Business Intelligence Application for CAD/PDM Solutions

Jasmin Laroche
Abstract

Business Intelligence Application for CAD/PDM Solutions

Jasmin Laroche and Jatin Ashra

Large enterprises have diverse applications for different needs which is utilized over different number of applications. These applications cannot communicate with each other frequently as it leads to slower productivity. The information needs to be exchanged in the organization for their seamless working so it is a major factor to consider this need of utilizing the information among different applications to run the process efficiently. In addition to this, there is also a need to represent this data that is generated through this web service into a web application so that the data can be clearly visualize in graphical format. Therefore, this research focuses on building a web service for an external application usage and representing that data into visualization format. It also involves using the Scania IT guidelines and the reference architecture for this implementation of web service.
Acknowledgement

My sincere thanks to my two thesis supervisors Magnus Persson and Madhu Cheluvaraju at Scania for their patience in supporting continuously and guiding me through this project.

I would like to express my sincere gratitude to my supervisor Lars Oestreicher at Uppsala University for his constant support in reviewing my master thesis and offering valuable suggestions and indispensable recommendations.

I would like to thank my thesis partner Jatin Ashra who has equally contributed for this thesis project.

Special thanks to Rodrigo Cid, Camilla Arnersten, David Ahlbeck for their support during the thesis. Also thanks to all the employees at the IWJP department.

Sincere thanks to my parents and brother for their love, endless support and encouragement throughout the thesis work. Finally, special thanks to my friends who have always been supportive and loving.
### Terminology Notation

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Mark-up Language</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
</tr>
<tr>
<td>CATIA</td>
<td>Computer Aided Three-Dimensional Interactive Application</td>
</tr>
<tr>
<td>ENOVIA</td>
<td>Enterprise inNOvation VIA</td>
</tr>
<tr>
<td>DELMIA</td>
<td>Digital Enterprise Lean Manufacturing Interactive Applications</td>
</tr>
<tr>
<td>VPM</td>
<td>Virtual Product Management</td>
</tr>
<tr>
<td>LCA</td>
<td>Life Cycle Applications</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer-Aided Manufacturing</td>
</tr>
<tr>
<td>CAE</td>
<td>Computer-Aided Engineering</td>
</tr>
<tr>
<td>PDM</td>
<td>Product Data Management</td>
</tr>
<tr>
<td>PLM</td>
<td>Product Lifecycle Management</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>CAA</td>
<td>Component Application Architecture</td>
</tr>
<tr>
<td>P&amp;O</td>
<td>People, Organization and Security</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>CAx</td>
<td>Computer-Aided (Design/ Manufacturing/ Engineering)</td>
</tr>
<tr>
<td>IID</td>
<td>Incremental and Iterative Development</td>
</tr>
<tr>
<td>SDLC</td>
<td>Software Development Life Cycle</td>
</tr>
</tbody>
</table>
## CONTENTS

1. Introduction .......................................................................................................................... 7  
   1.1 Background ....................................................................................................................... 7  
   1.2 Problem Motivation ......................................................................................................... 8  
   1.3 Related Work .................................................................................................................... 10  
   1.4 Research Question ......................................................................................................... 11  
   1.5 Limitations of this research ............................................................................................. 12  
   1.6 Tool Kit ............................................................................................................................ 12  
   1.7 Current architecture ....................................................................................................... 16  
   1.8 To-be architecture ......................................................................................................... 16  
   1.9 About Scania CV AB ..................................................................................................... 17  
2. Theoretical Framework ....................................................................................................... 18  
   2.1 Service-Oriented Architecture (SOA) ............................................................................ 18  
   2.2 Defining Web Services .................................................................................................... 19  
   2.3 Web Services Model ....................................................................................................... 20  
   2.4 Components of Web Service .......................................................................................... 21  
   2.5 Web Service Framework ................................................................................................ 23  
   2.6 Overview of how Web Services work .......................................................................... 24  
   2.7 ENOVIA V5 Suite ......................................................................................................... 25  
3. Research Methodology ....................................................................................................... 29  
   3.1 Overview of Design Science Method ............................................................................. 29  
   3.2 Research Strategies and methods .................................................................................. 32  
   3.3 Why Design Science ..................................................................................................... 35  
4. Implementation .................................................................................................................. 37  
   4.1 Requirement Specification .............................................................................................. 37  
   4.2 Functional Requirements ............................................................................................... 37  
   4.3 Non-Functional Requirements ...................................................................................... 38  
5. Results and Discussions .................................................................................................... 42  
6. Conclusion and Future work ............................................................................................... 44  
7. References .......................................................................................................................... 45
List of figures

Figure 1: Consumer of ENOVIA V5 - Current setup ................................................................. 9
Figure 2: Landscape setup with Web Services .............................................................................10
Figure 3: Java Maintenance Group, Java Reference Architecture ...............................................12
Figure 4: Current Architecture ..................................................................................................16
Figure 5: To be Architecture ......................................................................................................17
Figure 6: SOA and SOA Technologies ......................................................................................19
Figure 7: Web Services roles, operations and artefacts .............................................................20
Figure 8: Overview of how Web Services work .......................................................................24
Figure 9: CATIA User interface .................................................................................................25
Figure 10: ENOVIA V5 VPM Navigator ....................................................................................27
Figure 11: People and Organization entities ............................................................................28
Figure 12: Phases in the Design Science Method .....................................................................30
Figure 13: Type 1 of chart .........................................................................................................39
Figure 14: Type 2 of chart .........................................................................................................40
Figure 15: JSON output of Enovia web service .......................................................................40
Figure 16: Data visualization of Enovia web service .................................................................42
Figure 17: Output of Delmia web service ...................................................................................42
1. Introduction

In this chapter, the background and the research problem is introduced. Followed by, previously related study, goals and tools used for this research are explained. The current and to-be architecture are briefly explained and lastly, about Scania CV AB and the structure of the report is described.

1.1 Background

One of the important strategy in Product Development and Manufacturing is “Time to market”. This, along with customer's demand like high configuration, customization etc., the organizations have more demands and tight deadlines to produce a product.

Products produced should not only be of good quality, but also be produced fast enough, to succeed in an ever increasing competitive market. Hence, great importance is given to new technologies, materials, machinery, progressive methods and information tools that enable more efficient use of materials to produce products faster in order to capitalize the market demands [1]. To improve their competitiveness in the market, many companies are now implementing the PDM solutions.

PDM or Product Data Management are systems that integrate and manage all applications, information, and processes that define a product, from design to manufacture, and to end-user support [2]. In other words, PDM systems are used to control information, files, documents, and work processes required to design, build, support, distribute, and maintain products [41]. PDM systems typically manage product-related information such as geometry, engineering drawings, project plans, part files, assembly diagrams, product specifications, numerical control machine-tool programs, analysis results, correspondence, bills of material, engineering change orders, and many more [2]. PDM allows people to contribute at the early stages of product design and development. In addition to these, PDM can be seen as an integration tool connecting many different areas, which ensures that the right information is available to the right person at the right time and in the right form, throughout the enterprise [2].

Large organizations often have diverse applications for different needs, which are built with different languages, run on different infrastructure, store and exchange data in dissimilar ways. These applications, hence cannot communicate or talk to each other productively. An important
goal for such organizations is to have an ability to seamlessly exchange information between various internal business units, its customers, and partners in order to be successful, irrespective of what kind of application it has or on what kind of infrastructure they are running their applications.

Web services have evolved as a practical, cost-effective solution for uniting information distributed between critical applications over operating system, platform, and language barriers that were previously impassable [3]. They help in answering the question on how to create an homogeneous environment for information exchange, while still leveraging the core abilities of existing applications.

This research details how web services can be utilized in order to make different applications communicate and exchange information with each other. This paper also briefly details Web service, how they work and benefits of using them. This thesis work focuses on the interest of CAD/PDM maintenance team from Scania IT, which is to expose the data from ENOVIA to other applications.

Two Master Thesis students (Jasmin Laroche and Jatin Ashra) have collaborated for this thesis work. The work was equally shared between both the thesis students, starting with understanding the existing architecture, understanding the Java Reference Architecture of Scania IT, requirement gathering, analysis, design, implementation, test and deployment activities. The presentation and documentation is also co-written by the authors.

1.2 Problem Motivation

Large number of applications are used by Scania IT for their software development projects and among them are CATIA, DELMIA and ENOVIA.

CATIA V5 is the primary CAD tool integrated with ENOVIA, which is used to define and manage the manufacturing of an product. CATIA is used to manage the 3D geometries whereas ENOVIA application helps in data management such as storing and retrieving the simulated data to and from the system [24].

ENOVIA is an information management system that provides flexibility to organise information according to projects, product or business model. It helps to manage, control and improve the process and relate information. For ENOVIA application, Oracle is used as the backend database. (In this thesis work, ENOVIA and ENOVIA database are used interchangeably and refer to ENOVIA with Oracle database as backend).
DELMIA allows manufacturers to do all the planning in the production processes that includes defining and creating each process separately.

Designers at Scania use CATIA to design trucks, parts of trucks, bus etc., The output from CATIA are datafiles and the metadata (of those datafiles), which are stored in ENOVIA database. These datafiles are known as the “CAD files” which are specific to CATIA with different file formats like .CATAnalysis, .CATDrawing, .CATKnowledge, .CATMaterial, .CATPart, .CATProduct, .CATProcess etc.

Different teams within Scania IT wants to consume the information/data "Mastered" in ENOVIAdatabase as shown in the Figure 1.

![Consumers of Enovia V5 mastered data](image)

Figure 1: Consumer of ENOVIA V5 - Current setup

To access the data in ENOVIA, each application needs to create an individual integration with it. This results in a tight coupling between the application and ENOVIA.

In the current scenario there are many applications accessing ENOVIA with almost identical requests. In such case, a dedicated integration gets created for each application to ENOVIA database, resulting in redundancy, network tariff, locks on database etc.,

To address this issue, there is a need to create a simple but effective mechanism using which, data from ENOVIA can be accessible without creating any additional integrations.

The solution to this problem is to use Web Services. As shown in figure 2. Web Services, act as an interface which accepts requests from different applications (requiring similar information), retrieves information from ENOVIA and forwards it to the request application. The application
requiring similar information are routed through these Web Service with minimal or no change in the query, resulting in elimination of creating dedicated integrations between each application and ENOVIA.

1.3 Related Work

During the last few years, new requirements and challenges have emerged; systems frequently need to exchange heterogeneous data and interoperate with other applications. The associated integration of data and services is a complex and challenging issue, which depends on many factors such as system architectures, operating systems, type of the components, information to be integrated, coupling and use of the systems, performance requirements, data heterogeneity and semantics, user interfaces, middleware, and availability of resources [4].

Discovery of Web services is of an immense interest and is a fundamental area of research in ubiquitous computing. Many researchers have focused on discovering Web services through a centralized UDDI registry [5,6,7]. Also, the Enterprise integration occurs when there is a need in improving interactions among people, systems, departments, services, and companies (in terms of material flows, information flows, or control flows) [8].

ENOVIA as the Oracle database as its backend and Oracle has its own Database Web Services. Some customers prefer the separation of the Web Services framework from the database while others prefer an integrated packaging [42].
[42] discusses about the shortcomings of using the Database Web Services. Firstly, it is technically possible to handle WSDL and SOAP requests and responses over HTTP directly in the Oracle Database which works satisfactorily for simple Web services, however they will not meet the scalability requirements of enterprise Web services and SOA. Secondly Web services and SOA framework vendors are actively adding new specifications (QOS, interoperability, and so on) or improving existing specifications, the Web Services Provider and SOA stack is better handled in an agile middle-tier environment rather than a more stable RDBMS environment. Thirdly, most vendors, including Oracle, use the same Web services and SOA framework for both the middle and RDBMS tiers, thereby furnishing a consistent Web Services development or assembly, deployment, and management across the database and application server tiers.

Based on the above shortfalls and although there is a lot of work done in the area of Web services and SOA, the implementation of using Web Services with ENOVIA, DELMIA etc. is still lacking. This thesis work is intended to understand how Web services can be created and used to overcome some of the above drawbacks and help to expose the data from ENOVIA (oracle database) to other applications and how that data can be consumed by different actors within Scania IT (display required information through web-frontend).

1.4 Research Question

An important aspect in any organization is to align business users with information technologies. The current market is more directed towards providing solutions which are flexible, loosely coupled, reusable, transparent, platform independent, scalable and so on. In order to reap the benefits of such solutions, Scania IT is looking towards SOA which delivers the above characteristics and also many others.

Another important aspect is to utilize the data which is exposed by these solutions (in terms of this thesis work, we call the solutions as Web Services). The goal is to extract the (business) intelligence part from the data and display it using appropriate presentation layer for business analysis. These two points leads us to the research question,

“How can data from ENOVIA (oracle database) be exposed to other actors in Scania IT in most effective and efficient way”.

Sub question:

“Which solution should be chosen so as to make it interoperable or independent of any technologies it executes on“.
1.5 Limitations of this research

This thesis work is limited to the requirements from the CAD/PDM maintenance department. The resultant solution from this thesis work can be reused with some modifications for other (similar) requirements by taking advantage of the Web Service properties of interoperability, reusability, scalability etc.

As there are numerous definitions and explanations on what Web Services are, their composition and how they work, this thesis work will present them in its most simplest form in Section 2.

This thesis work also has limitation with respect to the use of P&O. The Web Service created will be accessible to all ENOVIA, DELMIA and CATIA users within Scania as these Web Services will be integrated with the Active Directory and no further security checks (like authorization or authentication) will be carried out for accessing these Web Services.

1.6 Tool Kit

Various tools are currently available in the market, however, for this thesis work, tools recommended by Scania’s JMG (Java Maintenance Group) are used. The following sub-sections briefly describes the tools used in the thesis work.

a. Java Reference Architecture:

Scania’s JMG recommends the following tools and frameworks for java projects with an intent to maintain commonality in the architectures of the different java projects with Scania IT. Each block in the Fig. 3 are described in the following sub-sections.
As described in the Scania IT’s internal wiki, “There are many Java projects in various stages of development of Scania IT.

At the same time, new projects are started using newer tools and architectures. In order to curb the growth of the Java toolset a reference architecture has been defined so that new projects use a similar tool set and have a similar structure. The reference architecture consists of tools that cover most aspects of a Java software development project, for example:

- Development environment
- Development tools
- Continuous delivery tools
- Static code analysis

b. IDE:

Many tools are available for software development. IDE’s help a lot during software development like source code automation, intelligent code completion and so on. A IDE is chosen depending upon the requirements and need, it should support multiple languages, highlighting syntax, In-product compilation etc.

For this thesis work, we have chosen to work with IntelliJ IDEA. IntelliJ IDEA is a Java Integrated Development Environment (IDE) for developing computer software, developed by JetBrains. IntelliJ has advantage over Eclipse where it allows to navigate between Java, JSP, JS, CSS and other type of file, however with Eclipse, plugin has to be installed in order to navigate through file types other than Java. Some of the important features of IntelliJ are coding assistance, built-in tools and integrations, plugin ecosystem and support many languages like java, groovy etc.

c. Build Tool:

Build tools help in compiling and packaging the source code. They also manage dependencies, which means by using a good build tool, there will not be any need to look for dependent JARs, as they will be pulled automatically and added to your class path [9].

Maven is an automated build system that manages the entire “build cycle” of software projects.

To use Maven in a software project, a pom.xml file is created that specifies the project structure, settings for different build steps (e.g. compile, package, test) as well as libraries that the project depends on. These libraries are automatically downloaded by maven, from specified repositories
and these repositories can be either private or public [10]. For this thesis work, Scania’s internal repositories were used.

**Maven** when compared to **Gradle**, provides dependency management, debugging, collaboration and creates powerful builds. Maven also has more componentized builds and is consistent in project structure and reduces duplicates.

d. **Java:**

There are different release of java available and depending upon the compatibility of the IDE, JDK release can be chosen. This thesis work uses the JDK1.8 (Java Development Kit) release.

e. **Spring Boot**

Spring Boot makes it easy to create stand-alone, production-grade Spring based applications that you can just run. Spring Boot is used to create Java applications that can be started using java -jar or more traditional war deployments.

One of the key feature of Spring boot is that it creates Stand-alone Spring applications. There is no need to deploy the WAR file as it has embedded Undertow, Tomcat or Jetty. It also handles Starter dependencies, automatically configures Spring whenever possible, produces “Production-ready” environment. It also does not generate any code and there is no requirement for XML configuration.

f. **Perforce:**

In software development of real projects, there is a need for version control system (version management). Github, BitBucket etc., are the common platforms for hosting software developments. In this thesis work, Perforce is used as Version control system and source code management, as it is a standard tool employed within Scania IT for software development projects.

g. **CATIA:**

CATIA has over the years developed into an software suite consisting of CAD, CAM and CAE. CAD is used for 2D and 3D design, CAM is used for manufacturing processes to be designed for 3D model manufacture and CAE allows verification through analysis of 3D models. This thesis work uses the version 5 (CATIA V5) which is most widely used version of CATIA.
h. ENOVIA:

ENOVIA is for collaborative management and global life cycle (PLM), with the historical VPM and its successor VPLM as well as DMU which came from the MatrixOne acquisition. ENOVIA provides a framework for collaboration for Company's PLM software [11].

i. DELMIA:

DELMIA is a factory simulation software which enables manufacturers in any industry to efficiently plan, manage, and optimize their global industrial operations [12].

It allows manufacturers to virtually define, plan, create, monitor and control all production processes, from the early process planning and assembly simulation to a complete definition of the production facility and equipment [11].

j. REST Services:

REST is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web [13]. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol [13]. The applications created using REST are simple, lightweight and fast.

A simple comparison between SOAP and REST is provided below. REST Web services meets the needs for this thesis work and hence we have employed REST Web Services.

k. Comparison between SOAP and REST

SOAP is a traditional approach for enterprises when creating Web Services. SOAP is Language, platform and transport independent whereas REST requires the use of HTTP for transport. SOAP works well in the distributed enterprise environments whereas REST assumes direct point-to-point communication. On the other hand, REST are easier to use and more flexible. REST is more efficient as it uses smaller message formats and faster as there is no extensive processing required. REST is also closer to other Web technologies in design philosophy.
1.7 Current Architecture

Figure 4 shows the current architecture of the system and below is the brief overview of the landscape components

**Client**, are the end users who use applications like CATIA to design trucks, their part sets etc.

**ENOVIA**, is the application which facilitates in storing the data to the ENOVIA database.

**ENOVIA DB**, is the database running on the DB server, where the information of the design parts are stored. It consists of the metadata of those CAD files.

**Vault**, is where the CAD files are stored.

![Figure 4: Current Architecture](image)

1.8 To-be Architecture

Web services make it easy for other applications to access the data. A web service is a unit of managed code that can be remotely invoked using HTTP, that is, it can be activated using HTTP requests. Web services allow you to expose the functionality of your existing code over the network. Once exposed on the network, other applications can use the functionality of the program [14]. To implement the web services, the following tools were added to the existing architecture.

**Database**, to hold the interim data

**Spring boot server**, which will host and run the web services
**Presentation Layer**, which is created by using AngularJs and jQuery framework. d3.js is used to display the data in graphical format.

**Maintenance Organization**, Maintenance team supporting the architecture.

---

1.9 **About Scania CV AB**

Scania AB was founded in 1891, and it is a major automotive industry, manufacturer of commercial vehicles specifically heavy trucks and buses with its headquarters in Södertälje, Sweden [15]. Scania IT, is a subsidiary of Scania group, which works on Information Technology and handles all the designing and developing of IT solutions, systems and products for Scania Group [16].

This research work was conducted for the CAD/PDM maintenance group within Scania IT whose core domain is implementing and maintaining the CAx/PDM solutions. Scania IT provides CAx/PDM tools and solutions to develop designs of Engines, Trucks and Buses. Dassault Systemes is one of the reputed vendors, among the many vendors for CAx/PDM tools in the market. For the development of IT solutions, Scania IT employs solutions like CATIA, ENOVIA, DELMIA and other solutions from Dassault Systemes.
2. Theoretical Framework

This chapter provides insights into the key concepts of frameworks used for the development of web services. We briefly describe SOA and web-services. The sub sections also gives a brief overview about the benefits of web-services, active directory and its integration with web-services.

2.1 Service-Oriented Architecture (SOA)

With the target to serve their customers better, be competitive and more responsive in the ever changing market, IT executives are facing a stiff challenge in how to utilize existing technology, as well as reduce the costs. In most of the well-established organizations, it is a huge task to integrate different applications or systems from different vendors and of different ages. Also, with globalization and accelerated growth of e-business, customer's requirements and needs, change more quickly thereby resulting in shortening the product cycles.

This leads to the question about how to make IT environment flexible, how to integrate heterogeneous applications seamlessly and how changes can be adapted more robustly.

Although there are many definitions for SOA, Hao defines SOA as, *The Service-OrientedArchitecture (SOA) is a software architecture that defines the use of services, to support software user requirements* [17]. The implementation of SOA application is made possible through the realization of Web Services [17].

The difference between SOA and Web Service is, SOA is the architectural style, however Web services is a technology that can be used to implement SOAs. The Web services technology consists of several published standards, the most important ones being SOAP and WSDL [18]. CORBA is another example of technology for implementing SOA. The relationship between SOA and SOA technologies is represented in Fig 6 [18],
As shown in Fig 6, SOA can be implemented via Web Services, CORBA, Jini etc. Web Services employ WSDL, SOAP, UDDI and XML, CORBA uses IDL, IIOP and other Internet technologies for implementations. In this thesis work, we use Web Services based on REST to answer the research question.

### 2.2 Defining Web Services

Web Service is a software component representing specific set of business functions that can be described, published and invoked over the Internet using XML-based open standards such as Simple Object Access Protocol, Web Service Definition Language, Universal Description, Discovery, and Integration [19].

According to W3C, the term web services can be defined as follows: “A web service is a software application identified by a Uniform Resource Identifier (URI), whose interfaces and bindings are capable of been defined, described and discovered as XML artefacts” [20]. Web-services use XML to encode all communication.

Another way of defining is - Web services are based on web applications and are of open standard like XML, HTTP, etc. which interact with other web applications to exchange information/data. They can also convert existing applications to web applications.

"Web service" describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, Simple Object Access Protocol is used to transfer the data, WSDL is used for describing the services available and UDDI lists what services are available [21].

Although there are other technologies available, the following advantages makes Web-services special,
a. Web-services use XML standards, which makes them to be platform and language independent.

b. HTTP is used by web-services for transmitting messages like a service request and response, which gives a major advantage if Internet-scale applications have to be built.

c. Web Services lend themselves naturally to build loosely coupled systems resulting in systems which are scalable more when compared to strongly coupled systems, further resulting in fewer architectural requirements on actual implementation of the Web services.

Compared with other software entities, a web service has the following features [20]:
- Can be described: A web service can be described by a service description language.
- Can be released: A web service can be registered in the registering centre and released.
- Can be found: The user can send a search request to a registering centre to find the service and the access information.
- Can be bound: The web service description information can be bound with a runnable service instance or service proxy.
- Can be called: A web service can be called by remote code with the description information.
- Can be composited: The web services can be composited together to a large granularity service.

2.3 Web Services Model

As shown in Figure 7Web Services architecture is based upon the interactions between three roles: service provider, service registry and service requestor. The interactions involve the publish, find and bind operations. Together, these roles and operations act upon the Web Services artefacts: the Web service software module and its description [22].

![Figure 7: Web Services roles, operations and artefacts](image)
2.3.1 Roles in Web Service Architecture

**Service Providers:** The owner of the service is known as an Service Provider. On the architecture level, Service Provider is the platform that hosts access to the service.

**Service Consumers:** A Service consumer or requester is the software entity which want to consume an service by calling the Service Providers. The Service Consumers can be a person or another Web Service.

**Service Registry:** Service Registry provides an platform for Service Providers and Service Consumers to communicate. Service provides publish their service descriptions in the searchable registry from where the Service Consumers can look up for the required services.

2.3.2 Operations in Web Service Architecture

**Publish:** For Service Consumer/Requestor to find service descriptions, they have to be published by the Service Providers. Services Description are published depending upon the requirements of the application.

**Find:** Find operation is used by the Service Consumer to either retrieve aservice description directly or queries the required service from the Service Registry.

**Bind:** In the bind operation, the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact and invoke the service [20].

2.3.3 Web service Artefacts

**Service:** A service is a software module deployed on network-accessible platforms provided by the service provider [20]

**Service Description:** Details of the interface and implementation of the service is contained with the Service Description like binding information, data types, network location etc. and it is published to either Service Registry or Service Consumer.

2.4 Components of Web Service

Many programming languages are often used to create different applications for different usages, and often cannot interact or communicate with each other. To enable communication between such different application, Web Services are employed. Web services uses a combination of open standards and protocols like XML, SOAP and **Web Service Definition Language** to enable such communications. XML is used to tag the data, SOAP is used to transfer the message and WSDL
Web services are powered by XML and three other core technologies: WSDL, SOAP, and UDDI. Before building a Web service, its developers create its definition in the form of a WSDL document that describes the service's location on the Web and the functionality the service provides. Information about the service may then be entered in a UDDI registry, which allows Web service consumers to search for and locate the services they need whether or not the company wants its Web services to be discovered by internal and/or external service consumers. Based on information in the UDDI registry, the Web services client developers uses instructions in the WSDL to construct SOAP messages for exchanging data with the service over HTTP.

A brief about XML, SOAP and WSDL is given in the following sub-sections,

**XML:** XML stands for Extensible Markup Language. There are three important characteristics of XML that makes it useful in a variety of systems and solutions:

- XML is extensible: XML allows you to create your own self-descriptive tags, or language, that suits your application.

- XML carries the data, does not present it: XML allows you to store the data irrespective of how it will be presented.

- XML is a public standard: XML was developed by an organization called the World Wide Web consortium.

XML is used by the Web services to code and decode the data. Being a text based language, it is human readable and also application independent. XML forms the basis for all modern Web services, which use XML-based technologies to describe their interfaces and to encode their messages [3].

**SOAP:** SOAP is an XML-based protocol from the W3C for exchanging data over HTTP. It provides a simple, standards-based method for sending XML messages between applications. Web services use SOAP to send messages between a service and its client(s). Because HTTP is supported by all Web servers and browsers, SOAP messages can be sent between applications regardless of their platform or programming language. This quality gives Web services their characteristic interoperability [3].

Data is sent between the client(s) and the Web service using request and response SOAP messages, the format for which is specified in the WSDL definition. Because the client and server adhere to the WSDL contract when creating SOAP messages, the messages are guaranteed to be compatible.
WSDL: WSDL is an XML-based format for describing Web services. Clients wishing to access a Web service can read and interpret its WSDL file to learn about the location of the service and its available operations. Through the WSDL, a Web services client learns where a service can be accessed, what operations the service performs, the communication protocols the service supports, and the correct format for sending messages to the service. The WSDL document that describes a Web service acts as a contract between Web service client and server. By adhering to this contract the service provider and consumer are able to exchange data in a standard way, regardless of the underlying platforms and applications on which they are operating [3].

UDDI: UDDI is a specification for creating an XML-based registry that lists information about businesses and the Web services they offer. UDDI provides businesses a uniform way of listing their services and discovering services offered by other organizations. UDDI often describes services using WSDL and communicates via SOAP messaging. By registering Web service in a UDDI registry, Web services can be accessed not only from inside the company network, but also from outside. By querying the UDDI registry, a developer can obtain the WSDL for the service he/she wishes to [3].

2.5 Web Service Framework

There are many frameworks for Web Services and in this thesis work, the framework REST is used for creating the Web Services that will support JSON and XML.

The Web Services created in the thesis work is based on Representational State Transfer (REST) architecture for sharing and merging information from multiple sources. REST is an architectural style of networked systems consisting of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of "representations" of "resources". A resource can be essentially any coherent and meaningful concept that is addressed. A representation of a resource is typically a document that captures the current or intended state of a resource [23].

Some of the advantages of using REST are that it makes complex things simpler. REST fully leverages protocols and standards that power the World Wide Web and is simpler than traditional SOAP-based web services. With the emergence of cloud-computing and the growing interest for web hosted applications, REST-based technologies can help both in the development of rich user interface clients calling into remote servers; and in the development of actual servers for manipulating data structures in a client application (written in any language) or directly in the browser [23].
In this thesis work, Json output is produced for the functional requirements 1, 3 and 4 and XML for the functional requirement 2.

Json stands for JavaScript Object Notation and it is a way to store information in an organized, easy-to-access manner and uses minimal data format. In order words, it gives us a human-readable collection of data that we can access in a really logical manner. One of the advantage of using Json is that it assists in loading the data quickly and asynchronously (without delaying page rendering). Json is used to transmit information/data between a server and web application. It is known to be an alternative to XML.

2.6 Overview of how Web Services work

Web services consists of 3 main components, SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration) and WSDL (Web Services Description Language) as shown in Fig 8.

In the first step the Web Service Provider registers the services it provides with the UDDI Directory using SOAP protocol. The Web Service consumer queries UDDI directory to locate the required services providers using SOAP protocol.

Once the service provider is noted, the service consumer then looks at the WSDL document published by the provider to know how its web services work. The service consumer now calls the web service which in turn returns the results back (as a response).

XML (EXtensible Markup Language) is a tag based language which is used by the Web service to describe the data. The request from the consumer and the response both are in the XML language.

![Figure 8: Overview of how Web Services work](image)
2.7 ENOVIA V5 Suite

The following section introduces the system components of ENOVIA suite. The applications in ENOVIA suite consists of ENOVIA V5 VPLM, ENOVIA V5 LCA, ENOVIA V5 VPM Navigator.

CAA - Component Application Architecture, is the core architecture on which CATIA, ENOVIA and DELMIA products are built. CATIA, ENOVIA and DELMIA are applications built on strong architecture which is made up of several thousands of components all seamlessly integrated together. CAA provides capability to integrate customer's processes and industry knowledge into Dassault Systemes products.

2.7.1 CATIA V5

CATIA is the primary CAD tool integrated with ENOVIA which is used to define, manage, and simulate the product [26]. It also has 2D and 3D rendering capabilities. CATIA addresses the product development process, from early product concept specification through to product in service [11].

CATIA aids in product engineering and manufacturing from initial concept to product in service, which is widely used in automobile and aerospace domains. To design a 3D model in CATIA, the user has to import Product Information into its Workspace. The bill of material, which provides details on all the required parts and product assembling, is obtained from a proprietary legacy system deployed in-house within Scania IT [24]. User interface of CATIA can be seen in Fig 9:

![Figure 9: CATIA User interface](image-url)
2.7.2 ENOVIA V5 VPLM

ENOVIA V5 VPLM is an information management solution helping enterprises in managing the design and fabrication of parts including the industrial space modeling and configuration of digital models. ENOVIA acts as a virtual collaborative data management tool to design, simulate, validate products designs and processes throughout the product lifecycle [25].

2.7.3 ENOVIA LCA

ENOVIA LCA is an object oriented PLM database management application used to manage Product Definition Data through the life of a product. It provides support for unified access from 3D product lifecycle information and the product, process and resource (PPR) information definition in ENOVIA V5 VPM and DELMIA V5 via a collaborative, web based environment [25].

ENOVIA LCA solutions provide support for business processes from product specification through logistics planning. Enovia LCA is implemented within a comprehensive set of four ENOVIA foundations (LifeCycle Applications, Enterprise Architecture, RADE et PPR Hub) that covers the complete requirements of a scalable information system including extensive data management, workflow support and user based life cycle applications [25].

2.7.3 ENOVIA V5 VPM Navigator

ENOVIA VPM Navigator is an object oriented PLM application very similar to ENOVIA LCA but functions within a “CATIA” type window in CATIA V5. This interface into the ENOVIA database is considered more “user friendly” than ENOVIA LCA (which can be referred to as “ENOVIA Classic”). VPM Navigator does not provide all the functionality of LCA but for designers it gives them the ability to do their job without all the management that goes into LCA [26].

VPM Navigator along with VPM Relational Design and VPM Configured Product Design are the cornerstone applications for concurrent engineering in configured product design and review. VPM Navigator allows one to access both product definition and product data stored in the ENOVIA V5 VPM with extensive capabilities for navigation, search and filtering using various methods such as queries or tree navigation and collaboration tools [24]. V5 VPM captures and manages engineering design intent by exposing specifications, manufacturing process and rules, operational parameters, simulation results, thereby accelerating the understanding of how change affects associated product components and processes [24]. ENOVIA V5 VPM Navigator example is shown in Fig 10:
2.7.4 ENOVIA VPM People, Organization and Security (P&O)

ENOVIA P&O is a set of tools and methods designed to help the ENOVIA database administrator, to determine which end users of ENOVIA Lifecycle Applications are able to work on which project, and specify which functions users will be able to use, and under which conditions. It also allows you to define security mechanisms for controlling access to ENOVIA objects manipulated using ENOVIA Lifecycle Applications. Some of the concepts regarding P&O are, Resource Management, Organization, Person, Role, Project, Context, Object, Authorization etc. It assist ENOVIA database administrator to setup and manage users in an organization. Some concepts are shown in the Fig 11.

<table>
<thead>
<tr>
<th>Person</th>
<th>A user that is supposed to be able to connect and use the ENOVIA V5 product.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org</td>
<td>Organization, each user in ENOVIA V5 need to belong to a given organization. And each organization has a parent organization except for the ADMIN organization that is the root of all ENOVIA V5 organizations.</td>
</tr>
<tr>
<td>Role</td>
<td>A role that can gather specific privileges defining what are the actions authorized for that role.</td>
</tr>
<tr>
<td>Proj</td>
<td>Project, the ENOVIA V5 view of a real project.</td>
</tr>
<tr>
<td>Level</td>
<td>A security level that can be used to define access rights to the data stored in the ENOVIA V5 database.</td>
</tr>
<tr>
<td>Mask</td>
<td>A mask that defines for each visible object what attributes should be visible, editable, what their alias and default or authorized values should be.</td>
</tr>
<tr>
<td>Ctx</td>
<td>Context, gathers a role, an organization, a project and a mask. A user has to be assigned to at least one context in order to be able to connect to ENOVIA V5. The only context existing by default in a fresh ENOVIA V5 database is the VPMADMIN.ADMIN.DEFAULT that has all privileges on all the data.</td>
</tr>
</tbody>
</table>
Figure 11: People and Organization entities
3. Research Methodology

In this section, research methodology used for this thesis work is explained. Research Methodology is the process of collecting information and data to make business decisions. It can include interviews, questionnaires, publication research and other research techniques and may include present and historical information.

3.1 Overview of Design Science Method

Design Science research involves the creation of new knowledge through design of novel or innovative artefacts and analysis of the use and/or performance of such artefacts along with reflection and abstraction. Examples of such artefacts are algorithms, human/computer interfaces, system design methodologies or languages etc. [44]

Table 1 shows the Design Science Research Methodology presented in [45], although for this thesis work we have used a different variant for theirs. Peffers et al. [45] created the design science research methodology or DSRM with three objectives in mind: “(1) provide a nominal process for the conduct of DS research, (2) build upon prior literature about DS in IS and reference disciplines, and (3) provide researchers with a mental model or template for a structure for research outputs” [46]

<table>
<thead>
<tr>
<th>DSRM activities</th>
<th>Activity description</th>
<th>Knowledge base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem identification</td>
<td>What is the problem? Define the research problem and justify the value of a solution.</td>
<td>Understand the problem's relevance and its current solutions and their weaknesses.</td>
</tr>
<tr>
<td>and motivation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define the objectives</td>
<td>How should the problem be solved? In addition to general objectives such as feasibility and performance, what are the specific criteria that a solution for the problem defined in step one should meet?</td>
<td>Knowledge of what is possible and what is feasible. Knowledge of methods, technologies, and theories that can help with defining the objectives.</td>
</tr>
<tr>
<td>of a solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design and development</td>
<td>Create an artifact that solves the problem, Create constructs, models, methods, or instantiations in which a research contribution is embedded.</td>
<td>Application of methods, technologies, and theories to create an artifact that solves the problem.</td>
</tr>
<tr>
<td>Demonstration</td>
<td>Demonstrate the use of the artifact. Prove that the artifact works by solving one or more instances of the problem.</td>
<td>Knowledge of how to use the artifact to solve the problem.</td>
</tr>
<tr>
<td>Evaluation</td>
<td>How well does the artifact work? Observe and measure how well the artifact supports a solution to the problem by comparing the objectives with observed results.</td>
<td>Knowledge of relevant metrics and evaluation techniques.</td>
</tr>
<tr>
<td>Communication</td>
<td>Communicate the problem, its solution, and the utility, novelty, and effectiveness of the solution to researchers and other relevant audiences.</td>
<td>Knowledge of the disciplinary culture.</td>
</tr>
</tbody>
</table>

Table 1. Design Science Research Methodology from [45]
Design science projects can be large undertakings, involving many people and being carried out over an extended period of time. Design science methods support in structuring the work, ensuring the quality of the results, presenting the work in a logical and easily understandable way [27]. The activities or phases in the Design Science is shown in the fig. 12:

Figure 12: Phases in the Design Science Method

3.1.1 Explicate problem: In this phase the practical problem is investigated and analysed. The problem is then precisely formulated and motivated by showing that it is significant for some practice. The problem should also be of general interest, i.e., significant not only for one local practice, but also affecting many actors within an organization. The underlying causes to the problem can also be identified and analysed [27]. This phase answers the question (about the problem) with descriptive knowledge about the characteristics and environment of the problem and in some cases can also involve explanatory knowledge on problem causes.

This phase mainly consists of 3 activities, firstly to define the problem precisely, so that everyone understand the problem in the same way. Next, the problem should be well-motivated so that everyone agrees that it is a worthwhile problem to address and lastly, the underlying causes are identified, analysed and represented after conducting the root cause analysis [27].
3.1.2 **Outline Artefact and Define Requirements:** In this phase, a solution for the explicated problem is outlined in the form of an artefact. Requirements are to be determined, which can be seen as a transformation of the problem into demands on the proposed artefact. The requirements will be defined primarily for functionality but also for construction and environment [27]. The functional and non-functional requirements are noted and feasibility of addressing is checked. There are 2 main activities within this phase,

i. Outline the artefact, starts by choosing which artefact type should be designed in order to solve the problem, i.e. choosing whether the solution should be a construct, a model, a method or an instantiation [27] and the artefact chosen should be described on the overall level.

ii. Defining requirement, which can depend upon the characteristics of the problem, the outlined solution, technological opportunities, previous research including documented solutions to similar problems, and stakeholder interests and opinions [27].

3.1.3 **Design and Develop Artefact:** The aim in this phase is to create an artefact that addresses the explicated problem and fulfils the defined requirements. Designing an artefact includes determining its functionality as well as its construction [27]. [27] discusses two sub activities in this phase, Generate, produces new possible solutions, while the second one, Search and select, evaluates and selects from the generated solutions.

The activities in this phase are carried out iteratively and in parallel so that the transitions through these phases need not be a single-pass sequential process. In other words, the process tends to be iterative and incremental in nature and should be agile enough to accommodate revisions in situations where the scope cannot be completely defined up front [40].

3.1.4 **Demonstrate Artefact:** In this phase, the developed artefact is used in an illustrative or real life case, sometimes called a “proof of concept”, thereby proving the feasibility of the artefact [27]. In other words, the artefact developed is demonstrated to show that the artefact solves the explicated problem which was defined in the first phase. It consists of descriptive knowledge describing how the artefact works, and also explanatory knowledge explaining why the artefact works. A demonstration can also help in communicating an idea behind the artefact to a wide range of audience in an convincing way. On the other side, demonstration can be seen as a weak form of evaluation as the developed artefact can solve the problem in one case, but might not work in other cases.

Activities in this phase include, choosing a case on which the artefact can be applied and documenting the outcome after applying the artefact on the case.
3.1.5 **Evaluate Artefact:** This is a phase to determine how well the artefact fulfils the requirements and to what extent it can solve, or alleviate, the practical problem that motivated the research [27]. The outcome of this phase will be the descriptive knowledge and may also contain explanatory knowledge detailing how the problem is solved by the artefact.

Activities in this phase includes, first choosing an evaluation strategy which determines how to carry out the evaluation and second, carrying out the actual evaluation.

3.2 **Research Strategies and methods**

Artefacts can help people fulfil their needs, overcome their problems, and grasp new opportunities. In this endeavour, design research not only creates novel artefacts but also knowledge about them, their use, and their environment [27]. Each phase can be executed using different research strategy and research method. A briefly explanation of different ways are below,

a. **Explicate problem:** In this thesis work, the research problem was already defined precisely, well-articulated, was in context and was well-motivated. This meant that the Explicate problem phase was a small activity and more emphasis was laid to understand the research problem in more detail. A Case Study was conducted to get more details with respect to the end users.

**Case Study:** A case study investigates in detail one specific case of the general phenomenon under investigation, e.g. one organisation, one systems development project, or one mobile application. Alternatively, the purpose of a case study is to paint a rich picture of a single object or situation as a basis for obtaining a deep and comprehensive understanding of some general phenomenon [27].

Other research strategies and methods are briefly described as below,

**Survey:** In simple terms, Survey is a method of gathering information from some sample of individuals. A survey starts by generating data from a large group of objects (people, organisations, systems, etc.) often through use of questionnaires or document studies. In surveys, it is possible to investigate the needs of different kinds of stakeholders, such as managers, employees, end users, and customers [27]. However, deep and elaborate problem analysis cannot be conducted using Survey.

**Interviews:** An interview is a communication session between a researcher and a respondent, where the researcher controls the agenda by asking questions of the respondent. An interview can be structured which means that it strictly follows a predefined protocol or, it can be semi-structured or unstructured, providing opportunities for digressing from a protocol and allowing the respondent to take initiatives. One of the drawbacks of interviews is dependency on the perspective and interests of the respondent, but this problem can be mitigated by interviewing several respondents [27].
**Group Discussions:** A group discussion is a communication session in which a researcher and a group of respondents interact under the guidance of the researcher and the participants influence each other in order to generate information. It is a research method in which several respondents in conversations may inspire each other to identify and define problems in a domain [27]. One of the drawbacks for such a process is that the dominant individuals have greater impact and others views might not get voiced.

**b. Outline artefact and define requirements:** In this phase, requirements were understood in more detail and additional requirements were added to this thesis work using brainstorming sessions with different teams with CAD/PDM maintenance. All the requirements were listed and then verified with the stakeholders in order to confirm whether or not the needs were completed understood by the authors for next phase. In this phase, requirements RQ01, RQ03 and RQ04 were elicitated for the CAD/PDM maintenance and RQ03 for Virtual manufacturing team within CAD/PDM maintenance.

**Brainstorming** is a cooperative approach in which number of people collectively agree upon a solution after all of their ideas are brought up and discussed [46]. Advantage of conducting brainstorming sessions in groups is that people can meet, discuss the requirements in detail and also generate creative and imaginative requirements which other research methods like Interviews, Questionnaires or Observations studies might fail to elicitate.

Other research strategies and methods that can be used are briefly described as below,

**Surveys:** Surveys can be employed for eliciting requirements as they make it possible to understand the needs of different kinds of stakeholders. Likelihood of finding all relevant requirements increases with an comprehensive coverage of personnel's like managers, clients, end-users etc. It is also very important that Stakeholders are committed, knowledgeable and take interest in providing all the information required during eliciting requirements. In some cases, stakeholders might miss in providing some information, as they might have limited knowledge in some areas or might be biased in some ways. Surveys offer relatively low cost approach, in terms of time as well as other resources, in identifying requirements [27] and hence are one of the preferred strategy

**Case studies:** With case studies, the requirements of the stakeholder are studied in greater depth and over an extended period of time. This also assists the researcher in identifying the requirements which were not explicitly stated by the stakeholders. Case studies were ideal for this thesis work, for eliciting the requirements as it overcomes the limitations of the Surveys like, identifying requirements even if they are not stated by the stakeholder, time required to conduct the elicitation - which is more as the project timeline is properly planned and so on.
**Interviews:** Interviews are the most common method for requirements elicitation. It is adirectway to gather requirements by asking the stakeholders about the features that they want the artefact to have and any other explicit requirements they want. Interviews are also highly effective. However, if the interview questions are not drafted comprehensively, then the process of eliciting the requirements might be counterproductive and can result in incomplete information/requirements. Semi-structured and Un-structured are other forms of interviews which to some extent reduces the problems raised in Structured interview.

**c. Design and Develop artefact:** In this phase, the artefact was designed and develop to fulfilltherequirements collected in the previous phase. The developed artefact should adhere to the guidelines for Design and Develop Artefact like, the functionality and construction of each component of the artefact should be clearly described, purpose of each artefact component should be detailed (like which requirement it addresses), describe how the artefact and its components are intended to be used in its intended practice, describe in what respects the artefact is different from existing ones with respect to both functionality and construction, Specify the sources of the artefact design and describe how you have designed and developed the artefact [27].

**d. Demonstrate artefact:** In this phase, the artefact was demonstrated or tested in ordertoconfirmwhether or not the developed artefact addressed the stakeholder’s requirements. Theresultant jar file from the previous phase was moved to the test environment where tests were conducted to see whether or not they met the functional requirements (its functionality) and non-functional requirements like performance, availability etc. The functional and the non-functional requirements are listed in section 4.2 and 4.3. The demonstration is detailed in the section 4.4.

**e. Evaluate artefact:** In this phase, the Web Service is evaluated to check whether thesolution solves the explicated problem and also fulfils the defined requirements.

There are 2 evaluation strategies, ex ante and ex post evaluation. Ex ante evaluation means that the artefact is evaluated without being used, while ex post evaluation requires the artefact to be employed [27].

**Case study:** An artefact can be evaluated in-depth using Case studies as it allows theresearcher understand the reasons for its success or failure. The result of this understanding can help the researcher to redesign the artefact and create an improved version of the artefact (using iterative development model). However, there are some limitations using case studies for evaluation like, the evaluation depends heavily on the knowledge and competence of the researcher, they can also be biased on the interests and limits the results of generalizability as the study is on specific site.
**Experiment:** Experiments are popular instruments for evaluating an artefact, as they allow a researcher to achieve high internal validity by carefully controlling the conditions under which an experiment is carried out [27]. Testers use experiments to test predefined qualities of the artefact, its behaviour, its performance and so on. In this thesis work, tests were conducted on various levels. Unit test was conducted on the source code to check whether or not it is working as intended.

**Interview:** Interviews are effective instruments for gathering stakeholder opinions and perceptions about the use and value of an artefact. Interviews also allow a researcher to delve deeper into stakeholder views, as the interviewer can ask follow-up questions when needed [27]. However, competence and experience of the respondent plays an important role in the interview and respondents usually want to be positive and polite when such feedbacks on the resultant application are taken. In this thesis work, a discussion was carried out regarding whether or not the resultant application met the stakeholder requirements or not and how happy were the stakeholders with the non-functional requirements like performance, availability etc.

### 3.3 Why Design Science

Empirical research aims at describing, explaining and predicting the world. For example, the Linnaean taxonomy describes and classifies the kingdoms of animals and plants into classes, orders, families, genera, and species. The goal of empirical research, at least in the natural sciences, is to faithfully describe and explain the world as it exists regardless of human interests and biases. The world is out there, and can be explained by science so that people have a common understanding of it, irrespective of their backgrounds, traditions and values [27].

In contrast to empirical research, design research is not content only to describe, explain and predict. It also wants to change the world, to improve it and to create new worlds. Design research does this by developing artefacts that can help people fulfil their needs, overcome their problems, and grasp new opportunities [27].

In this endeavour, design research not only creates novel artefacts but also knowledge about them, their use, and their environment. Design science in these areas aims to create innovation in the form of ideas, models, methods and systems that support people in developing, using and maintaining IT solutions [27].

This thesis has explanatory and predictive objectives and an aim to create prescriptive knowledge by building an artefact. Design science research methodology is adopted in this thesis work, with an aim to create an artefact with a specific purpose to solve the real business problem which is “How can data from ENOVIA (oracle database) be exposed to other actors in
Scania IT in most effective and efficient way” and develop an presentation layer using Design Science, to answer the sub question, “Which solution can be chosen so as to make it interoperable or independent of any technologies it executes on".
4. Implementation

In this section, requirements for this thesis work are explained. Requirement specification is the process where all the requirements of the project are listed and well formulated. For the implementation of this project interviews were conducted to gather all functional and non-functional requirements.

4.1 Requirement Specification

The main outcome of Requirement Engineering is the requirements specification (RS) which is a short statement of the requirements to be fulfilled by the software [2-3].

Over time, requirements specification becomes a contract, of the product, to be built. The requirements for this thesis work were collected from different groups/actors/teams within the IWJP unit of Scania IT. Based upon the analysis of the requirements collected, the following -

functional and non-functional requirements were derived and discussed with all stakeholders. These requirements were then implemented to address the problem of this thesis.

4.2 Functional Requirements

A functional requirement describes the product features, the work that a software system/product should do. They relate to the actions that the product must carry out in order to satisfy the fundamental reasons for its existence. The plan for implementing functional requirements is detailed in the design phase of the system development model. The functional requirements for this research are listed below,

4.2.1 Requirement 1
Requirement number: RQ01
Requested by: CAD/PDM maintenance organization

Requirement Name: Exposing ENOVIA data to other project teams within SCANIA IT Expected outcome: Web-services running to display ENOVIA data

4.2.2 Requirement 2
Requirement number: RQ02
Requested by: Production Planning – Virtual manufacturing team
Requirement Name: Accessing the Project Name from the DELMIA database via Web-services
Expected outcome: Displaying the project name in an XML format with the use of Web-services

4.2.3 Requirement 3
Requirement number: RQ03
Requested by: CAD/PDM maintenance organization

Requirement Name: Active Directory Integration of the Web-services created in RQ01
Expected outcome: Activation of Single Sigh On for bypassing the login mechanism

4.2.4 Requirement 4
Requirement number: RQ04
Requested by: CAD/PDM maintenance organization

Requirement Name: Creation of presentation layer to display data of RQ01 using D3 chart libraries
Expected outcome: Displaying data using AngularJs, jQuery framework and D3 charts

4.3 Non-Functional Requirements

Non-functional requirements describes the product properties and places constraints on how the system will perform. They generally make up a significant part of the requirement specification and end user may well might accept or reject the product if these requirements are not properly incorporated to the end product.

The non-functional requirements for this Web service conforms to the guidelines as defined in the document “SOA Design Guidelines” of Scania IT. Although there are many non-functional requirements for this research, based on the discussions and feasibility of execution, the following are important non-functional requirements consider for this thesis work to address,

4.3.1 Performance: Performance in a service oriented environment can be divided into two different flavours; throughput and latency. Latency means the time taken by individual method to perform a call and throughput is the overall rate to complete a method call [47]. A Web service is considered to have good performance if it has high throughput and low latency.

4.3.2 Authentication Test: The credentials provided by the user are compared with those presented in database or file system. A user is said to be authenticated if he/she
produces the valid identifier and password which is accepted by the system, if not they are denied access. In general, it is a process of determining whether someone or something is, in fact, who or what it is declared to be.

**4.3.3 Availability:** Availability is the quality aspect of whether the Web service is present or ready for immediate use. Availability represents the probability that a service is available. As per the guidelines of Scania IT, aspects like what is the availability of the service or what can be done to increase the availability of the service has to be analysed.

**4.3.4 Scalability:** A service has to be designed with scalability in mind as it might need to handle the load (as it becomes popular). A good strategy is to design the service (which are scalable) with important aim in mind, monitor the service for its usage and redesign the service if it seems to cause any bottlenecks or affect the performance of related applications.

**4.3.5 Regulatory compliance:** Regulatory is the quality aspect of the Web service in conformance with the rules, the law, compliance with standards, and the established service level agreement. Web service uses a lot of standards such as SOAP, UDDI, and WSDL.

Output of Web Service is shown in Fig 13 and Fig 14.

![Figure 13: Type 1 of chart](image-url)
Another chart type that is available is the Line chart.

![Line chart](image)

Figure 14: Type 2 of chart

**4.3.6. Evaluate artefact:** Experiment is used to demonstrate an example for performance test. The jar file obtained from the compilation of the Java code is deployed on the Test server

```
# ls
ergoproject  log.log  stress_test.ksh
```

Spring-boot server (xxxxxx.xx.scania.com). Performance of the Web Service was checked using the test scripts (stress_test.ksh). The name of the Web Service is ergoproject and the URL for the test server is http://XXXXX/ergopject

The log.log folder under the home directory displays the Start and End time for an individual operation of the Web Service named is **ergoproject**.

```
# ./stress_test.ksh
```

Command “./stress_test.ksh” is used to execute the script

```
START 2016-05-12 11:50:59.913
END 2016-05-12 11:51:01.643
```

The output in the log.log displays the Start and End time of the individual operation which shows it only took less than 2 seconds for the Web Service to address 100 web requests. Based on the above test it can be noted that the Web Service was able to meet the stakeholders requirement of response time for throughput and latency.
**Deployment:** The jar file was deployed on test server.
5. Results and Discussions

This section consists of the results and discussions for each requirement. It details how each requirement was fulfilled and how tests were carried out.

JMG (Java Maintenance Group) Reference architecture is used as a platform for developing web services. In the persistence layer actual code is implemented and web service is triggered from this layer and Fig 15 represents the output achieved in the browser:

![Figure 15: JSON output of Enovia web service](image)

Above is the outcome in the json form and then in the presentation layer front-end code is implemented and the output can be seen in Fig 16:

![Figure 16: Data visualization of Enovia web service](image)
2nd requirement also involves the development of web service and its users are virtual manufacturing team. For this web service same framework is used as for the 1st requirement and its output can be visualized in Fig 17:

![Output of Delmia web service](image)

**Figure 17: Output of Delmia web service**

This is the XML format which consists of list of projects in that environment.

**Testing of web services**

Testing the ENOVIA and DELMIA web services API using Postman REST client. Postman REST client is a Google Chrome App for direct HTTP manipulation and testing. This tool provides an easy way to perform HTTP requests and view the responses.

1. Install Postman REST Client
2. Launch Postman REST Client
3. Point to Enovia and DELMIA Web Services and make API calls

In this way we can test were conducted for testing these web services.
6. Conclusion and Future work

Building the web services can help the organisation evolve its IT system by helping in achieving business needs promptly. These web services can be reused across different areas in an organisation which in turn will save lot of time and money. Instead of querying the data for a specific application which takes time and also an inefficient way as well as it hinders the performance of the organisation so implementation of web services is a better option. The goal of this thesis focuses on developing web services and visualizing that data in a graphical format.

In order to build the web service a literature review was conducted about existing technologies and approaches. From literature study, two api's were considered one was REST and other was SOAP. Among these two SOAP was not considered because its more complicated than Rest. Also, SOAP only allows XML but we also needed the JSON format and also in terms of performance and scalability REST is better. So from research, it is concluded that REST is a better option.

Similarly for the front-end as well research was conducted to decide which charting library to be used for data visualization. And after comparing some libraries D3.JS is chosen which is very flexible.

D3 stands for Data-Driven Documents. It is a JavaScript library for producing dynamic, interactive data visualizations in web browsers [48]. The graphs produced through D3.JS can be rendered through other libraries as well, For example, C3.JS which is a library above D3.JS provides much more functionalities than D3 so it can be used as a basic library to build the graphs and on top of which we can add other features.

The working prototype is created which meets all functional and non-functional requirements but it still lacks the security measures as the database used for this web service is accessible to everybody within Scania so we need to control this and limit the access of database to few users who are using this web service and this can be implemented using Active Directory integration. Using Active directory integration we can create group of selected users and provide access to only that group.

The other functionality which can also be part of future work is implementation of system alerts. System alert functionality will work like a news feed which can provide daily reporting regarding any numbers of users, any fluctuations in the database or it can be any information related to the organisation.
7. References

1. Implementation of CAD/CAM system CATIA V5 in Simulation of CNC Machining Process, Rozmarina Dubovska, Jaroslav Jambor, Jozef Majerik, University of Hradec Kralove, Czech Republic.

2. A web-enabled PDM system in a collaborative design environment, X. William Xu, Tony Liu, Department of Mechanical Engineering, University of Auckland.


8. Interoperable enterprise systems: Principles, concepts, and methods, F.B. Vernadat, LGIPM:


22. Web Services Conceptual Architecture (WSCA 1.0), Heather Kreger, IBM Software Group


26. ENOVIA VPLM VPM Navigator Product Design, Wichita State University.

27. A Design Science Primer, 1st edition by Paul Johannesson and Erik Perjons


30. Development Methodologies, Dr. James A. Bednar, Dr. David Robertson, SEOC2 Spring 2005: Methodologies


37. An Introduction to Agile Software Development, June 2007, SERENA, serena.com


41. Integrating Advanced Computer-Aided Design, Manufacturing and Numerical Control: Principles and Implementations, Xun Xu, University of Auckland, New Zealand


45. K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, J Manage Info Syst, 24 (3) (2008), pp. 45–77

