29
	3.4 ENSURING TRANSFER OF TRAINING .....	32
	3.5 DEVELOPINGA TRAINING EVALUATION PLAN .....	34
	3.6 SUMMARY .....	36
4.	IMPLEMENTATION.....	37
	4.1 LAYER-BY-LAYER TRAINING .....	37
	4.1.1 PHYSICAL LAYER.....	37
	4.1.2 REPRESENTATION LAYER .....	40
	4.1.3 ORCHESTRATION LAYER.....	43
	4.1.4 VISUALIZATION LAYER.....	44
	4.2 USE CASE .....	45
	4.2.1 THE PHYSICAL LAYER.....	47
	4.2.2 THE REPRESENTATION LAYER.....	48
	4.2.3 THE ORCHESTRATION LAYER.....	52
	4.3 TRAINING CENTER .....	55
	4.4 SUMMARY.....	55

5.	RESULT.....	56
	5.1 LEARNING STRATEGY.....	56
	5.2 STRUCTURE OF THE TRAINING PROGRAM .....	58
	5.3 FEEDBACK FROM THE TRAINEE .....	59
	5.4 PEDAGOGICAL TOOL .....	60
6.	DISCUSSION AND FUTURE SCOPE .....	61
7.	CONCLUSION.....	62
8.	REFERENCES.....	63
9.	APPENDIX A .....	68

## LIST OF FIGURES

<i>Figure 1. 3 pillars of The eScop Project.....</i>	<i>7</i>
<i>Figure 2. FESTO Line Distribution Station.....</i>	<i>13</i>
<i>Figure 3. Deployment Function-Waterfall Diagram.....</i>	<i>15</i>
<i>Figure 4. Deployment Diagram for OKD-MES component.....</i>	<i>16</i>
<i>Figure 5. Deployment Plan Activity Diagram.....</i>	<i>17</i>
<i>Figure 6. Technology Acceptance Model.....</i>	<i>18</i>
<i>Figure 7. Training Methodology.....</i>	<i>23</i>
<i>Figure8. Training Need Assesment .....</i>	<i>26</i>
<i>Figure9. Block diagram for training need assesment.....</i>	<i>27</i>
<i>Figure10. Activity diagram for setting up learning environment ... ..</i>	<i>30</i>
<i>Figure11. Flowchart to ensure transfer of training .....</i>	<i>33</i>
<i>Figure 12. Pre Training Know-how .....</i>	<i>35</i>
<i>Figure 13. PHL Installation guide .....</i>	<i>37</i>
<i>Figure 14. RPL Installation Guide .....</i>	<i>42</i>
<i>Figure 15. ORL Insatallation guide .....</i>	<i>44</i>
<i>Figure 16. Activity diagram for Use Case .....</i>	<i>46</i>
<i>Figure 17. Graph for Classes and Individuals for Olingvo Ontology Editor.....</i>	<i>50</i>
<i>Figure 18. List of Devices and Services .....</i>	<i>53</i>
<i>Figure 19. Port Manipulation of ORL, RPL and PHL.....</i>	<i>54</i>
<i>Figure 20. Online feedback form for OKD-MES.....</i>	<i>59</i>

## LIST OF SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
CPS	Cyber Physical Systems
DRE	Distributed, Real-Time, Embedded
EMR	Electronic Medical Records
ERP	Enterprise Resource Planning
FL	Functions Layer
F2F	Face to Face
HMI	Human Machine Interface
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IoT	Internet of Things
JSON	Javascript Object Notation
KD	Knowledge Driven
KPI	Key Performance Indicator
MES	Manufacturing Execution System
MESA	Manufacturing Enterprise Solution Association
MPS	Modular Production System
MSO	Manufacturing System Ontology
OACG	Open Architecture and Control Group
OKD-MES	Open Knowledge Driven-Manufacturing Execution System
OLS	Oil Lubrication System
ORL	Orchestration Layer
OWL	Web Ontology Language
PHL	Physical Layer
R pi	Raspberry Pi
RDF	Resource Description Framework
REST	Representational State Transfer
RPL	Representation Layer
RTU	Remote Terminal Unit
SOA	Service Oriented Architecture
SPARQL	Simple Protocol and RDF Query Language
SPARUL	Simple Protocol and RDF Update Language
STL	Structured Text Language
TAM	Technology Acceptance Model
UI	User Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VIS	Visualization Layer
WS	Web Services

# 1. INTRODUCTION

According to Open Architecture and Control Group (OACG), rapid adoption and widespread use of automation technology is achievable when someone develops and implements technical specifications that address majority of problems pertaining to businesses and publish them as open standard [54]. In today's world of manufacturing where diminishing product life cycle and increasing product customization pressure are big issues, selection and adoption of a technology is a never-ending challenge. Hall and Khan [55] in their book *New Economy Handbook* rightly pointed out that adoption is merely an absorption phase. This phase is marked with uncertainty hence; the industries are always skeptical of it. An approach is made here to structure the process of adoption of Open Knowledge Driven-Manufacturing Execution System (OKD-MES).

One of the basic goals of any adoption process is to create sustainable changes in behavior and cognition so that individuals possess the competencies they need to perform a job [1]. The purpose of this research is to develop a methodology for communicating the ideas of an Open Knowledge Driven Manufacturing Execution System (OKD-MES) to users of this system from different sections of industry and academia. This section starts with a background introducing the topic. Then the problem is defined and hypotheses are formed keeping in mind the challenges of a Knowledge driven system and effective communication. Finally, the limitations of this research are discussed.

## 1.1 Background

Technology adoption is a psycho-social process. People would not consider adopting new technology unless it gives them tangible benefit and it is easy to learn. This "ease of learning" a complex system in the ICT world is a necessity. Cathy Moore in her website [48] defines a flowchart which boils down the problem concluding whether training is required or not and if yes, what kind of training is required for the problem at hand. Here we try to analyze the same flowchart by her in terms of the problem at hand, i.e. OKD-MES.

An Open Knowledge Driven - Manufacturing Execution System (OKD-MES) approach is used to address the problems of modern MES system where flexibility and extendibility of the system are issues. In the eScop project, a Knowledge driven system is developed which has concepts, vocabulary and ontology rather than data points. The knowledge representation is done with ontologies. Web services enable the communication between devices and layers. The idea



of this thesis works to develop an approach for training the concepts of OKD-MES and best practices, which will guide the user to utilize his resources effectively.

To understand the research gap, the training problem is visualized from a multi-disciplinary and System Engineering point of view [2]. One can ask questions like; what to communicate as a part of training, which technologies of the knowledge driven system are to be demonstrated and how to evaluate the outcomes of those trainings. It is not a bad idea to even ask fundamental questions like how people learn. Pedagogical techniques like F2F training, brainstorming, debates and virtual learning are simple yet extremely useful to communicate a concept. Demonstration and simulation is one of the most effective pedagogical techniques known [4]. Considering these facts, the method of training is decided.

The eScop project has 4 layers. The lowest layer is the Physical Layer (PHL) where Embedded devices are used as a remote terminal unit (RTU). In this layer training aspect include devices and their capabilities. The knowledge representation layer is called Representation Layer (RPL). It forms the Knowledge repository of the entire system. It uses Web Ontology Language (OWL) for modelling, visualization and integration of ontologies. It contains knowledge about device functionality, device description, device interfacing and the HMI. RPL broadly contains of two parts, the manufacturing system ontology (MSO) and ontology services. The Orchestration Layer (ORL) is composed of two parts the service composer and service orchestrator. This layer is responsible for orchestration of the tasks in the factory floor. ORL interacts with all other layers to orchestrate services composition. The functional requirements are defined with 11 MES functions as outlined by MESA. Because of the layered architecture, a layer-by-layer training methodology is adopted. UML state charts are used to model the state of the system and sequence diagrams are used to demonstrate the communication in and between layers. Tools such as Sematic Workbench, Olingvo and RESTful APIs are demonstrated in step-by-step manner to make the technology easy to use and acceptable.

A right mix of Live, Virtual and Constructive model (LVC Technology) can be adopted for spreading the ideas of OKD system [2, 3]. With live model, real people operating real system are demonstrated. This can be demonstrated when actual devices can be operated with web service interface at the factory floor or in pilot cases. This is demonstration on actual system offered F2F to the trainee. This is the most effective model as any concerns of the trainee could be answered on the spot. A virtual model is where real people operate virtual system. A simulator is a good example of this approach which provides the real feel of the system in a virtual environment. It is difficult to track the transfer of training in this case. Finally, in constructive approach simulated people operate simulated systems.

An important aspect of systematic training process is transfer of training [1]. Learning transfer is the extent to which learning of one task facilitates the learning of another task. Transfer of training is a tangible benefit of training. It validates the training process and gives feedback as to how effective the training was. The training techniques may not be 100% efficient but transfer of training gives a qualitative indicator of the content. This technique helps us to see the

training and development process as a black box. Another significant aspect of the training process is to define the actors. Who is the training for? For the operator training, will be mostly focused on how to configure the system, run the system, and control the system with UI etc. For a manager on the other hand it will be more focused on the UI level where he can monitor the system.

## 1.2 Problem Definition

In present day, there exists no formal procedure, guideline or methodology for adoption of Knowledge Driven system in the industry. While working with OKD-MES the user will get overwhelmed with information and codes. There will be many obstacles in the learning process. Many questions may arise from technical and conceptual point of view. If they are not resolved in a systematic manner, the result would be resistance in adoption of the technology.

## 1.3 Hypotheses

Based on the available knowledge following hypotheses can be made –

**H1:** Conceptualization of layers of OKD-MES can be done and guidelines for adoption of OKD-MES can be produced from the obstacles faced and lesson learned.

**H2:** The development of OKD-MES can be demonstrated with a simplified use case which will help the user to understand the communication between major layers of the OKD-MES

**H3:** Lesson learnt can be demonstrated to internal and external users of the system and their feedback can be collected to analyze transfer of technology

## 1.4 Limitations

Adoption of a technology is successful in true sense when it is adopted by all the people in the organization. The procedure for adoption by different departments will be different. The process of adoption for the engineering team will not be same as the sales team. Hence, a big challenge is producing standard set of guidelines for adoption of a technology by everyone in the organization. There are many actors of the OKD-MES system and the guidelines will be a little biased for some group of people. Though training is a good way to spread the ideas of the OKD-MES system there is some limitations to this process. Training contents have to be modified and validated over the life cycle of the system. Also, the actors of the training process are from various fields of industry and academia. For a manager or owner detailed technical know-how is not required whereas for operator it is mandatory.

Before approaching these guidelines, it will be assumed that the user or trainee is familiar with the problems of existing MES. The user should be able to understand what is meant by flexibility, re-configurability and extendibility which OKD-MES brings to a conventional MES.

A use case approach is used to communicate the concepts of OKD-MES. However, it is just an abstraction of the concept and requires more learning and training to be able to make a full scale OKD-MES by the user. Another important challenge is to analyze the adoption procedure. Rate of Adoption is relative in nature and the calculations involve a larger time period to infer about it. Transfer of training can be one measure to analyze understanding of the trainee or user of the system. An attempt is made to provide the training to another employee in our Lab who is not familiar with OKD-MES and work on his feedback. But that person will not be a professional developer who wants to build his own OKD-MES. The possibility of handling and processing all concerns of actual users of this system is out of scope.

Continuous learning and skill development is a way of life in modern organizations. This equips employees with knowledge and makes them more independent and responsible. But there is also a threat of increasing cost of training in modern organization with multicultural and multilingual backgrounds.

## 2. STATE OF THE ART

To build a good methodology for adoption of OKD-MES, understanding state of the art adoption processes are crucial. The tools and technology used to adopt and disseminate such systems are also discussed. A part of the FESTO line was used to demonstrate the OKD-MES concept. Deployment of the system which was a major challenge in understanding the outcome of the system is also discussed. Finally, the lifelong learnings from similar systems are discussed. A summary is drawn at the end based on present state of the art of adoption processes and its usefulness for this adoption system.

### 2.1 Adoption of OKD-MES

Technology adoption can be defined as a choice that a company makes to acquire and use new inventions or innovations [30]. This process of adoption is affected by several factors such as cost, time, technology level etc. These factors state when and how the technology adoption can turn into technology diffusion. Technology diffusion is a process when the new technology spreads throughout the population. Adoption of a technology is a complex techno, socio-economic process. Invention of a technology is a fast process but adoption and diffusion are very slow processes being effective over time, sometimes years. In adoption of a technology like OKD-MES by manufacturing organizations, the question is not whether to adopt or not to adopt rather the question is when to adopt.

Workers and capital goods are two main resources of a company. Rate of adoption depends directly on these two resources [30]. Skill level improvement of the workers determines the rate of adoption. Proper training and know-how about the system will accelerate the adoption process. The overall skill level of the enterprise has to grow for diffusion of the technology. Diffusion is a state when the technology is so commonplace someone can learn it from his colleagues or co-workers.

For many years' companies were driven by the concept of globalization and technology. The increase in competition has led these companies to adopt new technologies to gain strategic advantage. Some companies are successfully adopting new technologies while some are failing in it. This technological innovation procedure generally has a lot of advantages like procedural benefit and cost benefit. Hence a lot of factors govern the technology adoption process. Fareena Sultan and Lillian Chan [20] in their research provided a generalized model for adoption of new technology within a firm. This model provides a comprehensive understanding of the factors or metrics.

Factors (or metrics) are the yard sticks that can be used to judge the adoption process. As pointed out by Fareena Sultan and Lillian Chan [20], there can be many factors which may or may not be quantifiable. It can be characteristic of individual such as experience. It can be group factor like ability to respond to risk or communication amongst team members. It can be company factor such as company structure or Technology policy of the organization. Individual's perception of technology will also affect the adoption process. Some examples of these factors which have been identified for OKD-MES are performance of the system, flexibility, re-configurability, worker's safety, employment safety, agility of the system, rework reduction and quality enhancement, reusability, diagnostics, decentralization of control and energy efficiency. E. M. Rogers in his book *Diffusion of Innovation* mentioned advantage of innovation can be measured under four categories economic, social, convenience and satisfaction [31]. Recently, environmental efficiency is also playing a big role in defining the metrics of a system.

G. Premkumar and Michael Potter [21] in their research identified seven factors that can evaluate an organizations decision to adopt or not to adopt a CASE (Computer-aided software engineering) technology. They divided the factors into technology variables and organizational variables. The technology variables are Relative advantage, Cost, Complexity, Technical compatibility and Organizational compatibility. The Organizational variables defined by them are Product champion, Top management support and Information System Expertise. G. Premkumar and Michael Potter [21] extended Zmud's [22] theory of "technology-push" and "need-pull" that affect the factors for adoption of new technology. In the same article they also state "Organizations adopt innovations to overcome performance gaps and deficiencies or to exploit new opportunities". This is coherent with the concept of OKD-MES where the "performance gaps" of the existing MES systems are addressed. This has opened up new scope for innovative technological models such as plug and produce.

Speaking from the OKD-MES point of view, adoption of this technology is the Final integration phase. It needs clear indicating factors or metrics which was mentioned before.

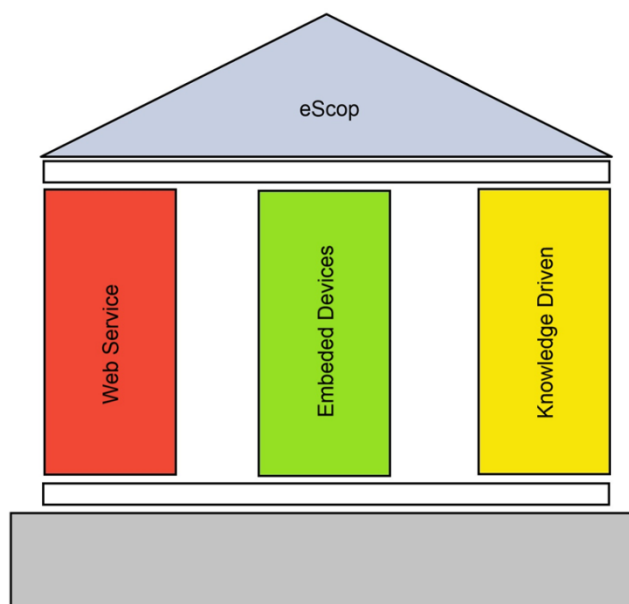
Before understanding the concepts of Open Knowledge-driven MES, one must know the current status of MES and why the MES has to be more intelligent. Commercial as well as open source MES exist in the market but a standardized form of MES does not exist. MES will have specific architecture, platform and functionality which will be capable of solving one kind of problem [7]. Information integration across the layers of the MES and MES with the ERP is a challenge. The existing MES also lacks in adaptability and re-configurability after deployment. This makes it rigid and applicable for a specific type of purpose only.

Manufacturing organizations deal with extremely large volume of production data. In an event-driven system, this data is generated whenever an event takes place. For example, presence of a pallet in workstation is a data which can lead to some action that will generate another data say, pallet presence will trigger the robot to perform an operation. A Manufacturing Execution System (MES) generates large amount of data which needs a robust information management system. This marks the advent of KD (Knowledge-Driven) systems. In a Knowledge-driven

system, there are concepts and vocabularies instead of just data points. These concepts can be represented with Ontology. Tom Gruber defines ontology as concept, relationship and distinctions that are used to model a particular form of domain. For example, a part can be a sub assembly of a bigger part or it can be a machined product out from a system hence, the context is important [1]. If an attempt is made to identify these concepts and build a system based on it, a Knowledge Model can be created.

Knowledge Modeling requires an architectural framework that will form a bridge between business needs and information technology. Service Oriented Architecture (SOA) is an architectural paradigm based on a consumer who needs a service and a provider who provides the service. OKD-MES is neutral in nature. This means it is not tied to any specific technology, standard or protocol. It is can work with plain XML or Web service family [3]. SOA is best exploited as a system with Open Architecture therefore; OKD-MES follows the same pattern. Open Architecture has clear boundaries between components and well-defined protocols to network devices. The goal is to monitor and control the Embedded devices in the shopfloor. This can be achieved by Cyber Physical System (CPS) which provides integration of Computation, Networking and Physical processes. CPSs are providing sufficient computational capabilities for low level factory shop floor devices to implement distributed web architectures [5].

The eScop project has 3 pillars. These are Web services, Embedded devices and Knowledge drive approach. Figure 1 shows this concept.



*Figure 1: 3 pillars of the eScop Project*

For “Web” services to exist there has to be electronic devices connected by a network. The devices are Embedded devices like INICO S1000 controller [32]. These Embedded devices allows CPS (Cyber Physical System) to be implemented on modern production lines [33]. CPS plays an important role in the interoperation of these devices to develop modular, robust solutions. Hence computational and physical capabilities of these devices get connected [4]. Now the devices are in place and a computational system to network it, a repository is needed. This repository will have “Knowledge” about the system. It will store information and provide concepts to connect random data points to form a vocabulary [5]. This is the Knowledge base. A system which has a Knowledge base like this can be called as a Knowledge-driven (KD) system. Combining the two concepts of KD systems and having an Open Architecture, the result is an Open Knowledge Driven (OKD) system.

Adoption of an Information system in industries is a complex process. It is difficult to say how people in the industry will respond to the different challenges that come along with the technology. The adoption process involves solving the problem with assessment and evaluation of an alternative solution [8]. This approach has a multidimensional perspective. It can prove that the solution is not platform dependent or inflexible. The adopter of the information system should be provided with guidelines for creating and customizing his own information system. The provider of the information system should aid the process of adoption by producing a step-by-step guideline for the adopter.

A study was conducted by Ajami et al. on adoption of Electronic Medical records (EMRs) system at hospitals. Despite the positive effects of the technology the adoption rate was very low. The researchers identified the barriers which was causing the low adoption rate. These are potential barriers to adoption of information systems [34]:

1. Cost of the adoption process as compared to the benefit – The benefits of the technology should motivate the user to adopt it
2. Time – The time required to learn a technology is not adequately provided by busy professionals who are associated with other works.
3. Security – Preventing of unauthorized user to access production data
4. Functionality – Right data for the right system
5. Workflow disruption – To be familiar with the technology sufficient time has to be invested to get familiar with it. This means one’s own work would be disrupted when making this effort to learn a new technology
6. Complexity – The way to use the technology might be complex. It might not be possible to do everything right with little or no training
7. Technical support – Tech support is an important adoption factor and it can be an important barrier for adoption as well
8. Formal Training
9. Other factors which can seem small but sometimes become important barriers for adoption such as connectivity, speed and institutional support.

Though adoption of technology can be used in every walk of life, our discussion of the adoption process is towards discrete manufacturing and process industries that have an existing MES in place and looking to migrate towards a Knowledge-based system. One of the reasons for adoption of the OKD-MES in these industries could be competition. Manufacturing organizations are accelerating product life cycle to combat the relentless pressure of innovation [35]. The demand for new, customizable products is rising and if the manufacturing system is not capable of handling these changes in demand, the organization will face big problems. We are now approaching towards a world where the machines at factory floor can take decisions and work autonomously to tackle any irregularities in order, process, resources or maintenance [35].

Technology transfer should be considered while training a new technology. Nazmun Nahar and Vesa Savolainen [23] mentioned the importance of IT aided training in technology transfer. They identified seven phases in which modern IT organization perform training activities. The seven phases are establishing or upgrading of computer networks, making the computer system familiar to the recipients of the technology, identifying training needs of the technology recipient, selecting proper training tools and methods, delivering the technology transfer package and tools, conducting training and evaluating the training.

Training plays a vital role in adoption of a new technology. After a realistic trial of a technology it is important to preserve the knowledge and train people on it for wider applicability of the technology. The information technology industry is continuously exposed to changes. Training and development of employees' play a crucial role in overall development of the organization. In the academic world students and researchers should be informed about the new technological developments to investigate new research areas. Collaborative learning is a process where people learn in small groups. This increases critical thinking, creativity and elaborative thinking [16]. There can be many methods of learning depending upon the location of the trainer and trainee. Face-to-face learning with real-time demonstration is an effective method of training. Asynchronous Collaborative learning on the other hand also plays an important role as it gives the learner time and freedom to understand the concepts, search for other solutions and try the solution by them.

Cathy Moore in [48] described a flowchart which answers very abstract questions like whether training is required for a scenario or not, how to grade the training procedure, how much is the management's involvement in training process etc. In her website, Cathy Moore provides a flowchart called, "Is training really the answer". This flowchart identifies measurable goals for abstract problems. This is a fine example of benchmarking. She also created "Action Mapping" which can analyze performance problems and design solution. "Activity Design" is designing challenging tasks for the actions mapped. This procedure of Action Mapping and Activity design helps to present the required amount of content to the trainee. According to [48], every stakeholder benefits from a process like this.



Pedagogical techniques such as Brainstorming, debates, buzz groups, quiz, and games are very effective in the process of learning a new technology. Dimensioning and analyzing of the pedagogical techniques can reveal that the tasks undertaken during learning a new technology can be divided into three categories; mandatory, optional and related [16]. The mandatory activities include the most important tasks which are to be performed to understand the technology. The optional tasks are the one which can be performed to support a technique and relate tasks are incidental and takes place in special situations. Other techniques which are extensively used for better learning are computer based simulation software, mobile communication, teleconferencing and on-job demonstration.

In [64], the author describes Passive and Active instructions in learning. Example of Passive learning instructions are power point presentations or F2F teaching where (from the trainer's perspective) no activity is performed or demonstrations given. In Active instructions like hands on training, an activity is performed to validate the learning.

A new technology is fully leveraged when it adds to real business values [41]. Any organization adopts or migrates to a new technology to have competitive advantage. However, this method is not smooth and rewarding all the time. Many examples say that when adoption of technology went wrong it resulted in budget cut downs, firing employees and even resulting in sick units [42]. When it comes to information system adoption people always think of when they will start getting profit out of it because an immediate return might not be visible. Soh and Markus [41, 43] came up with a process model to show when the Information system converts expenditure into asset. But this process model is based on many assumptions. Presumably, use of a new technology and adoption of it are different according to [43].

Enterprise systems (ES) are a category of Information Systems (IS) [44]. ES targets large scale integration of data. Because of this complexity is higher in ES.

System engineering approach can also affect the training methodology to be used. Training media decides the cost of training. A cost-effective approach in training is always desirable in an information management and networking solution. The solution is always subjected to change and improvement. So, it is a wise idea to keep the cost of training minimum without affecting the quality of the training. A LVC approach in training can be followed [17]. The Live model demonstrates real people operating real systems. This is the most effective way of learning. A virtual model is where real people learn with a virtual system, for example a simulator. This is implemented in the eScop project with simulators which replicate the model of the real system. A constructive model is where simulated people operate a simulated system. This is difficult to understand and implement in the form of training format. Multi player online computer games are a good example of this which can be also applied to learning.

With lots of learning material produced for OKD-MES by different partners involved in the project, there was a necessity of a container which will hold the materials and can be accessed. A web application was developed for this purpose. iSpring is an e-learning software which can

be used for sharing slides and presentation in the internet [29]. iSpring plug-in can be used to embed the presentations into an html file. It is also possible to convert the presentations into flash or videos. The slides can be shared in web conferencing mode and can be easily hosted in the form of a website. These informations about OKD-MES is available open source. These materials can be used for individual or group training purposes and user's feedback or comments are collected.

A Knowledge-based system (KBS) is a kind of information system where a computer program reasons a situation and executes a task. It can provide information about the system with KPIs. This field of developing a KBS is called Knowledge Engineering [7]. Ontologies can be used as a resource for KBS building smart automation systems. Ontology raises the level of abstraction and helps the user to think about the problem and its possible solution rather than thinking about the computational complexities involved in a problem. Ontologies can also support new ways of managing and controlling operations. Ontologies can describe all data, information and knowledge relevant to modelling the system [13]. An ontology editor is needed to write the ontologies. The most common ontology editor is Protégé. Olingvo, a Protégé inspired ontology editor, [36] is used to write the ontologies. These editors use web ontology language (owl). SPARQL and SPARUL are used to query and update the ontology model.

The complexity of a solution in Advanced Manufacturing Systems is growing every day. The amount of data is very large and due to advancement in fields like ICT and CEP these data are growing even bigger in size [10]. The user of an existing MES system is aware of its system's constraints. Adoption of a new technology or upgrading to a new technological framework can stimulate a complex set of changes across the organization. The data complexity is overwhelming. It is very difficult to understand the entire process if one does not have a good knowledge of development. It is possible to break down the entire process into small steps and try to learn the concepts.

An industrial use case approach is followed in many manufacturing research projects in the field of Ontological Engineering. As demonstrated in a use case by N. Chungoora et al. a combination of Model driven approach and ontological engineering approach can give birth to a manufacturing knowledge-sharing environment [12]. A use case approach can be used to demonstrate and test a specific part of the technology. The platform independent nature of the KD system can be demonstrated in various ways. Platform independence can be defined with a model at a high level of abstraction where the model defines a software solution methodology. UML (unified modeling language) diagrams are used to model the system. UML is graphical language that represents conceptual and physical representation of a system. It provides a bird eye view of the system as well as gives details about the structural and the physical aspects of the system [37].

The framework for overall evaluation for production systems can be adopted to those production systems that follows and implements the eScop approach. The pilot for the project covers discrete manufacturing and process industries. The approach for the pilots is the same which

has 3 core layers; Physical Layer, Representation Layer and Orchestration Layer. In both the cases the main point of focus is re-configurability of the production system [13].

The Physical Layer consists of the service enabled embedded devices. These devices can be a controller such as INICO S1000 or a system such as a Raspberry Pi. Raspberry Pi is a small, barebones computer developed by Raspberry Pi corporation. The purpose of development of such a computer is to promote computer science education in students with negligible cost [38]. In either of the cases these devices will be capable of performing deterministic real time control [13]. Device functionality description and interface description is done by the Representation Layer. The functionality description defines the integration of the device into the system. For example, when a new device is added into the system a deviceID is generated for it. Interface description defines the procedure in which a device can communicate with another device or with the Representation Layer. These embedded devices will update the knowledge model as and when a new update takes place.

The Representation Layer forms the knowledge repository of the system. It utilizes ontology modeling to support, integrate and visualize the system [13]. The advantage of having a central knowledge model core is that it reduces time and cost for rectification of system after making a change. Any change updates the system automatically and the knowledge about the change is preserved. It helps the system to manage complexity by standardizing device interfaces.

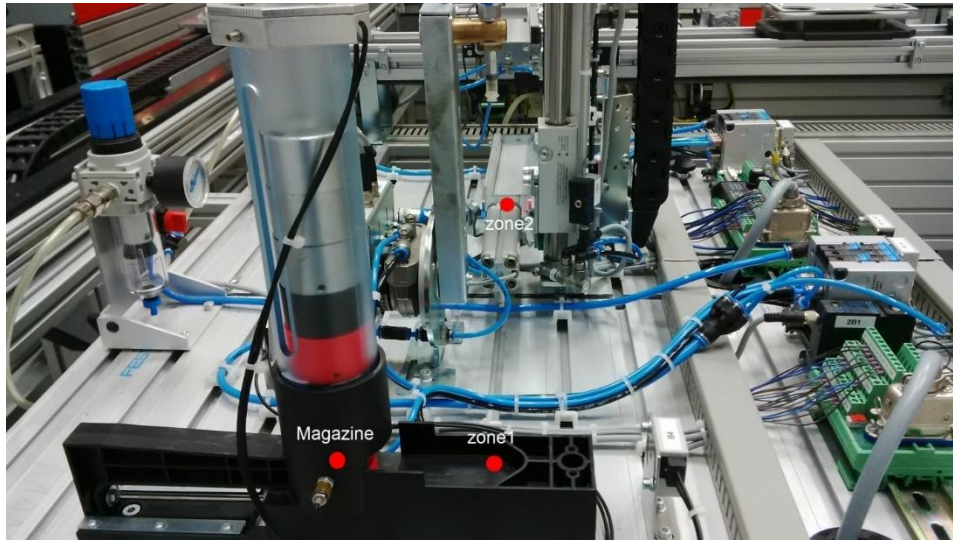
The Orchestration Layer is responsible to orchestrate the services required to control a SOA based system. The approach in the Orchestration Layer is to develop the functionalities and requirements of services leading to solution of a complicated problem.

The representation of knowledge and Orchestration of the entire manufacturing system are extremely complicated processes. In order to understand it, one has to break down the problem into small steps. A holistic approach will conceptualize the problem and show the user how to progress towards building their own OKD-MES with small steps. Holistic Models are good at presenting the problem and solution with simple content [14]. This abstraction of the problem is done parallel to the pilot cases in the eScop project with the FESTO Line.

## **2.2 FESTO MPS**

To demonstrate the idea of KD approach, a demo system is produced with the FESTO MPS line. FESTO MPS is chosen for the tutorial series as well. It is selected because of its modularity and flexibility. Demonstration of the Knowledge driven architecture is done with the help of one module, the distribution station. The distribution station has a stack magazine which can hold up to 8 workpieces. When a workpiece is present in the magazine, a double acting cylinder pushes the workpiece from the magazine to the side. This spot is named zone 1. Then the

swiveling arm puts the work piece from zone 1 to the next station. This spot is called zone 2. This is indicated in the following picture.



*Figure 2: FESTO Line Distribution Station*

FESTO Modular Production System provides a learning environment for various purposes. They are fully assembled and are a good solution for learning complex automation operations that take place in the industries [15]. The basic ideas of service definition, service invocation and orchestration can be demonstrated with the line. The methodology followed is the same as in the layers of OKD-MES.

FESTO has its own hardware and software learning modules. The modules help Engineers to get accustomed with digitalization and will become more prevalent in Industrie 4.0 [39].

### **2.3 Deployment of the OKD-MES**

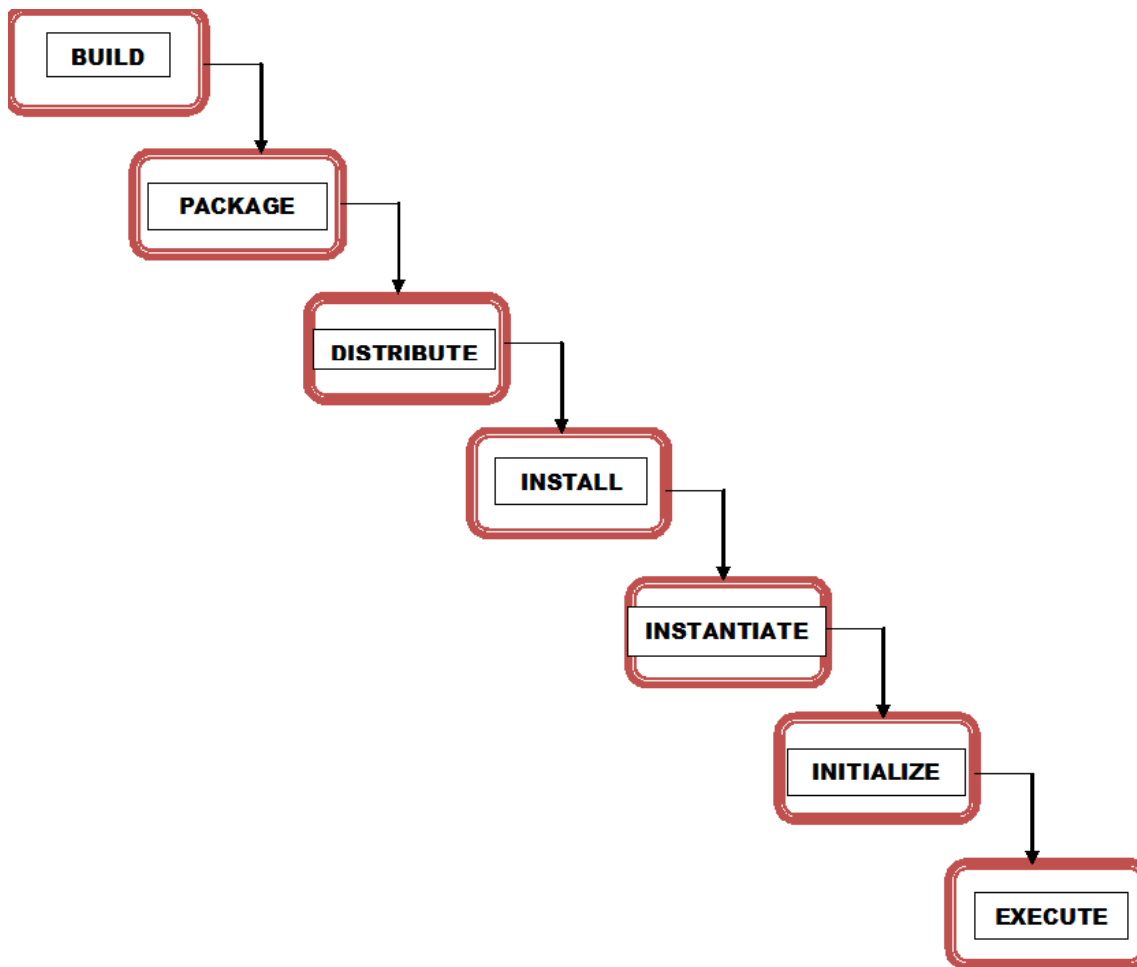
Deployment process can be very complex depending on the architecture of the solution. Software deployment can be defined as “the implementation of the definition of process on the operational environment” [18]. The deployment process covers three aspects on the organizational level; implementation of formal procedures which is required to launch the system as a solution, technological aspect which involves configuration of system and adoption of a specific tool and social aspect such as teamwork and training required for adoption of the technology. Inconsistencies in the process can be avoided with a proper deployment method. Deployment of the OKD-MES solution can be in three layers. The first is the hardware layer, then is the core framework and on top of the core are MES function layer that addresses specific business solutions.

In Enterprise and DRE (distributed, real-time and embedded) systems, a deployment plan is developed for software components. It is a challenge to make an accurate deployment plan. It

is difficult to deploy a group of components which meets various non-functional requirements [24]. Some functions which are considered in the deployment plan, such as performance and security pulls the deployment plan down acting against each other. Training and deployment are used to address usability issues of a system [25]. Usability testing can be applied to the training websites. Usability testing is a fundamental evaluation method and can be of great help when designing a training website that the user has to navigate to find the information he needs. To study and understand the real usability issues, one has to deploy the actual system. Training and exercises are essential in usability evaluation. The website will also contain tools that the user can download and simulate a real-life system. The simulator will provide a real-life experience to the user. The user can be interviewed about what he did and how he did it. Experience sharing workshops will help the deployment of a new technology which will be more user-friendly [25].

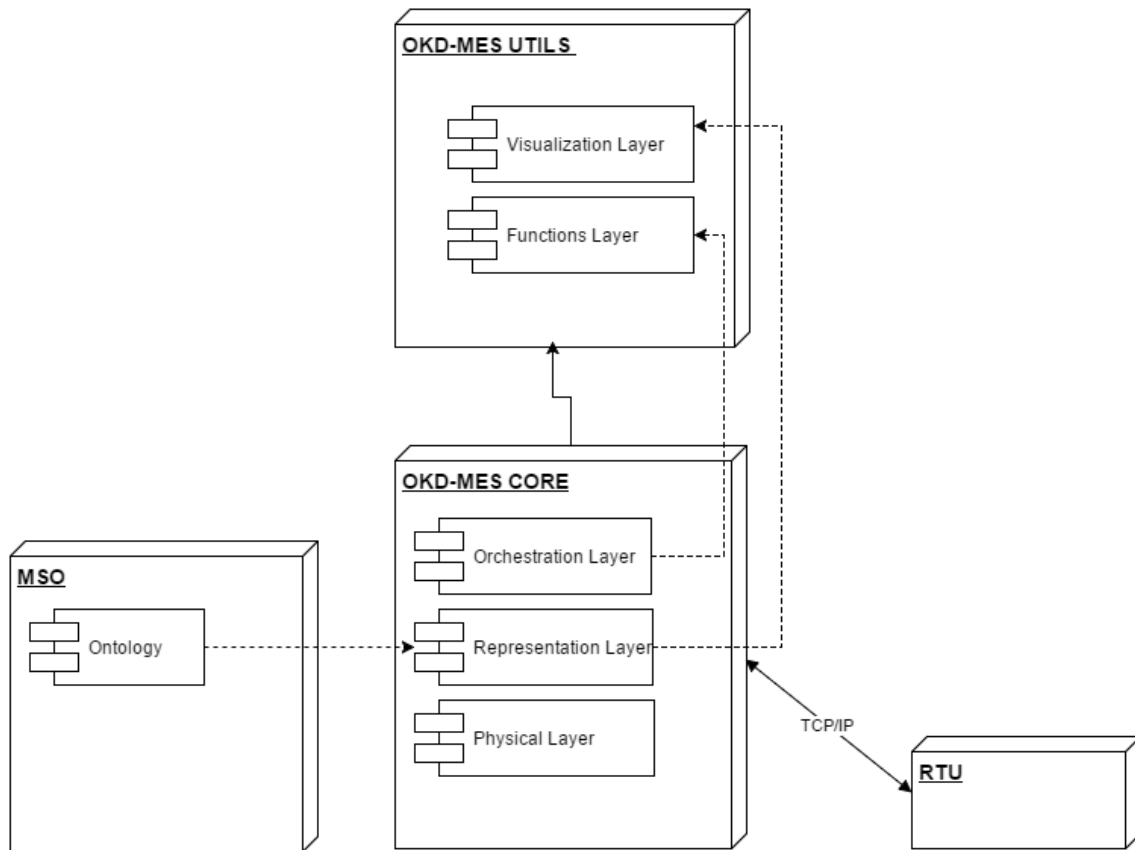
SmartFrog is a tool for automating complex deployment [24, 26]. It is a java based framework for configuring, managing and deploying distributed software systems. SmartFrog allows the user to execute a manually constructed deployment plan which will consist of series of scripts and other installation activities that runs on a specific host. It helps the user to encapsulate and manage the system so that they can be automatically installed, started and shutdown. SmartFrog is open source software where people can make their own deployment scenarios and share their experiences [26].

According to IF4IT Deployment Functions [19], a waterfall diagram can be constructed which vividly demonstrates the process of deployment of a software solution as shown in Figure 1. This can be applied to our situation in deployment of OKD-MES.



*Figure 3: Deployment Functions-Waterfall diagram*

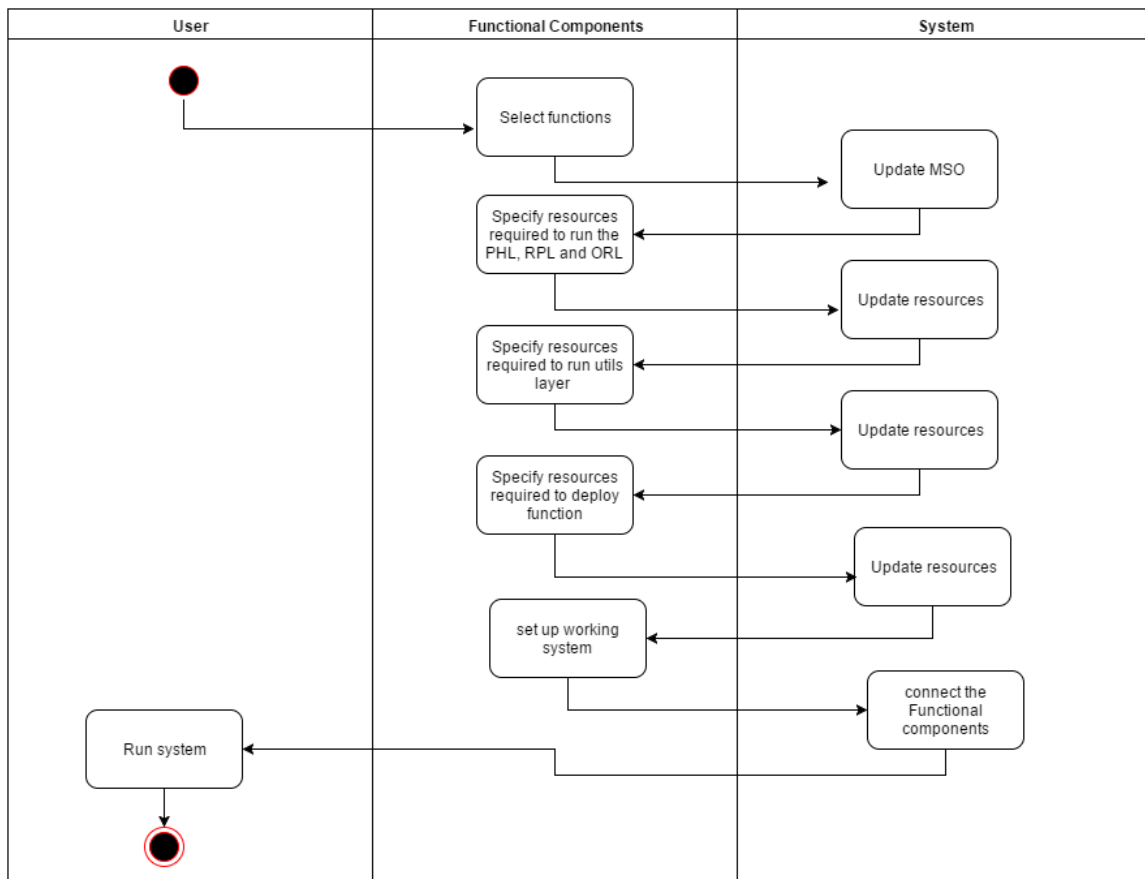
Both training and deployment are common activities in a development project [25]. Deployment can be defined as a set of repeatable, measurable and quantifiable activities that help to generate an environment where construction, preparation and delivery of a system can be done to the targeted end user. OKD-MES needs a simple deployment solution. The layered architecture of OKD-MES should be deployed in packages. The main packages involved in the deployment process are the Core, Utils and MES functions. The core contains the Physical Layer containing the devices, the Representation Layer and the Orchestration Layer. The Utils contains the Visualization Layer and any other auxiliary layers. The Functions Layer contains the MES functions.



**Figure 4:** Deployment Diagram for OKD-MES components

The core contains the core for MES execution. The core represents the actual building block of the OKD-MES. The “utils” will be a combination of auxiliary components such as the Visualization Layer. It may include communication adapter modules to integrate OKD-MES with particular protocol or just have a visualization interface. The function layer is the most interesting block in the deployment process. The function layer contains the MES function. The sequence of implementation of each of the functions is important. For example, to deploy “Resource Allocation and Status” MES Function all the time when a pallet moves to a work station “Maintenance Management” will not be called into play.

A UML activity diagram can be drawn to represent the deployment plan of OKD-MES. The UML diagram is basically a sequence diagram of sequential events that should take place to deploy the system. The steps in deployment of the system can be abstracted with this sequence diagram. The prioritization of the eScop project, MES function deployment is planned and done in the pilots. This process may be case dependent. For example, the deployment of MES function in different pilot cases would be different. The deployment process will be influenced by the requirements of the pilots and the way in which the MES Functions are defined in them. But to generalize and standardize the process a common approach can be followed. This approach is refined in the waterfall diagram shown in Figure 3.



*Figure 5: Deployment Plan Activity Diagram*

Engineer is the person who actually operates the system. The deployment tool will provide an option to the Engineer to select which MES function he wants to deploy. The choice of the user will update the MSO asking the deployment tool for resources required to be updated. On updating the resource, MSO prompts the deployment tool to “Deploy” the solution. The deployment will create a link between the core, Utils and Function packages which will enable the user to start using the OKD-MES and perform operations like Order Entry.

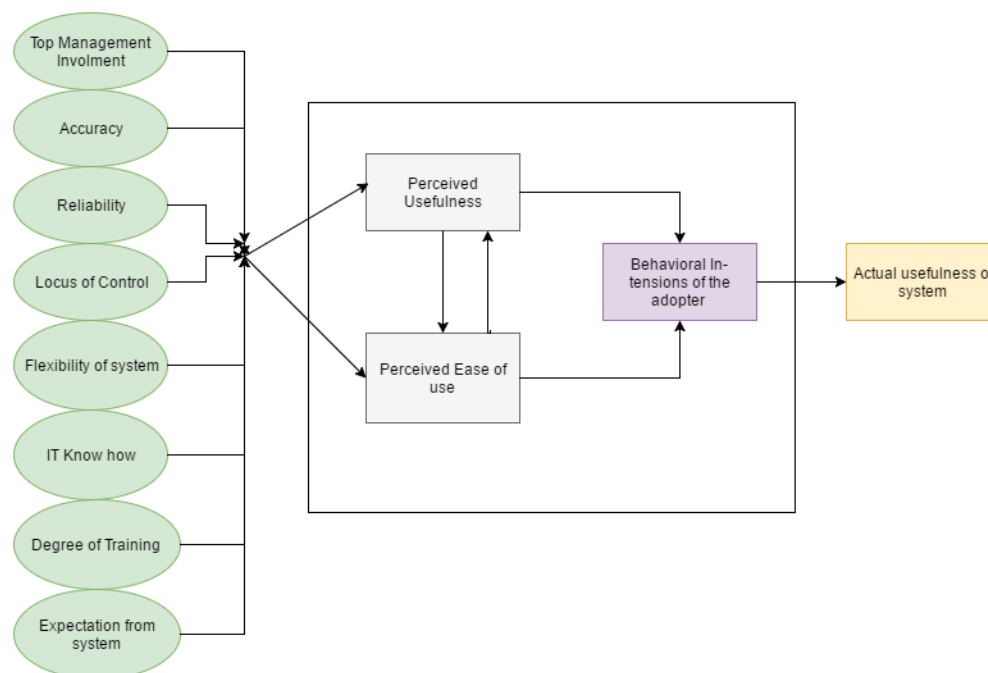
In the deployment process the MSO is updated to run the OKD-MES. There can be a simple tool or a wizard that can deploy the OKD-MES in this specified sequence. It can be a simple configuration tool but the challenge lies in deploying the MES functions. The order and the way of deployment of the MES functions are to be defined.



## 2.4 Technology Acceptance Model

A lot of research has taken place on Technology acceptance model (TAM), Theory of planned behavior, Unified theory of acceptance and use of technology, Diffusion of Innovation (DOI) and Theory of technology, organization and environment (TOE) framework. TAM captured maximum attention from the Information System Community [52]. There are two different school of thoughts for TAM. Some people believe that TAM can be used effectively for identifying the effects of the technology on humans with the help of several factors. 37 such factors are listed in [52]. The relationship of these factors can be used quantitatively to measure satisfaction the technology brings in humans. A little consideration will tell that more the satisfaction obtained on using the technology, more likely it is to be adopted. A different school of thought says there is a question to TAM's practical effectiveness. TAM has been used by the Information System community to quantify adoption.

In OKD-MES context, a TAM can be drawn by identifying some factors that highly effect the adoption of it [52] [56].



*Figure 6: Technology Acceptance Model*

Like any other conventional TAM, the factors effecting adoption are identified on the extreme left. These factors are:

**Table 1:** Factors affecting Adoption of OKD-MES framework

Factors	Definition
Top Management Involvement	The management plays a significant role in adoption of a technology. It has the authority to decide and push the adoption process at the beginning when the employees are not very inclined towards it.
Accuracy	The accuracy of the proposed solution is imperative. The accuracy determines adoption in every level of the system.
Reliability	Highly reliable Information systems will have a faster adoption rate.
Locus of Control	This factor targets the employees. If the training of the technology gives the user a strong foundational understanding.
Flexibility of the system	A flexible system in terms of order acceptance, product delivery and product customization will be an influential factor for adoption.
IT know how	The OKD-MES demands certain Information Technology knowledge from the user's side. More positive this factor is, higher are the chances of adoption.
Degree of Training	This factor targets the amount of training required to start working with OKD-MES. This can be extended to improve or modify the system.
Expectation from system	Expectation has many forms. It can be client expectations, end user expectations or employee's expectations. Expectation from the system can be personal or a group. The adoption process should strive to meet both forms of expectations.

With all the factors in place, perceived usefulness and perceived ease of use is addressed. The framework developed should be useful to learn key aspects of the OKD-MES and it should be easy to use by everyone who wishes to learn it. When all of these factors are considered, actual usefulness of the system can be understood.

Fred Davis in 1986 proposed a simple yet effective tool called Technology Acceptance Model (TAM). According to Davis, user's motivation to learn a technology can be explained by 3 factors [52]:

- i. Perceived ease of use
- ii. Perceived usefulness
- iii. Attitude of the user towards the system

This TAM targets user satisfaction of an information system. Psychologists who study this behavioral patterns of people define "satisfaction" in a broader sense [53]. It is not only what the system can do but with what complexity. It is important to give the user a sense of achievement. Bailey and Pearson quantified satisfaction of technology adopters as:

***Equation 1: Satisfaction of Technology Adopters***

$$\text{Satisfaction} = \sum w_{ij} R_{ij} (j = 1, \dots, n, i = 1, \dots, m)$$

Where,  $R_{ij}$  = the reaction to factor  $j$  by  $i$  and

$w_{ij}$  = importance of factor  $j$  to individual  $i$  [53]

For example, let's take factor accuracy as 'j' in the above equation. If twenty people in the organization think that the Information system is satisfactory, 'i' is 20.  $R_{ij}$  is the reaction to factor  $j$  by  $i$  individuals.  $w_{ij}$  is the importance which the individuals provided to each factor. Satisfaction is the weighted sum to the ranked quotients  $R_{ij}$  and  $w_{ij}$ . Satisfaction can be defined as the sum of one's feelings or attitude towards a list of factors affecting a situation. Satisfaction score can be helpful to find out how many people are likely to accept and adopt the technology.

In a broad perspective satisfaction is the need or want from the system. According to Michalos (1985), satisfaction is often studied as it is believed that satisfaction leads to sought-after feelings, attitudes, intentions, and behaviors [56]. Satisfaction, in an Information system context, can be useful to judge the difference between what is desired and what was obtained. In OKD-MES adoption framework, satisfaction of the user can be quantified according to the above formula. It can be an effective parameter to judge the adoption process. +

## 2.5 Life Long Learning

In [61], the authors introduce a specification framework for the delivery of industrial learning and training addressing the needs for the “knowledge” workers of the factories of the future. “Factories of the future” would have challenges in Industrial Learning systems. New skills would be required for the future workers. The authors in [61] state, “To that direction, an adaptation of the training content and its delivery mechanisms to the new requirements of knowledge-based manufacturing is required.” The authors emphasize on a strong multi-disciplinary background of the workers in the factory of the future. The authors also state, “Engineers and blue-collar workers will need new lifelong learning schemes to assist in keeping up with the pace of change.” Lifelong learning is regularly introduced in the corporate level to assist engineers and managers cope up with the rapid advances of technology. But it is also argued in the literature that not only technological advances but stress should also be provided on multiple objectives of the education policy such as economics, social and cultural outcomes and personal development.

In [60], the authors propose a service framework based on NMS (Networked Manufacturing Systems) and verifies its effectiveness by industrial field experiments. In the manufacturing world in the last decade, NMS (Networked Manufacturing System) and other relevant supporting technology such as the mobile agents, holonic systems, model-driven methods and middleware, have been developing rapidly [60]. The authors propose a UPnP (Universal Plug and Play) framework and validates it with a numerical framework. The use case based approach adopted by the authors help people to understand the methodology used. A similar approach is proposed for OKD-MES. A use case promotes understanding of the working system and promotes lifelong learning.

In the present world of Information systems where technological advances take place daily, a flexible learning framework is very desirable. The flexible learning system should be such that it gives one opportunity to learn anytime, anyplace. Lifelong learning is on the rise [57]. With increase of technology in daily life, people having more leisure time and an increasing pressure to keep oneself updated, the need for lifelong learning is growing. According to [57], interactive technologies has the potential to address lifelong learning.

In [58], the author rightly states, “There is no accepted definition of lifelong learning and the term has been interpreted in The Learning Age as the training of a workforce capable of adapting to a rapidly-changing world.” This will require continuous learning. The technology used today to develop and deploy OKD-MES may be outdated in the coming years. But the learning from this system will lay down foundational blocks that can be used to extend, modify or make a completely new OKD-MES. Learning resolves to addressing an idea, solving a problem or gain an understanding. In OKD-MES adoption framework, fixed metrics are used to solve the problems of present day MES like flexibility or extendibility. The understanding of the system is developed by layer-by-layer training methodology which provides adequate insights to the system. Demonstrations for other forms of training like, hand held training, online training with

the eScop Training center, description of tool used for the system are all made to address the lifelong learning. The success of the adoption framework strongly correlates to the lifelong learning generated by it.

Tools for lifelong learning of OKD-MES are researched. In [58] the author points out that the act of learning involves tools, resources and artful deployment of the environment. An attempt has been made to bring all these three factors of lifelong learning together. Learning can be assisted or individual. In assisted learning, better known as pedagogy, a teacher-student interaction is expected. Pedagogy takes a strong stand in adoption of OKD-MES. Though this is an age old method of learning but even today it is an effective mode of learning. In industries or in academic institutions a pedagogical discourse can be adopted for dissemination of the OKD-MES concept.

A conceptual framework is as important as the technology itself. For adults learning to work with a software or other technology a flexible learning system should be developed. Development of pedagogical software agents are promising area of research and commercial development [58]. For example, commercial websites such as udemy, Codeschool, Lynda etc. are great frameworks to start learning the basics of any programming language.

## 2.6 Summary

Adoption of an information system by a manufacturing industry is a complex process involving scientific and humanitarian approaches. Manufacturing industries often face problems when using, deploying or learning these kinds of systems. Challenges are also faced in updating legacy systems and motivating work force to learn new technologies. Well-known techniques such as TAM show that some factors are crucial for the adoption of technology. State of the art research in this field provides metrics, factors and indicators which guesses about the level of adoption. People understand that pedagogical techniques and training are essential. But there exists no standard methodology which can be used to adopt OKD-MES or systems of this kind. This will require understanding of development and working of the system as well as the art of information delivery to other people. In most cases the learning gets compartmentalized and is accessible or understandable to only a few. The knowledge created by OKD-MES should be disseminated to every section of the industry and academics who wish to work with this system and not just a selected few and promote a lifelong learning for it.

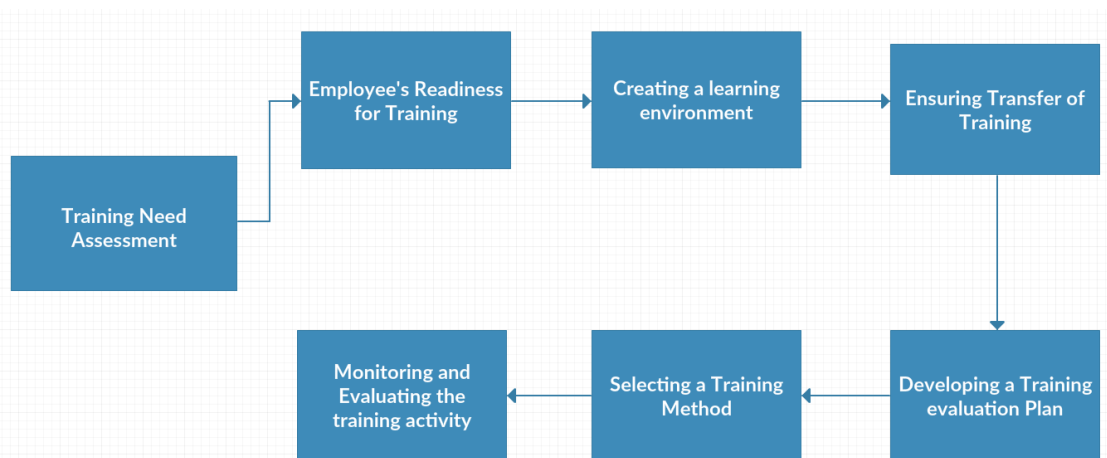
An approach to structured learning of OKD-MES can be done with Cathy Moore's idea of Action Mapping and Activity Design.

### 3. METHODOLOGY

A methodology of adoption and dissemination of a new technology should be long term. To develop a methodology is a complex affair because the factors effecting such a process is dependent not only on the technology but also on people who will be working with it.

Adoption of a Technology depends on many factors. Sometimes very sophisticated technology with lot of features is not well adopted by organizations. The organizations using automated manufacturing systems or organizations which are just starting to automate their production can be potential customers of the OKD-MES technology. As the OKD-MES is open source, cost of the system will not be a factor in adoption but social, economic and environmental aspects will be called into play. Human beings are resistant to change. Hence the learning of new software or a new system may not be embraced with open arms in the industries. It is a well-known fear among the workers that automation might lead to job cuts. On the other hand, the manager may be concerned about the power consumption, modularity of the system, diagnostics, performance efficiency etc. Hence, there exists a need for guidelines which will induct the user to the system. It is believed that training and continuous learning techniques will play an important role in adoption of OKD-MES in industries and academia.

To define a methodology for training the concepts of OKD-MES, a fundamental question is asked, what is needed to be learnt to understand the concepts and working of the system? There can be many answers to this question. The approach should be multidisciplinary and multilevel. The complete procedure of training involves training design and delivery, training evaluation and transfer of training. Training design can be represented with a flowchart which analyses the entire training procedure from a psychological and social point of view [Tata Mc-Graw Hill Company, 2013]:



*Figure 7: Training Methodology*

The training methodology proposed in Figure 7 can be used to implement a layer-by-layer training. The entire cycle in the methodology should be executed for the OKD-MES core and utils training. The need assessment will dictate why the layer-by-layer training in the form of guidelines and instructions are needed. Employee's readiness should be understood for the system. To understand the basic of OKD-MES, one has to learn the steps involved in working with or developing PHL, RPL and ORL. A classroom learning environment or an industrial learning environment can be created. If Vocational training has to be provided for the OKD-MES, an industrial environment is preferred. There are 2 sides of this training methodology, technology and people. Though learning the technology is duly important, it should not be forgotten that in industry and academia people have variable capacity to learn and comprehend new technology. Rather than providing them with thick blocks of code a simplified form of the system will make people understand the overall architecture and working of the OKD-MES which is the prime target.

Transfer of training requires consistency. When the layer-by-layer guidelines are provided to the trainee, the systematic approach will help him grasp one concept and then move on to another gaining a control over the topic. Use case for example are beneficial in this case. Tutorials that the user can follow and make his own OKD-MES will give a sense of accomplishment to the user. Technology transfer is a continuous process which tries to find a match between the companies need and newly available technology that can serve the need. Research and development teams at the industry would also benefit from a methodology of this kind. The layer-by-layer learning methodology for OKD-MES can also help those people to adopt to the system who are searching continuously for new technology to implement in their work to tackle competition.

The training method would be remote at this point. Handheld training can also be provided and the layer-by-layer guidelines contains necessary materials for both remote and pedagogical training. The methods widely used in the layer-by-layer training are presentations, videos, realistic simulation and use case. The environment for realistic simulation should be safe. The trainee should not be allowed to learn on real production systems. So, a computer based simulation and understanding of the work is important. In the layer-by-layer training two simulators are briefly discussed.

### **3.1 Training Need assessment**

Need Assessment can be performed in the organization level or personal level depending on the task requirement. Need will drive innovation which can improve the state of the art of the entire production system. The main purpose of this need analysis is to make the user realize the problems and shortcomings of the current shop floor control level of automated production systems such as Manufacturing Execution Systems (MES). When the problem is understood

clearly, it is easier to introduce concepts such as “Open Automation” and “Big data”. The technologies that can be used to build open automation system can be stated here. The customer of this technology can be identified at this stage. The two different application targets, the discrete manufacturing and the process industry are identified. Pilot cases will serve as testbeds for validating the solution in discrete manufacturing and process industry.

The employees here pertain to the stakeholders of the system. The stakeholders are kept in mind while designing this approach. The actors of the system can be from different section of the organization. They can be the owner, the manager, the operator etc. The knowledge level and technology each actor should be exposed to will vary considerably.

Training is viewed as a tool for skill enhancement. The actors who wants to learn this technology should be provided with information on what is OKD-MES and how can OKD-MES improve flexibility, make their manufacturing system more modular and save time and cost. The learning modules includes the core of the eScop project consisting of the Physical Layer, Representation Layer and the Orchestration Layer. Presentations and demonstrations provide a high-level view to the architecture of OKD-MES. It produces the basic and overall understanding of complicated concepts such as Service Oriented Architecture (SOA). A use case approach is the best way to learn the communications between and inside the layers, the working of the overall system, the technologies used such as RESTful web services etc. An actual physical line such as FESTO MPS or FASTory at FAST Lab in Tampere University of Technology demonstrates the concept of OKD-MES which will bring a touch and feel experience to the learner. It can be argued that in the last case, transfer of training will be the highest.

The need assessment can be represented with the help of a flowchart based on Cathy Moore’s training need design [48]. The idea is simple. Before generating resources and emphasizing on training discourse, the flowchart in figure 5 is followed to answer the question, is training really the answer to the problem. The problem, which not being able to work with OKD-MES can be due to environment, knowledge, skill or motivation of the worker. Action Mapping is done to identify which are the areas where training should be provided and what would the nature of that training be.

Environmental problems may originate from unsuitable or too sophisticated equipment, lack of expertise and cultural differences. Identifying knowledge gap can reduce unnecessary training. If the gap is in hardware or software, focus is on that field. Business knowledge gap should be bridged for any investor or stakeholder of the system. Realistic simulation is a very effective tool which can make the user understand the working system. Some common tools for OKD-MES are mentioned in the flowchart. Skill level of the employee can be enhanced with demos, videos, presentation and other learning material of that kind. Motivation level of trainee is very important as well. The lack of motivation can result due to lack of environment, knowledge or skill. At every level, solid measurable goals are set. Achievement of those goals signifies success in learning.



If this flowchart is followed the training activity will get streamlined and what exactly to train becomes more visible.

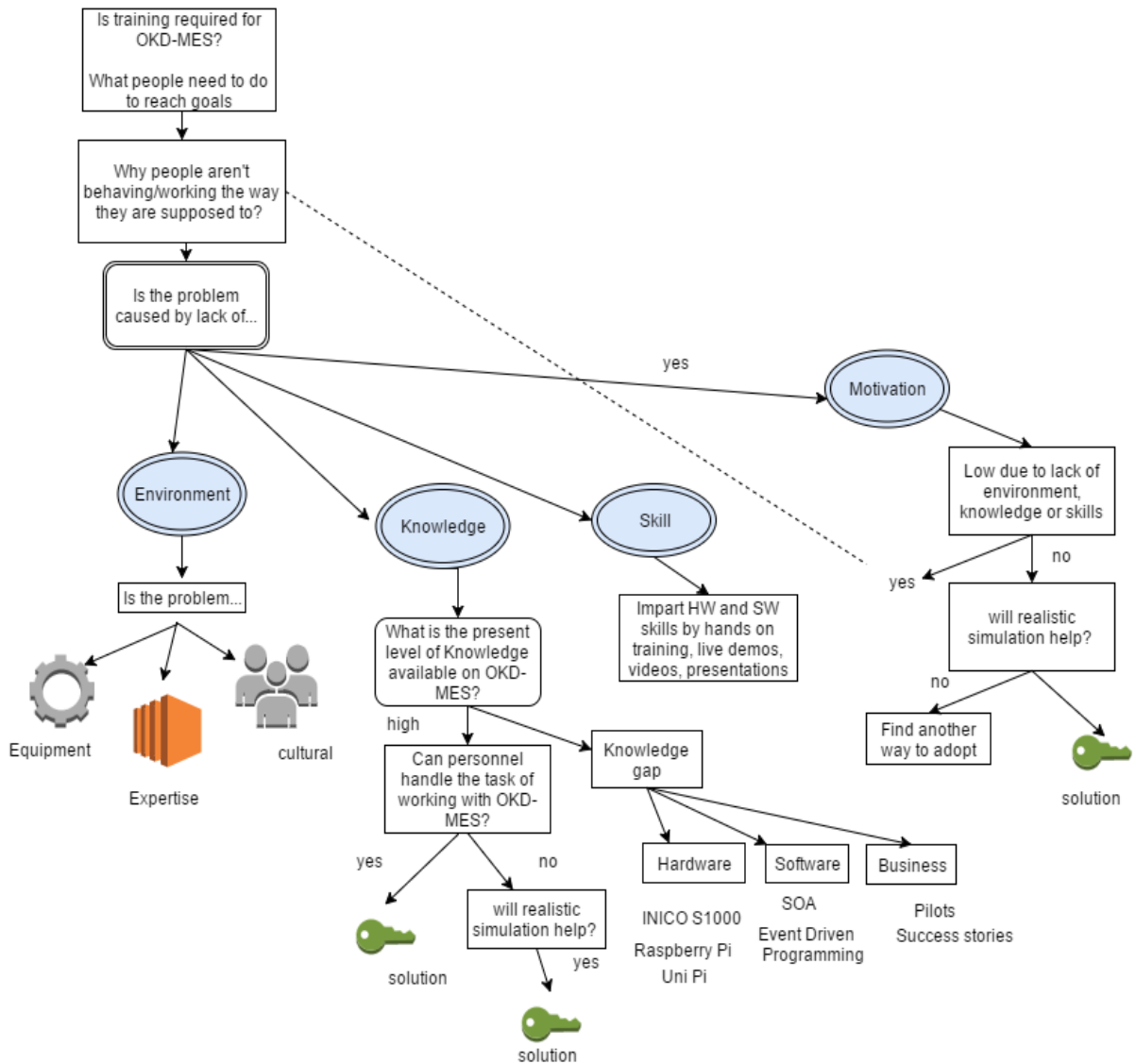
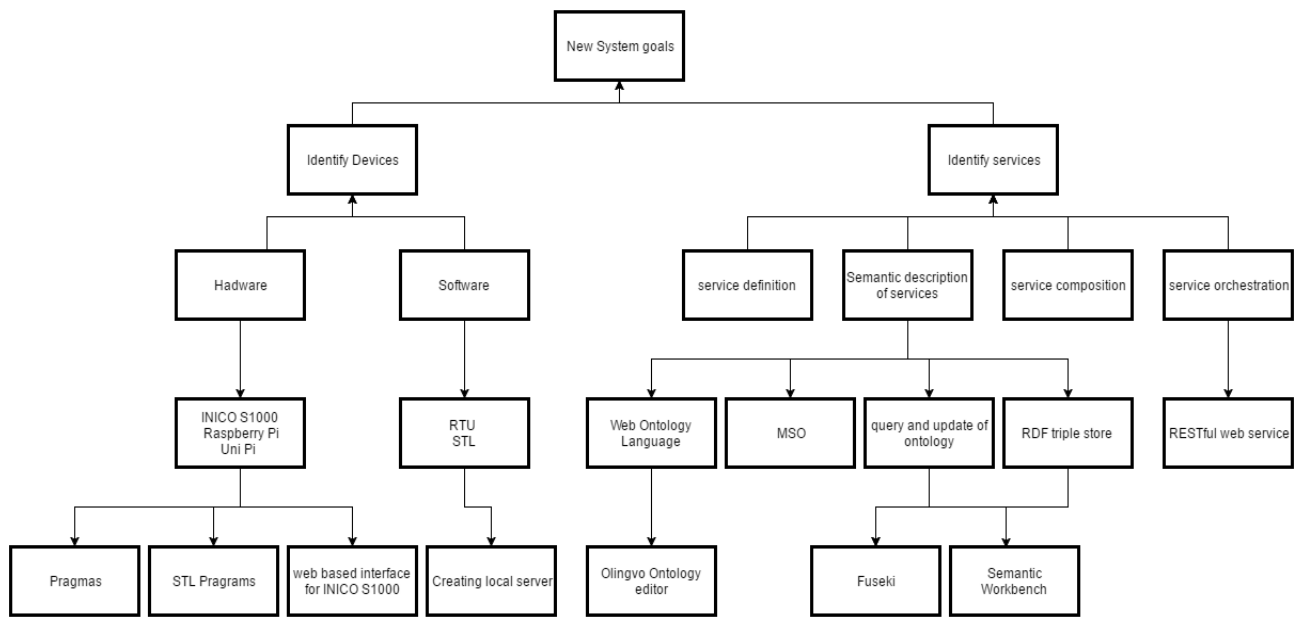


Figure 8: Training Need Assessment

The need of learning a new system like OKD-MES is based upon the existing problems of MES and legacy systems in the industry. To migrate from those systems to OKD-MES, the building blocks of the system must be understood. At the lowest level, the OKD-MES can be broadly classified into devices and service. The new system will propose and ensure learning and it can be ideated by a block diagram.



**Figure 9:** Block Diagram for Training Need Assessment

Overall need assessment of the training program is split into 2 parts, the device level and the service level to demonstrate the SOA paradigm. The outcome of learning of OKD-MES is validating the metrics as proposed earlier. The metrics are achieved with implementing SOA in the factory floor. So, the new system has to adopt the SOA paradigm. Learning an OKD-MES device involves learning of basic hardware and software. INICO S1000 and Raspberry Pi with Rex Control system compose the hardware layer of OKD-MES. Pragmas for STL programs, web based interface of INICO S1000 are potential learning mines to understand the working of the system.

SOA brings a service based thinking to the shop floor control level. These services are building blocks of OKD-MES. Understanding the service are very important not only in the context of OKD-MES but SOA in general. This will help the trainee to learn SOA approach and not just OKD-MES. Service definition, service composition and service orchestration are the principal value adding steps of the OKD-MES.

Another extremely important value adding step of OKD-MES is Semantic description of service. A target of the training process would be learning the semantic modeling of the process and storage, query and updating the ontologies. The technologies that this process demands are also considered in the implementation where step by step guideline is provided to the user on how to work with the system.

### 3.2 Employees Readiness for Training

The readiness of employees who will deal with the technology directly has a huge impact on adoption of the technology. Different groups of people can be formed like employees, managers, owners, workers, students etc. and study Rate of Adoption of the technology in every group. According to [45], the rate of adoption of a technology is relative and can be quantified as:

*Equation 2: Rate of Adoption of a Technology*

*Rate of Adoption of a Technology =*

$$\frac{\text{No. of people who start using the technology in one group}}{\text{No. of people who start using the technology in the other group}}$$

Organization's readiness and employee's readiness to change are two completely different things. In a popular blogpost in [46], a story of a company was mentioned where the manager said, "we are ready to change, the hardware and software are set up and employees have been trained". When an informal assessment was made with the employees, they had a different story to tell. The employees were trained but not much insight was given to them on why the change was required and what role they have to play in making that change. OKD-MES requires a shift not only in the hardware and software components but also in the mindset of people to migrate from legacy systems and adopt the new technology. Different group of people might have their own comfort zones. When OKD-MES demands more knowledge or expenditure than the legacy systems, resentment to its adoption is expected. But due to the relative nature of rate of adoption, some group would be faster to adopt. For example, students of Factory Automation and Industrial Informatics would be more willing to learn OKD-MES than workers in the industries who are working with legacy systems for many years.

One way of promoting the employee's readiness to learn is to bring the technology to him even before he starts using it. A simulator-driven approach or abstraction of the system with a simplified one is used in OKD-MES. Attempts have been made to bridge the gap between research and implementation. Tangible benefits obtained from the system would result in employee's readiness to learn and implement it.

The readiness can be described as the gap between what the employee has and what he wishes to learn. Providing sufficient knowledge in an easy and simple way is important. It should be considered that even beginners can be a part of the adoption process. The flow of information should be simple and easily transferable.

### 3.3 Creating a learning environment

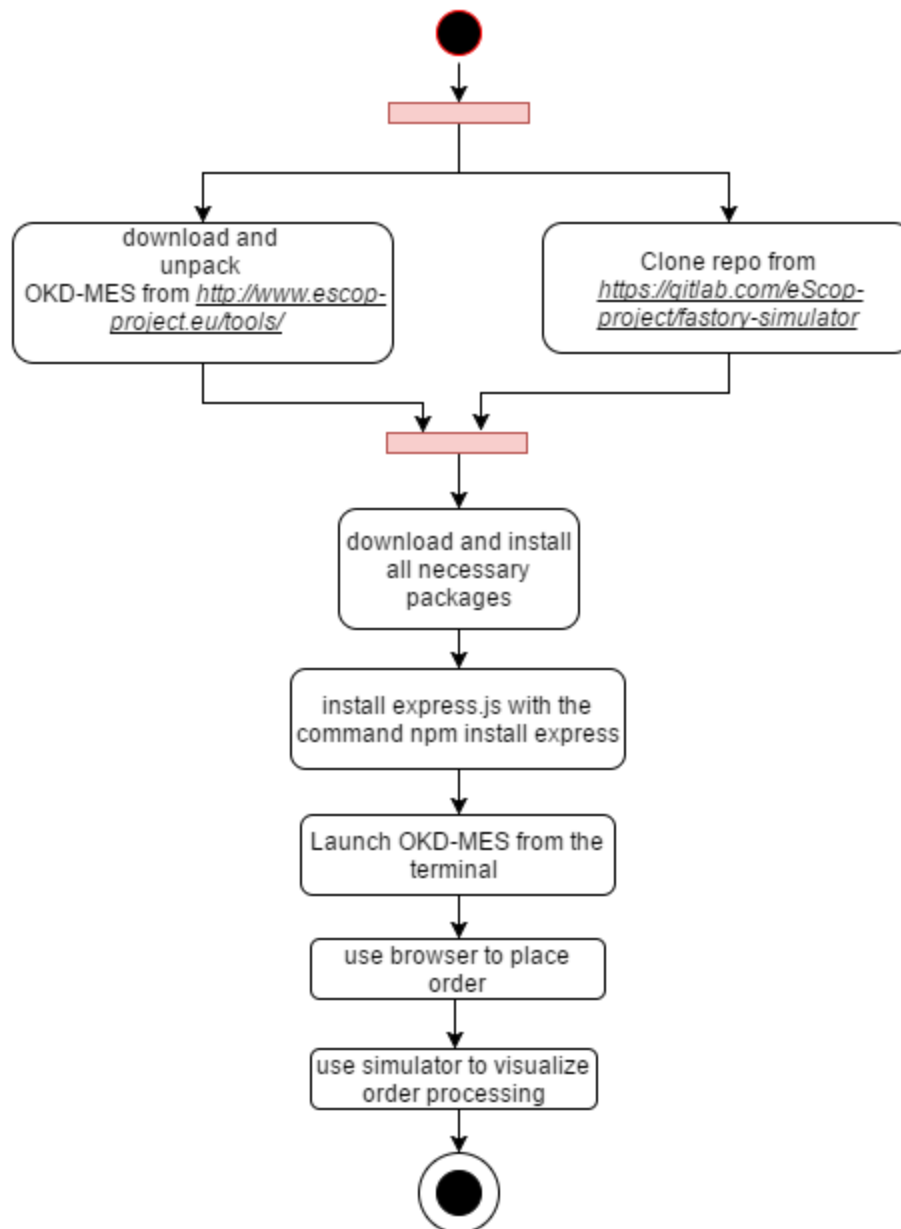
Learning environment is often pictured as class rooms with white boards, computers and projectors and a person talking about the topic. In the industry, it is viewed as intensive training programs where employees attend whole day sessions, seminars and workshops. However, pre- and post-training learning gets little attention. The accessibility of information is not a problem but using the right information as a tool for solving problems is a challenging task.

Virtual learning has become very popular in the last decade. There are many definitions of virtual learning. The most interesting one is; “static web content which contains information about the topic” [47]. This definition can be further extended to say a static webpage which contains the necessary instructions to set up, install and start working to exploit the benefits of OKD-MES. A virtual learning in a virtual university can also provide degrees and diplomas. The training on OKD-MES would not lead to a degree or a diploma but it will impart sufficient knowledge to the user so that he can understand and start working with OKD-MES.

To setup a learning environment and describe the process for it, an activity diagram is adopted.

The processes to be learnt are parts of the whole OKD-MES system. These processes are a starting point for setting up a learning environment. The environment is expected to be a lab or factory floor but it is also possible to get a touch and feel experience with the simulation. So, individual learning is also possible. The environment set up can be broadly classified as download of open source software, set up, configuration, installation and running. Major hardware and software involved in the process is used to set up the learning environment.

A process of setting up a learning environment is discussed in Figure 10. The activity diagram simply describes the necessary steps to be undertaken to set up a learning environment. The technologies are discussed in the implementation section where the steps in the activity diagram are combined to form a layer-by-layer guideline for OKD-MES.



**Figure 10:** Activity diagram for setting up learning environments

The first step can be considered as downloading and unpacking the system. For that the necessary steps should be provision of links and websites from where people can download the solution. Sometimes the software is shared with the help of code storage and sharing engines like Github. The tools are available online in the website of the project where the trainee can download the tool. The environment here is a virtual environment which will help learning. This server is used to launch an express.js application. For simplicity, a simple batch file is made. When that batch file is run, the OKD-MES solution is launched from the terminal. The idea is to make a plug and play system where thick blocks of code is not required to be written to run the system. The system is rather easily deployable across any hardware or production line.

OKD-MES is Plug and Play type. In the PHL it means that the system should be linkable to the sensors and actuators. Ontology allows the description of a generic system that can be instantiated in different possible configurations that are implemented in the factory floor and modified at runtime with plug and play.

Software and Hardware environment set up is critical in OKD-MES. A host of software and hardware options are available to set up an environment for OKD-MES. A few are mentioned in the Figure 9. IDEs such as Net Beans, Komodo, Eclipse, Cloud9, Intelli J IDEA, etc. are all compatible for OKD-MES. Server side platforms like node.js can also be used. Ontology editors such as Olingvo [36] are used to write the ontologies. Other ontology editors such as Protégé can also be used. Ontologies should consist information of the services, the device level and operation level information. UML diagrams of the system are made with standard software environments like Visual Paradigm but in this case genmymodel and draw.io are used extensively which are open source and free for public usage.

INICO S 1000's web based configuration and programming interface is extensively used for the I/Os. The web interface allows user to write and test STL code.

Virtual learning environment is best suited for OKD-MES. Virtual learning targets a large user base and is not constrained to any geographical location. Simulation can be shown in virtual learning environment. The working systems or software can be readily downloaded and installed in order to follow instructions in the tutorial videos. OKD-MES is open source. Many people ask questions like whether manufacturing applications are ready for open source software? And a more interesting question is if the software is available for free, how can the software vendors do business with it? Many surveys reveal that totally open source software have smaller amount of market share than others. But the trend is growing. Even if the company is using closed source or partially closed source software, it can still reap the benefit of open source [50]. Open source software should be the first choice for small and medium size manufacturers as it promotes innovation, supplier responsiveness and system accessibility and support [51]. Software like OpenERP, Oodo and OFBiz provide full scale benefit to the small business with good community supports and updates [50].

A popular misconception is all open source software are free. But in reality, it is not so. Most of them are free but for some there is a licensing fee involved and if it is paid, one can see the source code or how the software is written. Sometimes the licensing fee can be as small as €35. This is a great tool for developers for learning and building similar system.

The techniques used in training the concept of OKD-MES are myriad. Brainstorming is a powerful pedagogical technique that is used for communicating the concepts of OKD-MES. This is done almost every day on some aspect of the project. Though it is not possible to capture all the brainstorming sessions, structured brainstorming documentation is possible.

### 3.4 Ensuring Transfer of training/skill

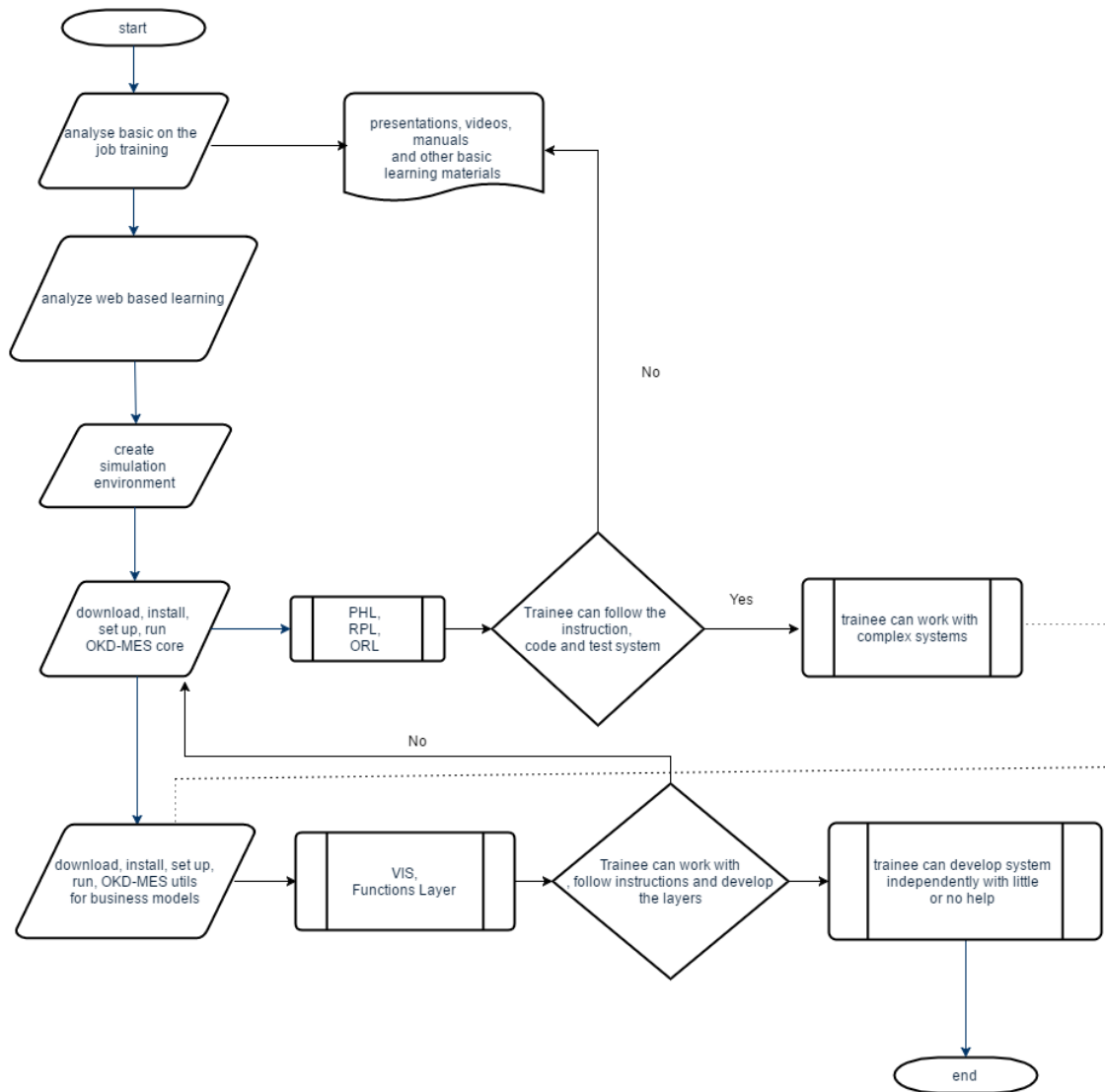
Transfer of training is defined as effective application of the training by employees in their jobs either immediately or by passage of time. In either cases the Key Performance Indicators (KPIs) are the output of the employee and the contribution to the metrics previously started.

To establish a system for ensuring and improving transfer of training, a quality check should be implemented. Transfer of training requires consistency. When participants of a training return to their work, their new skill feel forced. Dennis. E. Coates in [49] rightly mentions that Training is an ongoing process. The strive to make industrial operations more flexible and modular will not end. Engineers and developer who wish to learn and work in this area need a continuous appraisal of knowledge. The KPI is some measurable improvement in performance. In the context of OKD-MES, it is a question of “transfer of skills” rather than “transfer of training”. With the emergence of technology, the nature of skill set required in the factory floor is changing rapidly. This is an age of digital-mechanical engineers in the factory floor who are already familiar with machines and software. So, what they need are guidelines and not only instructions or manuals. It is also crucial to communicate the business results when talking about transfer of training/skill. A through research pre-and post-deployment of the system can result in identification of many areas where skill set improvement can be done.

Any task or assignment to be performed is repetition of behavior. Cognitive neuroscience explains that training inspires humans to adopt new sets of repetitive behavior. Behavior modification of humans is a slow process. Computers existed in manufacturing since 1970s yet now some people have to be convinced about the utility of software in manufacturing. The process from adoption to diffusion is extremely slow.

Feedback is a good tool to measure the training process to know the worries or concerns of the trainee. Anonymous feedbacks can help improve the overall skill transfer process. If the training is given to a multidisciplinary audience, a through guideline should be prepared which contains detailed information regarding where to find the software, how to install and configure it, how to work with it and what to do if an obstacle is faced.

A Flowchart can be drawn to summarize the process:



**Figure 11:** Flowchart to ensure transfer of training/skill

To establish KPIs for OD-MES to measure transfer of training/skill is a challenging task. To understand how much the actor has understood the process or development of the system is difficult to analyze without collecting actual feedback from them. A simple process based on the post-training activity of the worker can be used to judge the training process and measure goals. There can be 2 different forms of training for current OKD-MES system. The basic “on the job” training activity is reading the documentation of OKD-MES available online and going through the videos of OKD-MES to learn the overall working process. The actors of this process of training are Business owners and managers. The second option is going through the tutorials, downloading the tools available online and developing one’s own OKD-MES. The actors of this option would be developers, technologists and students.

Web based learning guide will help the trainee download tools like FASTory Simulator, OLS simulator and Semantic Workbench. Information and guidelines are provided to work with those tools. Beginners can use the tutorials to start from scratch.



As the OKD-MES is ramified into layers, the transfer of skill can be analyzed layer-wise. If the trainee can download, install or clone, configuring and running the system he is considered a beginner. If he can build his own OKD-MES core, i.e. PHL, RPL and ORL then he can be considered as someone who has the capacity to work with or make complex systems. If the trainee can handle even more complex tasks such as working with or developing the OKD-MES Utils, he is considered as someone who can develop and work with complex MES systems with little or no help.

Feedback from the trainee plays an extremely important role in development and updating of any training process. A method of capturing the feedback from the trainee is developed in the eScop Training Center website where an online form can collect the concerns of the trainee.

### 3.5 Developing a training Evaluation Plan

A post training evaluation plan helps to measure the effectivity of training. But it is better to monitor or plan the training in advance. Keeping in mind that if training is provided and there is not sufficient increase in knowledge or skill level, the purpose is not served. Hence it is advisable to do a quality check on the training before delivering it.

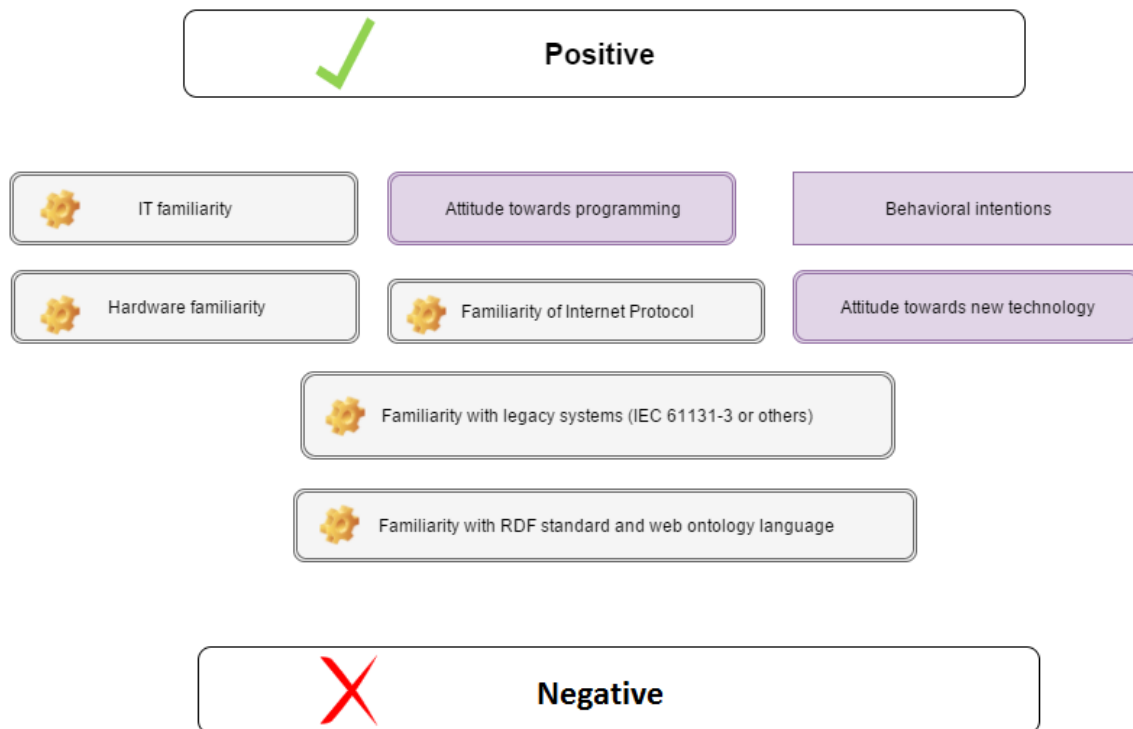
Clear goals help the evaluation plan. During training the concerns of the trainee should be addressed. Feedback and contact information of the concerned person should be provided if the trainee wants to request more information about the topic. A questionnaire can be developed to know from the trainee what he found helpful and what would he like to improve.

**Selecting a Training Method:** To start with training or skill development for people who will work with OKD-MES, what sort of training to be provided should be evaluated. To select a training method one has to understand the basic knowledge level of the trainee. The training session is divided into 3 parts, pre-training learning, the main training and post-training learning.

#### *The pre-Training*

In the pre-training phase Technology Acceptance Model (TAM) can be used. In the case of OKD-MES, a technology model can be designed for its adoption. Figure 7 shows the TAM which can be made for OKD-MES. On the left, there are the factors which influence the adoption process the most. The factors are as follows:

In the pre-training learning phase, a simple block diagram is made to find out what is the trainee's fundamental understanding and what he needs to learn and understand:



*Figure 12: Pre-Training know-how*

The field marked with a gear sign are expected from the trainee or a person undergoing training on OKD-MES. If the person is not familiar with them, it is advised to get familiar with those concepts before taking any course on OKD-MES. The fields with gear sign can be learnt. If the trainee possesses a prior knowledge of that field, he can mark it “Positive” if not he can mark it “Negative”. It is possible to rank the fields and find out a score before starting the training session but if 50% of the fields are positive, it can be considered a good starting point.

The fields marked with violet are something that the trainee should have. If any of the 3 fields are Negative for the trainee, an analysis of his objective for learning the technology has to be done.

There has been some criticism of the TAM and people have argued that the TAM might not always work because of the human factor involved. But this can be a base to start from and make suitable materials for technology transfer.

### *The main Training*

The learning medium for the main training is virtual. Training materials are developed in all forms. The following learning mediums are used for this purpose:

1. Documents about the process and architecture
2. Power point presentations explaining the concepts of OKD-MES
3. Videos describing how to work with the system
4. GitHub web URL for open source software
5. Hyperlinks to other webpages such as the main project website
6. Simulation tools
7. Ontology editor

### *The post-Training*

Post-Training analysis is important to form clusters or groups and monitor the adoption process in each group. The ideal way to identify successful adoption is when the employee is able to perform the task which brings forth improved business results. As monitoring of post training was not possible, feedback channel was created for people undertaking the training to share their experience and report any difficulties.

## **3.6 Summary**

There are many ways to capture training needs, develop training content and evaluate training. For OKD-MES, an Action Mapping and Activity Design is followed. This methodology section identifies a process of learning OKD-MES as well as provides an outline to identify which actor needs what kind of training.

## 4. IMPLEMENTATION

The layer-by-layer adoption methodology is applied to all the layers of the OKD-MES. As described in Figure 10, the process is cyclic. First instructions, guidelines and manuals are made for the user to understand the process. Then a use case approach is followed to give the user an idea about communication within the core. Ontologies are written and a mock RPL is developed to mockup the real RPL's working. Similarly, at the PHL mock devices with device ID are created. Guideline are made for working with the actual system. As stated in Figure 10 in the layer-by-layer methodology, 2 levels of outcomes are expected from the system. 1. Trainee can work with complex systems; 2. Trainee can develop his own OKD-MES with little or no help. Both the points are considered as achievements.

### 4.1 Layer-by-Layer Training

#### 4.1.1 Physical Layer:

At the PHL, guidelines for Installation and working with a technology provide a hands-on training and realistic simulation is used to demonstrate the working system. Some approaches like a training manual and User Guide can provide all necessary information needed. In the eScop project, working of the layers follows a pattern of download, install, configure and running. In the Physical Layer a training flowchart looks like:



*Figure 13: PHL installation guide*

Each step involves a series of steps in it. Installation of node.js can be learned from various sources. The installation process is also dependent on the platform used. Guidelines are outlined for different platforms. Hyperlinks are provided for the website from where the current version of node.js can be downloaded and installed. This is one of the simplest forms of training where the user can visit the link and learn about other versions of the software with no additional support provided. The required version of node.js is mentioned and if any specific requirements are there they are addressed. A step by step guide is effective for the convenience of the user to follow the instructions to setup and configure the devices. The following documents will facilitate the understanding of the OKD-MES solution:

- i. User guide: This contains instructions like a “how to do” guide. This is a big document containing all information about:
  1. Configuration of the RTUs and exposing them via web services
  2. Where to find the achieved file in the repository, how to unpack and run it with node in Linux based system
  3. How to create a web services with STL for the RTUs
  4. The code required
  5. How to use a web based IDE for configuration of the RTUs
  6. Setting up an environment for the STL
  7. Pragmas and their attributes

But before going through this guide, user must be made aware of the terms and the context in which they are used. Following this document will be time consuming and requires some additional knowledge from the user’s side. On the hand, the content of this document is extensive and this user guide is extremely useful when configuring a new device.
  
- ii. Illustration of OKD-MES devices: Devices in OKD-MES are programmable hardware. The RTUs can be generic devices which are customized by software written locally or they can be commercial devices which has a standard way of working. Commercial RTUs are Raspberry pi, Single board computers (like ALIX), Inico S1000 as a remote I/O units. The motivation to learn the devices are many. The trainee, developer or user of the device needs to learn how to host events and services. It is also important to know the method of communication of the devices with other layers of the OKD-MES. The RTUs reduce the time and cost of operation which makes the system more productive. From the end user’s perspective, this is a very interesting and important factor to understand to exploit the devices to the fullest.
  
- iii. Illustration on resources at the Physical layer: The RTU can be considered as a main class and it contains resources in the form of subclasses. The RTUs at the Physical Layer provides following resources:
  1. data – The data represents the raw data from a device. In the pilot cases of the project the data is specified with an id and url. The user can use a web interface to read or write data. This data is of interest to the user as this is a raw and synchronous value which should be available at any time.
  2. notifications – This is the implementation of the publish/subscribe mechanism which happens at the device. A notification is send to the client when an event or data exchange takes place in the system.
  3. events – An event is an indication that an action has happened. This can be material added to pallet, tool change by robot arm etc. In an event driven system, whenever an event takes place a call back is issued to notify the system that the event has taken place. This event notification mechanism is demonstrated with a video.

4. services – These are functionalities of device that can be invoked remotely or locally. A service can trigger many events when it is executed.

The method of information delivery is with presentations and videos. The materials are prepared with the help of a simulator developed for the pilot in the project. The presentation lists all the important points that the trainee must understand to exploit the resources. The videos demonstrate how the system functions. A visual aid is an effective way to learn the functions of software. The following training videos were prepared in the project for the Physical Layer:

Event Notification: In an event driven system, notification is a way to understand that the event has taken place. Destination url is the location where the notification is send when the event takes place. With this video the user can learn about the communication between layers, event driven nature of the system and how to test the system manually. The following points are described in the video:

1. downloading and installation of the simulator tool
2. unpacking and running it
3. adding order
4. using a REST client to invoke events
5. using destination url as event body
6. receiving notification in the RTU when the event has been performed

Order Entry and product tracking: In this video training material, order entry and product tracking concept is demonstrated with FASTory simulator. It is a very easy way to learn the working. The product can be tracked with palletID. Manually this procedure can be tested with a REST client by using HTTP POST method to move the pallet from one workstation to another. If the pallet is present in the work station, the system will reply with a palletID and “-1” if there are no pallets. This video gives an idea to the user how the communication between the PHL and the other layers take place. It also shows how can someone run the OKD-MES from the terminal and navigate to the device catalogue with a web browser by visiting “/RPL/RTU” and track the product with “/tracking/Products”.

Oil lubrication System simulator is another pilot case for the OKD-MES for process manufacturing industries. The video training material demonstrates how to create an OLS simulator tailor made by user defined numbers of Measuring stations, Lubrication Units, and flow meters. The video also demonstrates how to add or remove measuring stations and how to step forward the simulation by minutes, hours or days. The purpose of this material is to give the user a simulated experience of creating his/her own oil lubrication system which is built upon OKD-MES technology.

In the Physical Layer a concept of “Plug-and-produce” is used. The “Plug-and-Produce” concept is the nearest analogy of the “Plug-and-Play” concept in the computer world [27]. The device can be added or removed from the system. The device description and discovery are main challenges in “Plug-and-produce” concept. There are many ways of doing Device description. Swagger is one of the most popular ways of describing the RESTful APIs which are written for the services. The capabilities of the device can be described fully with the help of this tool. It has an interactive interface where the user can visit and try out the specs to learn the APIs. For example, the data, notifications, events and services for a device can be described with the help of swagger.

Device discovery is another very important challenge in the Physical Layer. The popular methods are hello/bye, Probing and heartbeat. Hello/bye method is followed in this case. OASIS Web Service Dynamic Discovery (WS-Discovery) version 1.1 describes the procedure of Hello/bye. The “hello” message is sent by a target service when it wants to join a network. The device is the target service and it connects to the Representation Layer (RPL) of the OKD-MES. The RPL contains modules for Ontology Services. One of such modules called Device registration module is used for registering or removing devices from the system according to the request. The device sends a hello message and the Device catalogue adds the device. When the device sends a bye message, the device catalogue removes the device. This process complements the plug-and-produce concept.

#### **4.1.2 Representation Layer**

The next layer for the layer-by-layer methodology is RPL. The approach of guidelines and installation manuals and realistic simulation with videos are followed in this layer.

The Representation Layer serves as a knowledge repository for the OKD-MES. The development of the RPL is in two main stages,

- i. The manufacturing systems ontology (MSO) – The MSO is instantiated as a “.owl” file. Web ontology language (OWL) is used to write the MSO which contains knowledge about equipment, processes and their representation in context of the manufacturing system. This is the heart of the OKD-MES.
- ii. The Ontology services – The knowledge repository can be queried or updated. For that RPL services are developed which will make SPARQL queries and updates to the MSO. A device catalogue handles critical tasks such as discovery of devices, searching for events and services etc.

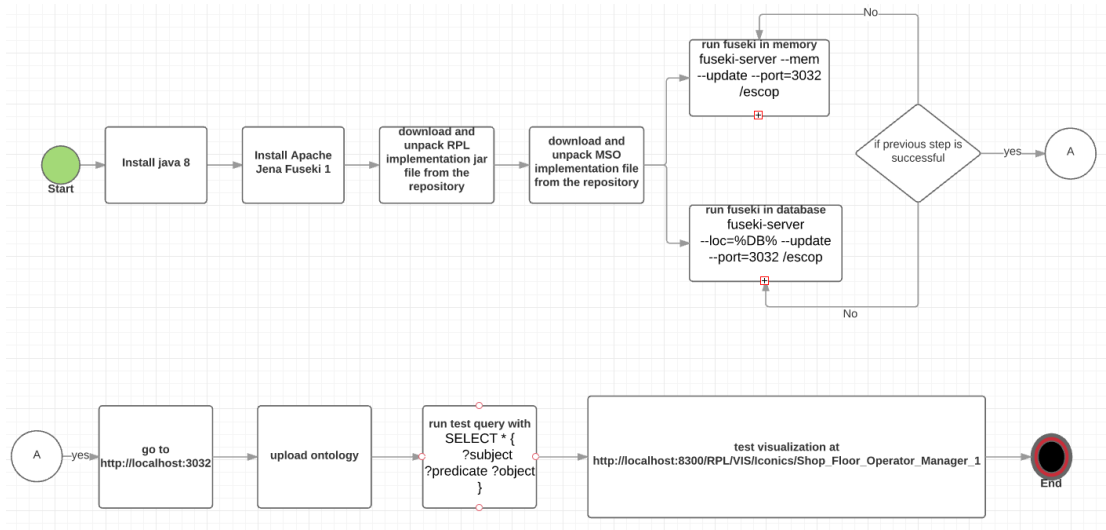
Different aspect of the RPL will be interesting to different actors of the OKD-MES system. The guidelines made for RPL are from two different perspectives. One perspective is from a developer's point of view where guidelines are provided to develop a simple RPL using spring framework. Spring-boot compiles the entire Representation Layer into a single jar file that can be run from a local computer. The development environment used is Java 1.8 Maven with Spring Framework 4.0. A concept called dependency injection can be used where the dependencies of Spring Framework can be injected to the java project with the help of the Project File "pom.xml". Dependency Injection can be defined as a concept of providing service to the class which needs them. The service class is provided to the other class which depends on it to perform a task [28]. The RPL uses this concept of dependency injection. A small snippet of code from the use case shows how the dependencies are added:

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>4.1.6.RELEASE</version>
    <type>jar</type>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-autoconfigure</artifactId>
    <version>1.2.4.RELEASE</version>
    <type>jar</type>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
```

***Program sample 1: Dependency Injection sample for OKD-MES***

The other perspective of the guideline is from a user's point of view where he needs to know how to work with the RPL (.jar file) and the ontology (.owl) file. A RPL installation guideline is produced which enlists the steps to be followed to install and run RPL in a local computer. This guide helps the user to set up the RPL and run it, upload ontology in fuseki and query and update it. It also informs the user how to use Apache's Jena Fuseki as a triple store for the ontologies. The following flowchart explains this procedure:





**Figure 14: RPL installation guide**

This guide teaches the user how to work with the Representation Layer of the OKD-MES. The prerequisites for this guide are Java 8 (tested on version 1.8.0\_51-b16) and Apache Jena Fuseki 1 (tested on version 1.1.3). The RPL can be obtained as a jar file from the central repository. The MSO implementation is available as an owl file from the central repository. Downloading and unpacking the jena-fuseki 1.1.3 is done. Fuseki server can run from the memory or database. Running the fuseki-server jar file with the command:

Running fuseki from memory:

```
fuseki-server --mem --update --port=3032 /escop
```

Where; --mem: running fuseki in memory

--update: this enables SPARQL updates

--port: specified port for running the server

/escop: dataset name

Running fuseki from database:

```
fuseki-server --loc=%DB% --update --port=3032 /escop
```

Where; --loc=%DB% : location of the database of the persistent ontology

When the server runs, a web browser is used to visit the <http://localhost:3032>. The server runs at port 3032. The MSO file is then uploaded to the server. After the MSO

file is uploaded, it can be queried or updated. To test the result a visualization file was used.

The information delivery is done both in the form of text and videos. The training material describes the procedure of installation of the OKD-MES RPL. In the final OKD-MES solution where all the layers are packaged, one may not have to perform these steps manually. But this provides an inside view to the working of the RPL.

The RPL prototype video describes a method of running the RPL, adding new device as an RTU, adding process to Orchestration Layer and getting visualization configuration screen. This video gives the user an actual feeling of working with the system. It starts from downloading the necessary files from the repository and follows a flow chart methodically for the convenience of the user to understand distinctly the steps involved. Much documentation is produced just on this layer of the OKD-MES.

### **4.1.3 Orchestration Layer**

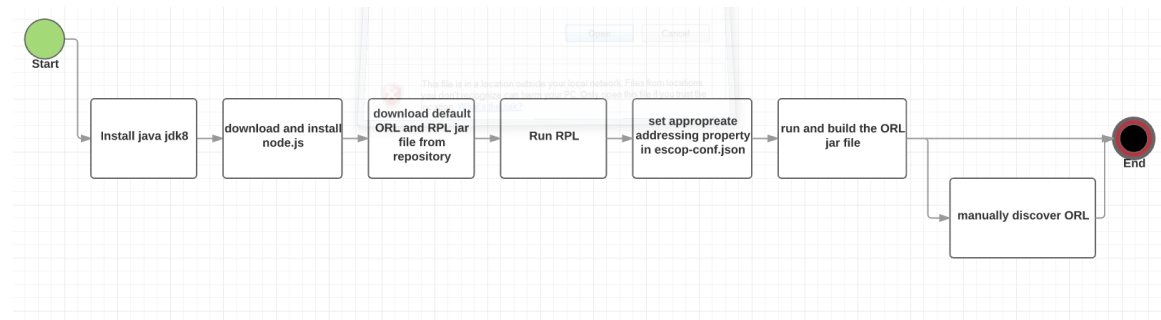
The methodology of layer-by-layer guidelines and adoption can be followed in ORL as well which is the top most layer in the OKD-MES core. Orchestration is a complex process. Orchestration can take place in the RTU level, shop floor level or Factory level. Orchestration process of RTU is a part of orchestration of shop floor and orchestration of shop floor is a part of orchestration at Factory level. Each level will have dedicated devices for orchestrating the services. It is the most complex process in OKD-MES which ensures everything happens the way it is supposed to. Actors who are touching the orchestration system should be well trained on the installation and configuration of the process, running of the Orchestration Layer and the possible trouble shooting methodologies. The OKD-MES solution does not provided ORL as a standalone application but in a packaged system. But understanding the working of the layer is crucial.

An important question here is who is this training for? Who will use the system and what exactly he/she has to learn. For working with the ORL, knowledge of programming and knowledge of how a production line should work is required. Production line functions to replicate a real-life scenario such as Order Receiving, Flow monitoring, Decision making, Resource Allocation etc. should be a part of the learning process hence, production line managers and other workers who touch the production system should be a perfect audience for these materials.

The first activity done by the ORL is Process Definition. Process definition can be done in two ways. Firstly, using json and exposing RESTful services which can be orchestrated. Secondly, process definition can be done in java as a class. In either way, Orchestration

Layer can interact with all the components of the system. Service composition depends on the process definition. These services are orchestrated in a manner which does the work of the actual physical line. The ORL orchestrates the services at the PHL. Videos of FASTory pilot case describe the communication between the ORL and the other layers of OKD-MES.

The running and configuration of the Orchestrator can be done in the following simple steps:



*Figure 15: ORL installation guide*

First a programming environment is set up. Then node.js is downloaded and installed. These steps are common to PHL, RPL and ORL. ORL can be downloaded from the repository as a .jar file and run locally. A config.json file will be downloaded with the package. It contains all the information about the instructions provided to the conveyors and robots in the factory floor. The host ip, port and the device catalogue url have to be modified in the config.json file to run the ORL in a local machine. Then using standard procedure of running a jar file through console the RPL and ORL are started.

Training materials in the form of videos and presentations are produced for ORL. The REST communication between the RPL, ORL and PHL are demonstrated with the video. The conceptualization of the process is done with a use case.

#### 4.1.4 Visualization Layer

The Visualization Layer in OKD-MES provides control and monitoring of the OKD-MES system. It handles the Human-Machine interactions. For the state of the art human-machine interaction system, the visualization should have some important features. The visualization should be easy to understand by the user and easy to work with. It is a challenge as the Visualization Layer should process complex business logics and handle concepts such as modularity and compatibility. A simple interface containing tables and drop down menus

help the user to easily familiarize with the system. The visualization screen can be accessed through a laptop, a desktop, a phone, an i-pad etc. as this is a web based technology.

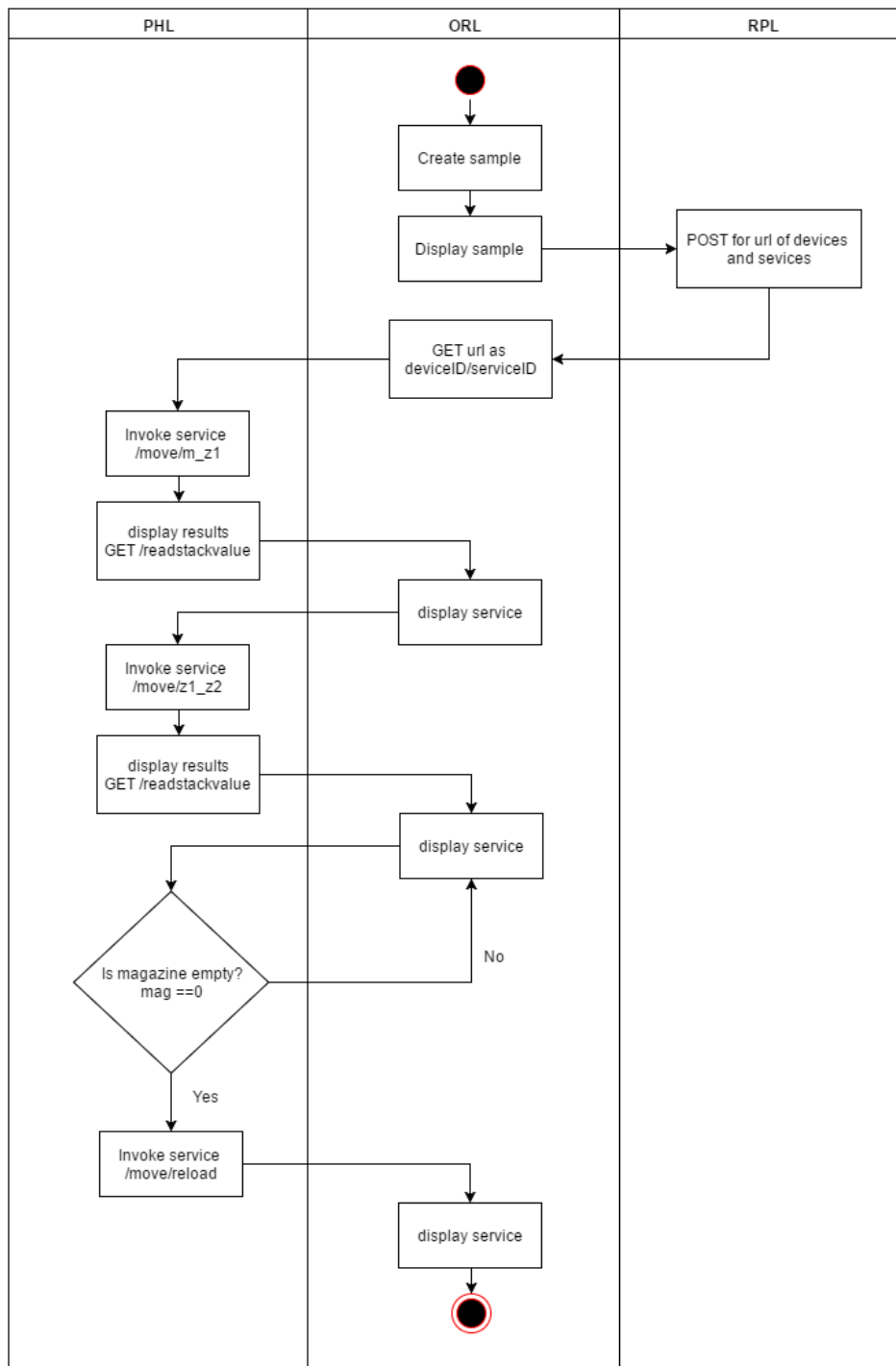
Visualization Layer of OKD-MES interacts with the RPL dynamically to update information about the system. Information on RPL and VIS interaction is important to know the present state of the system. RPL provides real time value of the system which the Visualization Layer exposes with the Visualization Provider.

The simulator is a good aid to visualize the system. The simulator is extensively used for preparing the content of training materials. The working of the system can be mocked with HTML and SVG visualization. Guidelines are produced to give the user a chance to work with the simulator.

## 4.2 Use Case

The OKD-MES has layered architecture. The core of the OKD-MES contains the Physical Layer, Representation Layer and the Orchestration Layer. A training methodology is proposed here which will guide the user to develop an OKD-MES tailored to his own needs. This proposes an abstraction of the actual working system and tries to underline the overall process of development of OKD-MES. Tutorials are made to facilitate the user's understanding. The user can write code or use the code in the tutorials and test it to have a good understanding of the interfaces and communication between layers. The idea is to give the user a starting point and a flow of information so that he feels he is gaining control over the problem. The user should understand the problem and the methodology adopted to solve it. Testing the code and getting a proper result will give the user a sense of accomplishment. There is a possibility for the user to test the solution after every layer or collectively at the end of the tutorials.

For the communication within the layers, RESTful web services are used. For the people who are familiar with the concept of RESTful web services it is easier to follow the content. But for people who are unaware of this technology should be provided additional knowledge on it. This is same for all the new technologies employed in OKD-MES. The communication between the layers can be described using an activity diagram as:



**Figure 16:** Activity diagram for use case

The process starts at the ORL level. Creating a sample here means creating “device name” and “service name” and assigning ids to the devices and services. Then at the ORL stage the sample is displayed. The following step is to ask RPL for the Uniform Resource Identifier (URIs) for the device and services that can be performed on the device. The RPL fetches the URIs from a

Knowledge base (an ontology store like the MSO) and gives back to the ORL. Now the orchestrator can invoke those services on the devices. The actual devices are placed at the Physical Layer. So, the ORL invokes the services on the PHL and displays the result. This is the basics of how OKD-MES works. Now the processes, devices and services can be built on to op it for any complicated system.

#### 4.2.1 The Physical Layer (PHL)

The PHL consists of the service embedded devices. Examples of these devices are REX control system, INICO S1000 controllers or a Raspberry Pi and Uni Pi. These devices provide deterministic real time control. The abstraction of the Physical Layer can be done with the help of FESTO Line to demonstrate the concept of OKD-MES. In FASTory, INICO S1000 controllers are used which has a rich web service interface and can be programmed with ST IEC 61131-3.

The goal is to create services which can be invoked. The first step is Service Definition and Service Description. It is convenient to demonstrate the concepts of SOA with the help of FESTO Line. Demonstration of the eScop project architecture is done with the help of a module, the Distribution Station. The Distribution Station can be divided in to 2 major devices. Stack Magazine Module which can hold up to 10 workpieces. The stack magazine module is followed by a changer module which has a swiveling arm that changes the location of the work piece.

A development environment is required. Java is selected and the tutorials are made with the help of Net Beans 8.0.2 IDE. The first step is the most crucial one. It gives a starting point and dimension the project. The Physical Layer of the OKD-MES contains RTUs (remote terminal units) with STL Manager. The main step is development of REST web services to communicate with the device and the later with the other layers of OKD-MES. Hence this conceptual framework provides access to the Physical Layer resources such as services.

Two devise are identified in the system, the “Stack Magazine Module” and “Changer Module” analogous to OKD-MES devices. Web Services are written to move workpiece from Stack magazine module to Changer Module. The Changer module moves the workpiece from the slide end and from slide-end to the transfer point of downstream station. These two positions are indicated with z1 and z2 notations. For the simplicity of programming 3 positions are defined:

Magazine – mag (m)

Slide-end – zone1 (z1)

Transfer point – zone2 (z2)

Simple services are defined in the Physical Layer as follows:

**Table 2: Service definition**

<b>Name</b>	<b>Description</b>	<b>uri</b>	<b>Method</b>
/m_z1	Move workpiece from Magazine to zone1	localhost:8081/move/m_z1	POST
/z1_z2	Move workpiece from zone1 to zone2	localhost:8081/move/z1_z2	POST
/reload	Fills up magazine with max number of workpiece	localhost:8081/move/reload	POST
/hidden	Puts a user defined number of workpiece in buffer stock	localhost:8081/move/hidden?m=n; n=number of workpiece to be added to buffer stock	PUT
/readstackvalue	Reads number of workpiece in the stack magazine at any point of time	localhost:8081/move/readstack-value	GET

A set of guidelines are laid down for writing RESTful web services to invoke the services mentioned above. The user can use the code mentioned in the guidelines or write his own code in an IDE of his choice.

#### **4.2.2 The Representation Layer (RPL)**

The Representation Layer is made in 2 parts. The first part is making the ontology and uploading it to Fuseki server and the second part is querying the server from the RPL to get the URL of services by deviceID/serviceID.

##### **A. Making the Ontology:**

To work with olingvo, first the classes are created. Two main classes are created, the class “device” and the class “service”. Then the creation of datatype and object properties is done. If it is a datatype property, the class in which it acts as an attribute is inserted in the “Domain” section. If there are limitations to the possible values that the attribute may take, that would be

stated in the “Range” section. If it is an object property, indicate what are the subject class and the object class, in the Domain and Range sections.

The list of properties written is:

**Table 3: List of Properties**

Property Type	Property Definition	Domain	Range
Datatype	URI	service	http://localhost:8080/move/m_z1 http://localhost:8080/move/z1_z2
Datatype	deviceID	device	1 (only 1 device)
Object	hasService	device	service
Datatype	serviceID	service	1 2

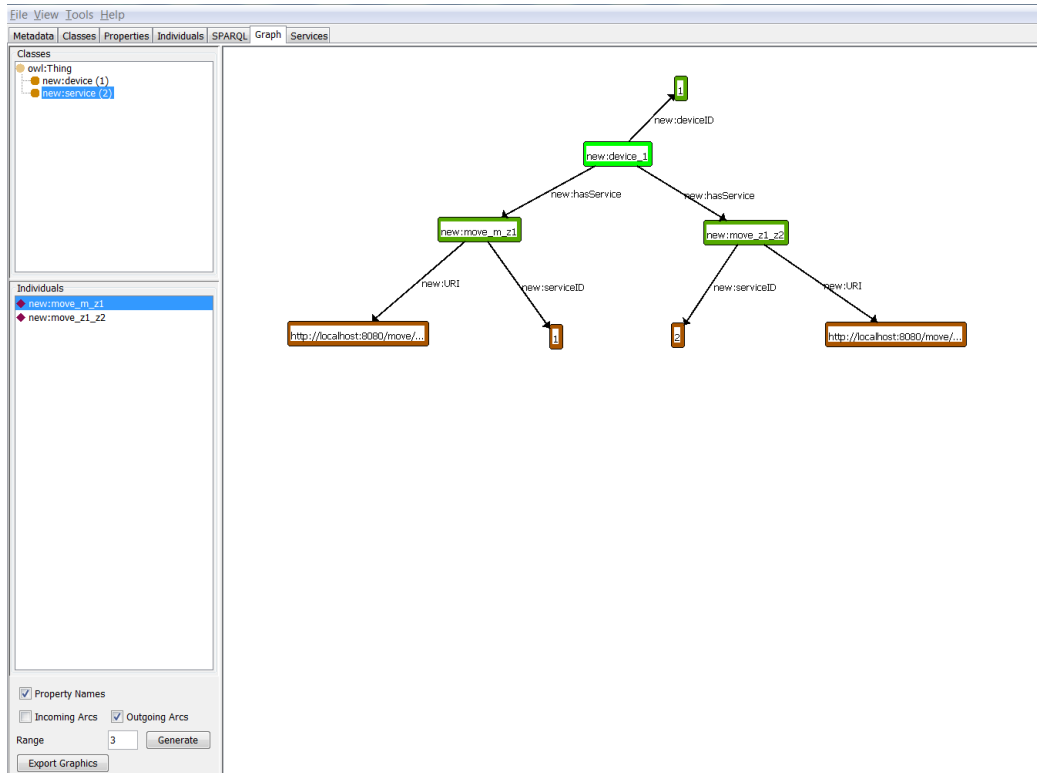
After the properties, the instances of the classes are created. The instances of the device and service classes are as follows:

**Table 4: Instances of Properties**

Classes	instances	Properties
device	device_1	deviceID hasServices
service	move_m_z1 move_z1_z2	serviceID URI



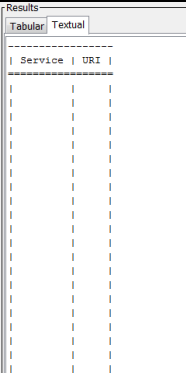
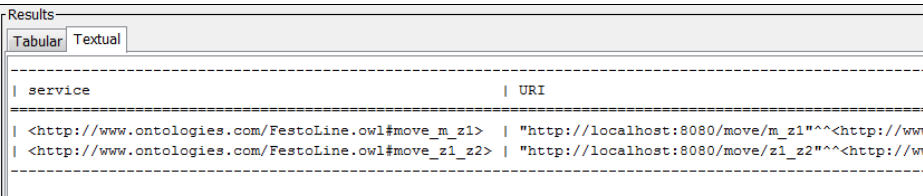
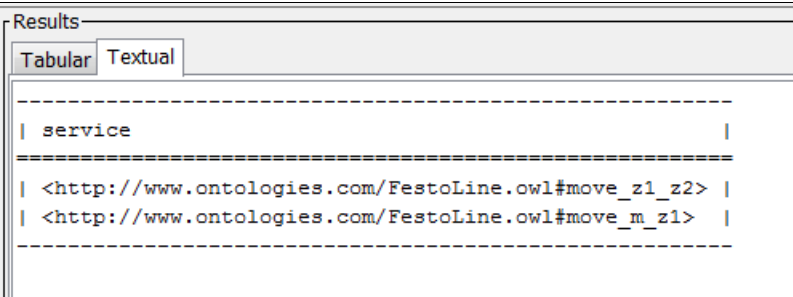
A graph of the system was generated to indicate the relationship between the classes and the individuals:



*Figure 17: Graph for classes and individuals in Olingvo Ontology Editor*

The ontology was saved by the name “festo.owl” and was uploaded in jena-fuseki 1.3.0. Queries were written in olingvo to test the festo.owl file. Some of the basic queries are as under:

Table 4: Basic queries

Query	Result						
<pre> PREFIX new: &lt;http://www.ontologies.com/FestoLine.owl#&gt; SELECT ?Service ?URI WHERE{ ?s ?p ?o. } </pre>	 <p>The screenshot shows a query result interface with a 'Results' header and two tabs: 'Tabular' and 'Textual'. The 'Tabular' tab is selected, displaying a table with two columns: 'Service' and 'URI'. The table is currently empty, with dashed lines indicating the column structure.</p>						
<pre> PREFIX new: &lt;http://www.ontologies.com/FestoLine.owl#&gt; SELECT ?service ?URI WHERE{ ?device new:has-Service ?service. ?service new:URI ?URI. } </pre>	 <p>The screenshot shows a query result interface with a 'Results' header and two tabs: 'Tabular' and 'Textual'. The 'Tabular' tab is selected, displaying a table with two columns: 'service' and 'URI'. The table contains two rows of data:</p> <table border="1" data-bbox="469 904 1396 1003"> <thead> <tr> <th>service</th> <th>URI</th> </tr> </thead> <tbody> <tr> <td>&lt;http://www.ontologies.com/FestoLine.owl#move_m_z1&gt;</td> <td>"http://localhost:8080/move/m_z1"^^&lt;http://www.ontologies.com/FestoLine.owl#move_m_z1&gt;</td> </tr> <tr> <td>&lt;http://www.ontologies.com/FestoLine.owl#move_z1_z2&gt;</td> <td>"http://localhost:8080/move/z1_z2"^^&lt;http://www.ontologies.com/FestoLine.owl#move_z1_z2&gt;</td> </tr> </tbody> </table>	service	URI	<http://www.ontologies.com/FestoLine.owl#move_m_z1>	"http://localhost:8080/move/m_z1"^^<http://www.ontologies.com/FestoLine.owl#move_m_z1>	<http://www.ontologies.com/FestoLine.owl#move_z1_z2>	"http://localhost:8080/move/z1_z2"^^<http://www.ontologies.com/FestoLine.owl#move_z1_z2>
service	URI						
<http://www.ontologies.com/FestoLine.owl#move_m_z1>	"http://localhost:8080/move/m_z1"^^<http://www.ontologies.com/FestoLine.owl#move_m_z1>						
<http://www.ontologies.com/FestoLine.owl#move_z1_z2>	"http://localhost:8080/move/z1_z2"^^<http://www.ontologies.com/FestoLine.owl#move_z1_z2>						
<pre> PREFIX new: &lt;http://www.ontologies.com/FestoLine.owl#&gt; SELECT ?service WHERE{ ?service new:URI ?URI. } </pre>	 <p>The screenshot shows a query result interface with a 'Results' header and two tabs: 'Tabular' and 'Textual'. The 'Tabular' tab is selected, displaying a table with two columns: 'service' and 'URI'. The table contains two rows of data:</p> <table border="1" data-bbox="469 1543 1203 1688"> <thead> <tr> <th>service</th> <th>URI</th> </tr> </thead> <tbody> <tr> <td>&lt;http://www.ontologies.com/FestoLine.owl#move_z1_z2&gt;</td> <td></td> </tr> <tr> <td>&lt;http://www.ontologies.com/FestoLine.owl#move_m_z1&gt;</td> <td></td> </tr> </tbody> </table>	service	URI	<http://www.ontologies.com/FestoLine.owl#move_z1_z2>		<http://www.ontologies.com/FestoLine.owl#move_m_z1>	
service	URI						
<http://www.ontologies.com/FestoLine.owl#move_z1_z2>							
<http://www.ontologies.com/FestoLine.owl#move_m_z1>							

- B. The second part is interacting with the festo.owl file with an application that can query and update the ontology file as well as POST to the Orchestration Layer. The name given to this file is rpl.jar. This is written in java 1.8 maven with spring framework 4.0. It has a model and controller architecture. In the model it accepts strings as deviceID and serviceID. It uses Request Mapper and Path Variable of Spring application to map the services as deviceID/serviceID. Queries can be written in this application to query fuseki server endpoints to provide data as url which can be posted to the ORL when asked for. A snippet of code shows how the query is written:

```

/**
 * Function to generate the SPARQL Query for getting List of Services
 *
 * @return
 */
public static String sparqlQueryRpl () {
    String query = PREFIX;
    query += "SELECT ?ser\n"
        + "WHERE{\n"
        + " ?ser a new:Service.\n"
        + " ?ser new:ID ?ser.\n"
        + "}\n";
    return query;
}

```

### ***Program 2: Writing Queries***

#### **4.2.3 The Orchestration Layer (ORL)**

The ORL process is started with defining class *ProcessDefLine(Pdl)* which contains the all the devices and the services. Then a class called *ProcessDef* is defined which contains the *ProcessDefLine* class as an array list. An *id* is provided to the *Pdl* class so that *Pdl* can be called by this *id*.

In the ORL controller class, end points are created where http methods can be employed. The orchestrator first defines a */sample* end point which consist the Process definition as a combination of device and services having fixed URIs. This sample is displayed with another end point */display*. When this sample is invoked and displayed, it provides the data about the system in a json format as shown in the figure.

```

{
  id: "sample",
  - pdl: [
    - {
      device: "StackMagazineModule",
      service: "m_z1",
      uri: "http://localhost:8081/move/m_z1"
    },
    - {
      device: "ChangerModule",
      service: "z1_z2",
      uri: "http://localhost:8081/move/z1_z2"
    },
    - {
      device: "Magazine",
      service: "reload",
      uri: "http://localhost:8081/move/reload"
    },
    - {
      device: "",
      service: "hidden",
      uri: "http://localhost:8081/move/hidden?m=10000"
    }
  ]
}

```

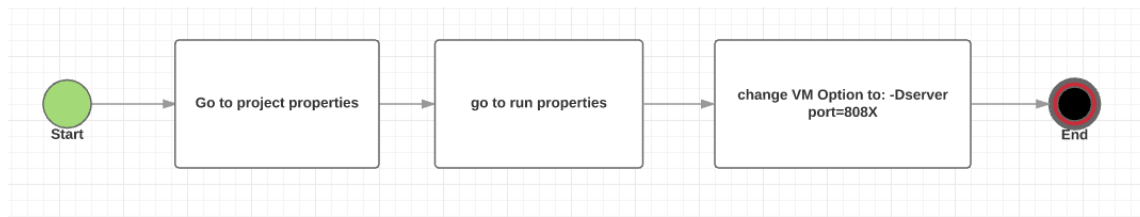
**Fig 18:** List of devices and services

The tables display the endpoints, their description and the Method used in the ORL:

**Table 5:** End point description of services

End points	Description	Request url for testing	Method
/sample	Creates a sample	localhost:8080/sample	POST
/display	Displays the sample	localhost:8080/display	GET
/RPL/RTU	Gets URIs of the services as a list from the RPL as /deviceId/serviceID	localhost:8080/RPL/RTU	GET
/service1	Invokes service move/m_z1 in the Physical Layer	localhost:8080/service1	POST
/ service2	Invokes service move/z1_z2 in the Physical Layer	localhost:8080/service2	POST

It has to be noted that when ORL, RPL and PHL run, they cannot run at the same port simultaneously. To change the port for running the package, these simple steps can be followed in all standard IDEs:



**Figure 19:** Port manipulation of ORL, RPL and PHL

*/RPL/RTU* lists the devices and the service as *deviceID/serviceID*. This is the concept of device catalogue which can also be generated from the Orchestration Layer with */display*. In the actual OKD-MES, the device catalogue is moved to the RPL.

When all the layers are running at different ports, ORL can get the device description from RPL and post to the PHL to perform services like moving workpiece from one location to another, reloading workpiece in the stack, reading value of the stack etc. From a conceptual point of view this process abstracts the OKD-MES solution. As of now, it is a training framework. This example of communication between the layers and the service description and invocation can be utilized by developers in the industry to make their own OKD-MES. A lot of variation is possible and there exists no standard way of development of the layers. It is possible to standardize this process in the future.

Training materials with all necessary instructions are produced. The code is accompanied with the explanation so that it is possible for the user to use the code directly. The user is exposed to a step by step information procedure so that he does not lose his locus of control. Successful running of the application gives the user confidence and that's why the code is tested and how to test the code is also mentioned in the tutorials.

### 4.3 Training Center

The eScop Training center is a website which can be used by internal and external users to learn about the project and get an idea about OKD-MES [40]. The website was developed because if hands-on training cannot be delivered, actors of OKD-MES can also learn individually, away from constructed training sessions. The above-mentioned presentation, videos, publications etc. are present in this website. The eScop Training center is easy to use and navigate. It presents the materials for learning in 4 steps. They are:

- (i) Understanding the overall approach
- (ii) Going through different components and tools
- (iii) Their (tools and components) illustration using project use cases
- (iv) References for additional publications and presentations on OKD-MES.

The Training center is simple to use. It targets people who want to learn about the technology from the beginning without prior knowledge of OKD-MES or techniques like simulation, terminal based working or querying a database. The tutorials are evenly and understandably distributed under headers like Approach, Components, Pilot cases etc. Additional questions and comments by users or students can be collected and addressed.

The training center can be visited at <http://www.escop-project.eu/training/>

### 4.4 Summary

Based on the Training Need Assessment flowchart in Methodology section in figure 8, Action Mapping and Activity design is done. “Action” can be described as, what to do to install, configure run and develop OKD-MES. These discrete actions when put together completes the training of the entire OKD-MES. These actions are different learning modules in layers of the OKD-MES. Activity design follows Action Mapping. Activity can be F2F training, video guidance, real life simulation etc. For every activity, a relevant action is designed. A little consideration will show that Action Mapping and Activity Design when put together form a structured training approach. Use Case is a good example of how to bring the two concepts together. The use case here defines how the Action of modelling the core of OKD-MES is done and the tutorials produced from them shows what Activity Design is followed.

## 5. RESULTS

The Adoption of OKD-MES is dependent on adoption of a host of other technologies by the industries. Adoption of web services as a standard of communication, adoption of controllers with web capabilities, adoption of Service Oriented Architecture, adoption of Ontologies in manufacturing etc. all affects the adoption of OKD-MES. OKD-MES adoption in whole and part inspires the modern industries to move towards a digital world. This thesis work proposes pedagogical approaches for adoption of specific technologies which will lead to initiation of a continuous learning process for technologies of this kind. The Concepts, tools and practices which were developed as a part of OKD-MES, provide important guidance to the user.

### 5.1 Learning Strategy

A learning strategy is developed for transfer of skill. It is demonstrated with a flowchart in Figure 10 under methodology section. In the factories, before the adoption process starts, understanding the information system and its integration with manufacturing is important. The flow of information in OKD-MES demands some amount of information science literacy. The user can understand concepts like web services, event driven system, communication etc. of the system and not treat OKD-MES as a black box to obtain desired output. People who are familiar with these technologies can use this work to have an insight for developing OKD-MES that suits their purpose.

Action Mapping technique is used to identify performance problems and where enhancement of performance might be necessary. The implementation section also provides a detail of these actions which must be learnt to understand OKD-MES. Activity Design is the next step. It is also described in the implementation section how to deliver these contents to the information seeker answering the question, “what Activity is best for which Action”.

The main purpose of the study undertaken here is to make OKD-MES concepts more understandable, accessible and ready to use. This guideline would be beneficial for beginners and intermediate level users who want to learn and work with Knowledge-driven systems.

A host of training materials and the eScop Training Center website is prepared for proper dissemination of OKD-MES. More than 1 GB of training materials were prepared but the materials which were simple, easy to understand and low difficulty level were chosen. Guidelines are prepared for OKD-MES. The guidelines describe how to self-learn, teach, practice or develop OKD-MES. Tutorials are prepared in the form of documents. These tutorials describe how to:

1. Setting up an environment for OKD-MES
2. How to write RESTful Web services
3. Obstacles faced
4. What to do to overcome those obstacles
5. Testing the system

It is important for the user to have a sense of accomplishment when s/he is trying to make her/his OKD-MES for the first time. After every major step, the obstacle faced are listed. What steps were adopted to remove those obstacles are also mentioned. Hyperlinks to other useful tutorials, software vendors' website and major online programming tutorial websites are provided. The system can be complicated to any extent based on these steps.

The tutorial was provided to departmental employees and students who were working on different projects. They were requested to take the tutorial and provide feedback on what they found was difficult and what would they like to improve. A summary of their response is mentioned in the table 6.

*Table 6: Response from trainees*

<b>Trainee</b>	<b>Difficulty to follow</b>	<b>Feedback</b>	<b>comments</b>
Trainee 1	Low	The layers are understood and it's easy to follow	In a fresh system or after updating your IDE, Maven indexes a repository saying "Unpacking Index for central repository" which may be extremely slow.
Trainee 2	Low	More explanation of the services  Explanation on port manipulation	

The feedback and the comments helped to restructure the tutorials. The main concerns of the trainees were addressed. A conclusion can be drawn that people in the industries with little or no knowledge of OKD-MES or its prerequisites will face low level of difficulty to work with the system.



## 5.2 Structure of the Training Program

A training plan is proposed in this section. The table 7 lists the transfer of skill/knowledge of the materials that are required to understand OKD-MES partly. There is no time limit for this training session. It can take a few hours to days. But it should be noted that this training will be best suited when used along with code samples. Lectures, video guides, online simulators etc. all sorts of materials can be used to conduct the training. The table also describes who is this training for. This can be for Students, researchers, industry personnel like managers, engineers and developers of any kind of MES.

*Table 7: Training Structure*

Topic	Material	Type of Material	Who are they?			
			Students	Academics	Business Owners	Developers
Introduction to OKD-MES PHL	Service Definition and Invocation, eScop Devices, Event Subscription, Event Notification, RTUs	Video, presentation, code, manual, online simulator	Automation Engineering, Information Technology, any other relevant branch of study	Researchers, Specialist, Consultants, Project Coordinators	decision makers and entrepreneurs	Who are interested to make OKD-MES or presently working with one
Introduction to OKD-MES RPL	Knowledge Representation with Ontologies, MSO, Ontology Triple Store (Fuseki jena), Olingvo (Ontology editor), query and updating Ontology	Video, presentation, code, manual, online simulator	Automation Engineering, Information Technology, any other relevant branch of study	Researchers, Specialist, Consultants, Project Coordinators	decision makers and entrepreneurs	Who are interested to make OKD-MES or presently working with one
Introduction to OKD-MES ORL	Process Definition, Service Composition, Service Definition, Service Orchestration, SOA	Video, presentation, code, manual, online simulator	Automation Engineering, Information Technology, any other relevant branch of study	Researchers, Specialist, Consultants, Project Coordinators	decision makers and entrepreneurs	Who are interested to make OKD-MES or presently working with one

Any other additional topic by Industry or Educational Institution such as:

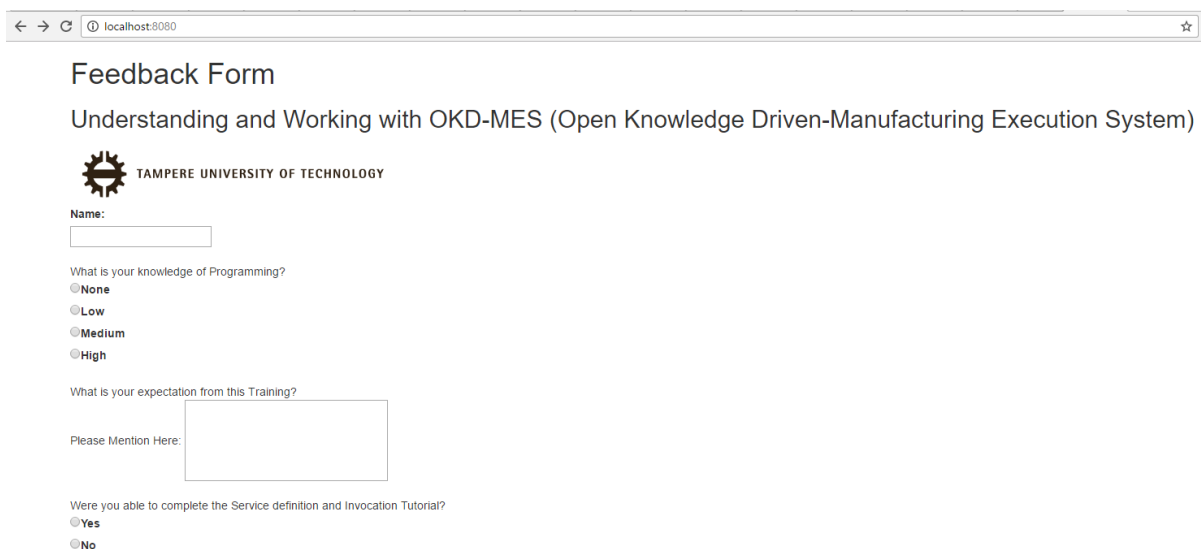
- a. Time frame of deployment of OKD-MES,
- b. Additional requirement from the company side,
- c. Investment planning if necessary,
- d. Do the company want to completely migrate to OKD-MES or try one work station or a small pilot at a time?

This structure can be tailored to fit the need of the training.

### 5.3 Feedback from trainees

After the training is conducted, feedback collection and analysis should be done. An extensive feedback form is made which contains details of the steps and difficulties faced by the trainee. There is also a chance to write any concerns and queries regarding the training process. An online feedback form is made which takes extensive information from the user to know his difficulties in solving the problem. The online form contains an extensive set of questions. This starts from collecting data about the trainee, his purpose of taking the training and his present skill level. Then the form delves into the core of OKD-MES. It collects data and reactions from the user. For example, the questions in the first page of the form shown in the Figure 19 collects information regarding the detailed difficulties the user faced in understanding the guidelines and tutorials. Here again a layer-by-layer approach is followed.


The online feedback form is made with chrome dev editor tool. The first page of the form is shown in Figure 20. This form would be provided to the trainee after taking the training. The response from the trainee can be used to improve the training process.



← → 🔄 localhost:8080 ☆

## Feedback Form

Understanding and Working with OKD-MES (Open Knowledge Driven-Manufacturing Execution System)

 TAMPERE UNIVERSITY OF TECHNOLOGY

Name:

What is your knowledge of Programming?

None

Low

Medium

High

What is your expectation from this Training?

Please Mention Here:

Were you able to complete the Service definition and Invocation Tutorial?

Yes

No

**Figure 20:** Online Feedback Form for OKD-MES

The full feedback form is available in Appendix A.

## 5.4 Pedagogical Tools

The pedagogical tools developed in OKD-MES are both Passive Instructions and Active Instructions. Passive instructions are slides and documentation about the project. Active instructions such as realistic simulation is used to give a better understanding of the working system. With Active instructions, a touch and feel experience of OKD-MES is obtained.

Brainstorming was the most common form of pedagogical technique used during the project with members of the project group. The technique can be extended to learners of OKD-MES. It is also possible to build collaboration software which includes social aspects for knowledge management. This can be useful to exchange information between trainee, developers of the OKD-MES and people working with OKD-MES. This can be called digital brainstorming. Mind maps were exclusively used in course of this work.

To disseminate the concepts of OKD-MES and to have a business focus, a story based approach is also followed. A man's experience of using OKD-MES is described in a story which starts with the problems he was facing in his factory when OKD-MES was not adopted. The change in his life and factory brought by OKD-MES is emphasized. This serves as a good marketing tool.

The strongest form of pedagogical tool applied in OKD-MES adoption is realistic simulation. The simulators developed for this purpose are available online. The FASTory Simulator can be reached at [62] and the Oil Lubrication System simulator can be reached at [63]. Instructions are prepared on how to work with the simulator. The simulator provides a visual and methodical approach of working of the system. The simulator can also serve as a platform for development of other application and customizable per one's needs. Championing this pedagogical tool is advised to understand the capabilities of OKD-MES.

## 6. DISCUSSION AND FUTURE SCOPE

The first thing which should be done at this moment is to go to the industries and conduct the training sessions. The training can also be provided to students and researchers in the academic environment so that the adoption of OKD-MES becomes wide spread. In an industry, which has already adopted SOA, Web standards, OPC UA etc. the adoption of OKD-MES is expected to be faster. A training team should be build up which can provide this training to users in any industry or academic sphere.

To analyses the effectiveness of this adoption framework, the trainee must be monitored. This framework can be considered successful if the trainee can work with the system or develop it. Rate of adoption and Transfer of technology can be used to quantify the training process. This can be used to validate this adoption framework. Online data available about the trainee and feedback provided by them should be analyzed. With that data, predictive models can be made about adoption of OKD-MES and future scope of development of it.

People tend to find comfort zones in legacy systems. Migration from those systems to OKD-MES is difficult. The impact is not only technical but psychological and social as well. An effort is made here to produce a cross disciplinary platform of learning. The purpose of this work is to facilitate the process of migration without intimidating the learner. The success rate of adoption of OKD-MES depends on increase of number of users. The process can be slow or fast, but wide spread adoption of a technology is time consuming. This thesis work provides an inside view of OKD-MES and the possibilities it can offer.

If the study is performed with different control groups, the rate of adoption can be quantified by the formula stated in section 3.2. Control groups of trainees can be formed based on location and nature of work. In a discrete manufacturing industry one control group of trainee can be formed. Another control group can be formed with the students of Factory Automation and Industrial Informatics at TUT. Rate of adoption can be studied in the various control groups. This will quantify the transfer of technology. This would give concrete results about the training process and future improvement of the methodology can be done.

Analysis of the feedback can reveal which is the most difficult part to understand for trainees. It will reveal questions and concerns of the learners and users of OKD-MES. A more adaptive framework for training can be developed from the feedbacks. Analysis of individual's weak point can be done and more materials can be provided to him/her based on his weak point.

## 7. CONCLUSION

Adoption of OKD-MES is successful if the system is used by everyone in the Industry and Academia. The open source nature of OKD-MES will help more people work with the system and adopt in their respective line of work. This thesis work identifies and demonstrates elements of OKD-MES with selected scenarios that need to be learnt and understood to integrate OKD-MES in the industry. Concepts such as reusable services, communication standards, re-configurability, scalability etc. are demonstrated.

It is believed that for adoption of any software or Information System, training is compulsory. The nature of the training could vary but the effectiveness of an adoption system is judged based on the level of understanding of people who work with it. Simulation is used as an effective pedagogical technique to provide the user with a touch and feel experience of the real system. All other supporting learning techniques are produced which will enable the learner to know basics of the system. Guidelines are provided on how to reach OKD-MES solution on the internet, configure the system, learn the layers of the system and develop those layers for one's own application.

A major challenge of this work was to define the complexity level of the adoption process. It was agreed and understood that the stakeholders of the system come from different parts of the industry and academia. It has been considered to keep the process of learning simple and easily understandable for non-technical departments as well. Feedback from the user is extremely important to continuously improve the quality of training.

OKD-MES research and development will continue in days to come. The present system will evolve and so should the process of learning. This work strives to lay a foundation in the learning process of OKD-MES based on which further advancements can be done.

## 8. REFERENCES

- [1] Eduardo Salas, Scott I. Tannenbaum, Kurt Kraiger, and Kimberly A. Smith-Jentsch, “The Science of Training and Development in Organizations: What Matters in Practice”, *Psychological Science in the Public Interest*, pspi.sagepub.com, 2012
- [2] Luminita Ciocoiu, Michael J de C Henshaw, Ella-Mae Hubbard, “A systems-of-systems approach to the development of flexible, cost-effective training environments”, 7th International Conference on System of Systems Engineering, 2012
- [3] Liu Yang, Zhigang Hu, Jun Long and Tao Guo, “5W1H-based Conceptual Modeling Framework for Domain Ontology and Its Application on STPO”, 2011 Seventh International Conference on Semantics, Knowledge and Grids, 2011 IEEE.
- [4] Sergii Iarovy, Borja Ramis, Xu Xiangbin, Anisa Sampath, Andrei Lobov, José L. Martinez Lastra, “Representation of manufacturing equipment and services for OKD-MES: from service descriptions to ontology”, IEEE 2015
- [5] Sergii Iarovy, Prof. Jose L. Martinez Lastra, Rodolfo Haber, “From artificial cognitive systems and open architectures to cognitive manufacturing systems”, IEEE 201
- [6] Borja Ramis, Luis Gonzalez, Sergii Iarovy, Andrei Lobov, José L. Martinez Lastra, Valeriy Vyatkin, William Dai, “Knowledge-based web service integration for industrial automation”, 2014 IEEE
- [7] Xiangbin Xu, Sari Räsänen, Roberto S. Camp, “State-of-Art of Manufacturing Execution Systems – A Technology Review”, *Open Knowledge-Driven Manufacturing & Logistics the ESCOP Approach*, Warsaw 2015
- [8] Aurelija Burinskiene, Vida Pipiriene, “Adoption of Information Systems by Trade and Manufacturing Enterprises”, ISSN 1822–8402, EUROPEAN INTEGRATION STUDIES 2013 No. 7
- [9] Sergii Iarovy, Borja Ramis, Xu Xiangbin, Anisa Sampath, Andrei Lobov, José L. Martinez Lastra, “Representation of manufacturing equipment and services for OKD-MES: From service descriptions to ontology”, *Industrial Informatics (INDIN)*, IEEE 13th International Conference on, 2015
- [10] Borja Ramis Ferrer, Sergii Iarovy, Andrei Lobov, José L. Martinez Lastra, “Towards processing and reasoning streams of events in knowledge-driven manufacturing execution systems”, IEEE 2015

- [11] Jiehan Zhou, Rose Dieng – Kuntz, “Manufacturing Ontology Analysis and Design: Towards Excellent Manufacturing”, 2004 IEEE
- [12] Nitishal Chungoora, Robert Young, George Gunendran, Claire Palmer, Zahid Usman, NajamAnjum, Anne-Françoise Cutting-Decelle, Jennifer Harding, Keith Case, “A model-driven ontology approach for manufacturing system interoperability and knowledge sharing”, Computers in Industries, 2013
- [13] Stanislaw Strzelczak, “Ontology aided manufacturing and logistics”, Open Knowledge Driven Manufacturing & Logistics The eScop Approach, Warsaw 2015
- [14] Ulrik Eklund, Jan Bosch, “Archetypical Approaches of Fast Software Development and Slow Embedded Projects”, 2013 39th Euromicro Conference Series on Software Engineering and Advanced Applications
- [15] FESTO 2015, “MPS The Modular Production System”, <http://www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/?fbid=aW50LmVuLjU1Ny4xNy4yMC41ODU>, Accessed on 10.10.2015
- [16] Aiman Turani, Rafael Calvo, “Sharing Synchronous Collaborative Learning Structures using IMS Learning Design”, 2006 IEEE
- [17] Luminita Ciocoiu, Michael J de C Henshaw Ella-Mae Hubbard, “A systems-of-systems approach to the development of flexible, cost-effective training environments”, 7th International Conference on System of Systems Engineering, Genoa, Italy - 16-19 July 2012
- [18] Iván Ruiz-Rube, Juan Manuel Doderó, Ricardo Colomo-Palacios, “A framework for software process deployment and evaluation”, Information and Software Technology 59 (2015) 205–221
- [19] The International Foundation for Information Technology “The Information Technology Deployment Framework”, [http://www.if4it.com/SYNTHESIZED/FRAMEWORKS/DEPLOYMENT/deployment\\_framework.html](http://www.if4it.com/SYNTHESIZED/FRAMEWORKS/DEPLOYMENT/deployment_framework.html), Accessed on 14.10.2015
- [20] Fareena Sultan and Lillian Chan, “The Adoption of New Technology: The Case of Object-Oriented Computing in Software Companies”, IEEE transactions on engineering management, vol. 47, no. 1, February 2000
- [21] G. Premkumar and Michael Potter, “Adoption of Computer Aided Software Engineering (Case) Technology: An Innovation Adoption Perspective”, Database Advances, Vol. 26(2&3), May/August pp. 105-124
- [22] Boynton E. Zmud, “An assessment of critical success factors”, MIS Quarterly, June 1985, ResearchGate

- [23] Nazmun Nahar, Vesa Savolainen, Najmul Huda, “IT Aided Training in International Technology Transfer”, Management of Engineering and Technology, 2001. PICMET '01. Portland International Conference on
- [24] J. White D.C. Schmidt, “Automating deployment planning with an aspect weaver”, IET Softw., 2009, Vol. 3, Iss. 3, pp. 167– 183
- [25] Bente Skattør, “Training and Deployment as a Basis for Usability Engineering of Mobile Systems”, 2008 IEEE
- [26] SmartFrog Open Source Website - Powered by Atlassian Confluence, Steve Loughran, <http://www.smartfrog.org/>, Accessed on 28.11.2015
- [27] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, J. Ota, “Agile Assembly System by “Plug and Produce”, Annals of the CIRP Vol. 200
- [28] Mengkeang Veng, “Dependency Injection and Mock on Software and Testing”, Master Thesis, Uppsala University, Department of Informatics and Media, June 2014.
- [29] iSpring Solutions, Inc, “E-Learning Software that really works!” <http://www.ispringsolutions.com/>, Accessed on 16.11.2015
- [30] Hall and Khan, “New Economy handbook”, National Bureau of Economic Research, NBER Working Paper No. 9730, Nov 2002
- [31] Everett M Rogers, “Diffusion of Innovation, Third Edition”, The Free Press, London, 1983
- [32] INICO Technologies Ltd, Canada, “S1000 User Manual”, <http://www.inico-tech.com/doc/S1000%20User%20Manual.pdf>, Accessed on 15.08.2015
- [33] Sergii Iarovy, Wael M. Mohammed, Andrei Lobov, Borja Ramis Ferrer, and Jose L. Martinez Lastra, “Cyber–Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems”, IEEE 2016
- [34] Sima Ajami, Tayyeb Bagheri-Tadi, “Barriers for Adopting Electronic Health Records (EHRs) by Physicians” June 2013, PubMed
- [35] Bill Poston & Joe Dury, “Semiconductor Product Life Cycle Management”, Kalypso White Paper, 2010
- [36] “A Semantically Enhanced Approach for Orchestration of Web Services in Factory Automation Systems”, Puttonen, J. 15 Aug 2014 Tampere: Tampere University of Technology. 157 p. (Tampere University of Technology. Publication; no. 1224)
- [37] Elizabeth Graham, “An Introduction to UML”, Journal of Informatics Professionals, February, 2009



- [38] Tim Verry, “What is a Raspberry Pi”, <http://www.extremetech.com/computing/124317-what-is-raspberry-pi-2>, Accessed on 04.07.2015
- [39] Leah Scully, White paper on “Building the gap between Education and Advanced Manufacturing”, Hydraulics & Pneumatics, July 2016
- [40] eScop Training Center, <http://www.escop-project.eu/training/>, Accessed on 12.01.2016
- [41] Jens Lauterbach, Benjamin Mueller, “Adopt, Adapt, Enact or Use? – A Framework and Methodology for Extracting and Integrating Conceptual Mechanisms of IT Adoption and Use”
- [42] Devadoss, P., Pan, S.: Enterprise systems use: towards a structural analysis of enterprise systems induced organizational transformation. *Commun. Assoc. Inf. Syst.* 19, 351–385 (2007).
- [43] Soh, C., Markus, M.L.: How IT creates business value: a process theory synthesis. *Proceedings of the 13th International Conference on Information Systems*, Paper 4. pp. 29–41., Amsterdam, Netherlands (1995)
- [44] Jens Lauterbach, Felix Kahrau, Alexander Maedche, Benjamin Mueller, “Re-Conceptualizing enterprise systems”
- [45] Richard W. Olshavsky, “Time and Rate of Adoption of Innovation” <http://jcr.oxfordjournals.org/content/6/4/425>, Oxford University Press, Accessed on 05.02.2016
- [46] Dawn-Marie Turner, “Are your employees ready for change” <http://thinktransition.com/are-your-employees-ready-for-change/>, Accessed on 22.06.2016
- [47] BBC Active, “Why you should use a Virtual Learning environment”, <http://www.bbcactive.com/BBCActiveIdeasandResources/WhyyoushoulduseaVirtualLearningEnvironment.aspx>, Accessed on 05.07.2016
- [48] Cathy Moore, “Is training really the answer, ask the flowchart”, Available at <http://blog.cathy-moore.com/2013/05/is-training-really-the-answer-ask-the-flowchart/>, Accessed on 12.05.2016
- [49] Dennis. E. Coates, “Enhance Transfer of Training”, American Society for Training and Development, 2010
- [50] Paul Ruben, “How to run your small businesses with free open source software” <http://www.cio.com/article/2380921/open-source-tools/how-to-run-your-small-business-with-free-open-source-software.html>, Accessed on 18.07.2016

- [51] Loek Essers, “Open source should come first when choosing new enterprise IT - Report says” <http://www.cio.com/article/2392230/open-source-tools/open-source-should-come-first-when-choosing-new-enterprise-it--report-says.html>, Accessed on 18.07.2016
- [52] Mohammad Chuttur, “Overview of the Technology Acceptance Model: Origins, Developments and Future Directions”, Sprouts: Working Paper on Information Systems, 2009
- [53] Paul Legrisa, John Inghamb, Pierre Collerettec, “Why do people use information technology? A critical review of the technology acceptance model”, *Information & Management* 40 (2003) 191–204, 2002 Elsevier Science
- [54] John Ham, “OACG- A framework for developing Open Control” IEEE 1998
- [55] Hall B, Khan B, “Adoption of New Technology”, The National Bureau of Economic Research, 2016
- [56] Yogesh K. Dwivedi, Michael R. Wade, Scott L. Schneberger, “Information System Theory-Explaining and Predicting our digital society”, Volume 28 of the series Integrated Series in Information Systems pp 441-469, Springer Aug 2011.
- [57] Marjan Laal, “Impact of Technology on lifelong learning”, *Procedia - Social and Behavioral Sciences* 28 (2011) 439 – 443
- [58] Mike Sharples, “The design of personal mobile technologies for lifelong learning”, *Computers & Education* 34 (2000) 177-193
- [59] Chee-Kit Looi, Lung-Hsiang Wong, Hyo-Jeong So, Peter Seow, Yancy Toh, Wenli Chen, Baohui Zhang, Cathie Norris, Elliot Soloway, “Anatomy of a mobilized lesson: Learning my way”, *Computers & Education* 53 (2009) 1120–1132
- [60] Dapeng Tan, Libin Zhang, Qinglin Ai, “An embedded self-adapting network service framework for networked manufacturing system” *Journal of Intelligent Manufacturing*, Springer 2016
- [61] D. Mavrikios, N. Papakostas, D. Mourtzis, G. Chryssolouris, “On industrial learning and training for the factories of the future: a conceptual, cognitive and technology framework”, Springer 2011
- [62] “FASTory Simulator”, <http://escop.rd.tut.fi:3000/>, Accessed on 28.02.2016
- [63] “Oil Lubrication System simulator”, <http://escop.rd.tut.fi:5012/>, Accessed on 28.02.2016
- [64] Robert Bacal, “Passive and Active learning in Lectures and Presentations”, <http://thetrain-ingworld.com/faq/passive.htm>, Accessed on 28.12.2016

## 9. APPENDIX A

### 1. Feedback Form Page 2

---

# Feedback Form

## Page 2



TAMPERE UNIVERSITY OF TECHNOLOGY

Where you able to place order with the simulator?

Yes

No

Where you able to invoke events and services with the simulator?

Yes

No

Please indicate the complexity level you felt while working with the simulator

Less than 20%

Less than 50%

Less than 75%

Easily Understandable

Did the result of the simulation match your expectation

Yes

No

If No, what went wrong?