A Distributed Approach for Global Product Lifecycle Management

JULIO VARGAS-ORELLANA

KTH Information and
Communication Technology

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

A Distributed Approach for Global Product Lifecycle Management

Julio Vargas-Orellana

juliov@kth.se

December 30th, 2013

Master's thesis
Master of Software Engineering of Distributed Systems

Examiner: Gerald Q. Maguire Jr.
Supervisor: Rolf Laudenbach

School of Information and Communication Technology (ICT)
KTH Royal Institute of Technology
Stockholm, Sweden

Abstract

Product Lifecycle Management (PLM) is a holistic approach for managing product information throughout its life cycle. It integrates different concepts that have emerged due to changes in the manufacturing process as a result of globalization, increased competition, demand for more innovative products, and other reasons. These changes have leaded to a shift from a model with a single-location for product development to a model in which a complex network of specialized companies collaborate.

This global collaborative PLM implies that companies from different parts of the world work together and must share information; hence the underlying PLM system is required to facilitate data management throughout this collaborative process. In addition, it is also necessary to address the challenges due to the new model being a distributed activity, as today this PLM system is a specialized distributed system.

Maintaining data consistency can be challenging because collaborators can use heterogeneous PLM systems together with their own databases. The later cannot be shared due to the risks of exposing their knowledge base and business processes. Another consideration in global collaboration is that data is transmitted to remote locations. As a result network latency can be large; this can cause problems particularly when large files are exchanged, such as may be the case for CAD design models.

This thesis proposes a solution enabling a global PLM which addresses the challenges described above. The approach consists of connecting collaborators' nodes in a network that is constructed by grouping them with respect to *intra-site* latency. Each group implements a coordination mechanism based on the election of a node which is subsequently in charge of coordinating data access. The groups communicate via a publish-subscribe communication pattern, publishing and subscribing to events related to the resources being shared. The integration of the solution is through a Service-oriented Architecture (SOA) implementing web services that can be consumed by a PLM system. A prototype has been implemented and its applicability is analysed by evaluating its functionality in a collaborative scenario based on the Aras Innovator PLM platform.

The evaluation was made by simulating the solution proposed and comparing it with a centralized approach. The results particularly showed that the proposed solution could reduce the intra-latency compared to a centralized approach if the collaborators are organized in collaborative groups, that exchange most of the information inside the group rather than inter-group.

Keywords

Product Lifecycle Management, distributed systems, cluster head

Sammanfattning

Product Lifecycle Management (PLM) är en helhetssyn som hanterar produktinformation under deras hela livscykel. PLM integrerar olika koncept som har dykt upp på grund av förändringar i tillverkningsprocessen som en följd av globalisering, stor konkurrens, efterfrågan på mer innovativa produkter , och andra orsaker. Dessa förändringar har blyad till en övergång från en modell med en enda plats för produktutveckling till en modell där ett komplext nätverk av specialiserade företag samarbetar.

Detta globala samarbete inom PLM innebär att företag från olika delar av världen arbetar tillsammans och delar information. Det underliggande PLM-systemet krävs att underlätta datahantering hela denna samverkande process. Dessutom är det också nödvändigt att hantera utmaningar beroende på den nya distribuerade modellen som gör PLM -system blir specialiserade distribuerade system.

Underhålla uppgifter konsekvens kan vara en utmaning eftersom kollaboratörer kan använda heterogena PLM-system med sina egna databaser som inte kan delas på grund av riskerna för att utsätta sin kunskapsbas och affärsprocesser . En annan faktor i den globala samarbetet är att data överförs till avlägsna platser . Som ett resultat nätverksfördröjningen kan vara stora , vilket kan orsaka problem speciellt när stora filer utbyts , exempelvis CAD-modeller.

Detta masterarbete föreslår en lösning för att möjliggöra en global PLM som tar upp de utmaningar som beskrivs ovan. Tillvägagångssättet består av anslutande kollaboratörer noder i ett nätverk som konstrueras genom att gruppera dem i förhållande till intra-site latens. Varje grupp genomför en mekanism för samordning grundas på valet av en nod som därefter ansvarar för samordningen av dataåtkomst. Grupperna kommunicerar via en publicera-prenumerera kommunikationen mönster av att publicera och prenumerera på händelser relaterade till de resurser som delas. Integrationen av lösningen är genom en Service-Oriented Architecture (SOA) genomföra webbtjänster som kan konsumeras av ett PLM-system. En prototyp har genomförts och dess användbarhet analyseras genom att utvärdera dess funktionalitet i en kollaborativ scenario baserat på Aras Innovator PLM-plattform.

Resultaten visade att den föreslagna lösningen skulle kunna minska intra-latens jämfört med en centraliserad strategi om kollaboratörer är organiserade i kollaborativa grupper, varje grupp är ansvarig för utformningen ett delsystem av produkten och därmed utbyta mesta av informationen inom gruppen snarare än inter-gruppen.

Nyckelord

Product Lifecycle Management, distribuerade system, cluster head

Table of Contents

List of Figures

List of Tables

Document Version Control

Version	Date	Purpose/ Changes	Author
1	02/05/2013	Literature Study	Julio Vargas
1.1	18/07/2013	Corrections, modifications	Julio Vargas
2	19/09/2013	Draft 1	Julio Vargas
2.1	26/10/2013	Draft 2	Julio Vargas
3	14/11/2013	Presentation version	Julio Vargas
3.1	30/12/2013	Final Version	Julio Vargas

List of Acronyms and Abbreviations

BOM	bill of materials
B2B	business-to-business
CAD	computer-aided design
CAE	computer-aided engineering
CAM	computer-aided manufacturing
CPD	collaborative product development
DSM	distributed shared memory
ERP	enterprise resource planning
EBOM	Engineering bill of material
FMS	file management system
FTP	file transfer protocol
IIS	Internet Information Services
IT	information technology
LOD	levels of detail
PDM	product data management
PLM	product lifecycle management
RWND	receive window size
SOA	service-oriented architecture
SOAP	simple object access protocol
UI	the user interface
WCF	Windows communication foundation
WSDL	web services description language
XML	extensible mark-up language

1 Introduction

This chapter gives a brief introduction to PLM and the context in which it has emerged. This chapter also gives a short explanation of the changes that have taken place recently in design and manufacturing processes and the different concepts that have appeared for managing a product throughout its different life cycle stages. This chapter should enable the reader to understand how PLM has emerged as a holistic approach for orchestrating product lifecycle information. Finally, a description of the current project is given as part of this chapter. This description includes a statement of the goals and aimed to be achieved.

1.1 Background

In recent years, the manufacturing process has evolved from single-location production to global production with companies changing their structures in response to the changing environment. Design and manufacturing has been spread to different locations around the world, taking competitive advantages of each country's labour cost, closeness to raw materials, etc.; while at the same time creating new market opportunities. As a result, product development has become a dynamic collaborative process in which several participants (offices, units, departments, companies, partners, suppliers, retailers, and customers) situated in different locations interact. For example, in the automotive industry, Ford Motor company utilized 37 suppliers in 11 different countries in 2008 [1].

In order to address the new challenges of product development, a new approach known as Product Lifecycle Management (PLM) has emerged. PLM is a systematic concept to manage products and the information related to a product from the initial idea until the product is retired. A *PLM system* provides all the mechanisms necessary for implementing PLM in a company and provides the means for managing all of the data that is needed to document the product and its development through its entire lifespan.

The increasing complexity of product manufacturing, complex business networks, and the dynamics of distributed development and manufacturing have introduced new requirements for PLM. The implementation of PLM becomes more challenging when one considers that each company has its own structure, objectives, and strategies. Moreover, many companies maintain different systems under different platforms as a result of mergers and acquisitions, making integration even more challenging.

Under these circumstances, PLM systems have to provide integrated management of product information in a collaborative network to facilitate the development of innovative products, make the process efficient and cost-effective, and enable the company to release new products in shorter periods of time. As an example, a company that renewed less than 10% of its products in the early 1990s, renewed about 75% of its products in 2009 [1].

1.2 Project Thesis Description

A global collaborative PLM implies that companies from different parts of the world work together sharing information through the product’s life cycle. In contrast to development at a single location, global collaborative product development introduces additional challenges not only from an organizational but also from a technical perspective. The underlying PLM system is required to facilitate data management throughout this collaborative process. In addition, it is necessary to address the technical challenges due to a PLM system being a specialized distributed system.

Distributed PLM collaboration involves data exchange between multidisciplinary experts. This data includes technical documentation, such as computer aided design (CAD) design models and other types of files. Since a product idea evolves constantly during its life cycle with in some cases changes being made very often, it is important to avoid any inconsistency in the shared information that could lead to errors or misunderstandings that affect the whole process.

Maintaining the data consistency can be challenging because collaborators can have heterogeneous PLM systems with their own data formats. Additionally, collaborators rarely want to provide direct access to their databases, which would disclose their knowledge base and business processes. Moreover, in some cases legislation regulates collaboration that involves intellectual property sharing between companies located in specific countries. Consequently, it is important to find a coordination mechanism that permits maintaining data consistency without complete sharing of the participants' databases.

Having global collaborators also implies that data is transmitted between remote locations. If these locations are quite distant then network latency may negatively affect the communication between participants. In addition, several collaborative applications have minimum latency requirements [2]. An approach that considers this latency and improves throughput can consequently improve the overall system's performance, which can yield a financial benefit. For example, Amazon found that every 100 ms of latency cost them 1% in sales and a broker could lose $4 million in revenue per millisecond if their electronic trading platform is 5 ms behind the competition [3]. In general, a reduction of latency in a distributed system can bring cost savings to enterprises [4].

A successfully connected global team provides a network of global resources that maximizes the usage of available bandwidth, facilitates innovation, speed time-to-market, and reduces costs [5][6]. However, if collaboration fails, the process can become long, resource utilization, and costly rework [7]. Therefore, the importance of well connected collaborative environment.

This thesis reviews the PLM concept, its current challenges, and it also glances at the future of PLM. This thesis gives an introduction to distributed systems and the Service-oriented Architecture (SOA) approach. SOA is an approach to building flexible systems and integrating different technologies. Next, related work in the field is studied. Next, an approach for sharing data in a global collaborative PLM environment based on SOA is presented, considering a mechanism for maintaining the data consistency and arranging the collaborators in order to minimize the latency. Finally, a case study of commercial PLM software is included, specifically Aras Innovator, which can benefit from the proposed approach.

1.2.1 Goal of the current project

The objective of the current thesis is to propose an alternative that enables data sharing between collaborators in a global PLM system. The proposal should provide mechanisms to coordinate data exchange in order to maintain consistency, including a concurrency control model adequate to a global collaborative PLM system. Before proposing such a solution, it is important to understand the concept of PLM and its future challenges and requirements. This understanding will provide the basis for designing a solution that can adapt to the dynamic context in which PLM is immersed. As the geographical distribution of the collaborators implies potentially high network latency the proposed solution should also consider this aspect in order to minimize its negative impact. Finally, the proposed solution should consider the integration of different PLM systems. For this reason SOA emerges as an obvious chose since it can provide interoperability and improve information flow.

1.2.2 Thesis structure

The first chapter presented the problem description and the goals of the thesis project. The second chapter introduces the background and concepts used, including an introduction to PLM and its challenges (section 2.1), collaborative product development in (section 2.2), and in section 2.3 provides a brief overview of basic concepts of distributed systems. Section 2.4 describes web services and section 2.5 is about SOA and its application in PLM. Section 2.9 presents different relevant works from which the current project got inspiration or used as a support. Chapter 3 describes the method, including a detailed description of the sharing mechanism. Chapter 4 presents the prototype and an analysis of the proposed solution in a global PLM scenario. Finally, Chapter 5 presents the conclusions of this master's thesis project, suggests some future work, and offers some reflections on the social, economic, and ethical aspects of this thesis project.

2 Background

This chapter provides the basic fundamentals of PLM and gives an overview of the challenges that have emerged for PLM systems and the collaborative approach it has taken. It is also introduces some basic concepts of distributed systems and current technologies, such as SOA, on which the current project is based. Finally, summaries of related works are presented.

2.1 Product Lifecycle Management

Product Lifecycle Management (PLM) has emerged as a holistic approach to support and coordinate all the separate activities previously taking place in different parts of the organization and as viewed from different perspectives over the life of a product. Previous attempts to standardize different stages of a product's lifecycle focused mainly on a single aspect of part of the complete lifecycle. Therefore, different concepts emerged such as Bill of Materials (BOM), Computer-aided Design (CAD), Computer-aided Manufacturing (CAM), Computer-aided Engineering (CAE), Product Data Management (PDM), and others [7] Each of these concepts addresses a specific problem of product development and is commonly supported by a specific information system. As a result, each of the product's development phases is managed by a different independent system, rather than a single integrated system.

The lack of a single integrated system to support the whole lifecycle through its phases (Figure 2-1) can result in a loss of control and cause a variety of different problems (such as delays in releases, failures, product recalls, exceeding the budget, and others). Any of those problems can cause consumer dissatisfaction, damage the company's image, and may result in a loss of revenues to other companies who bring new products faster and more efficiently [8].

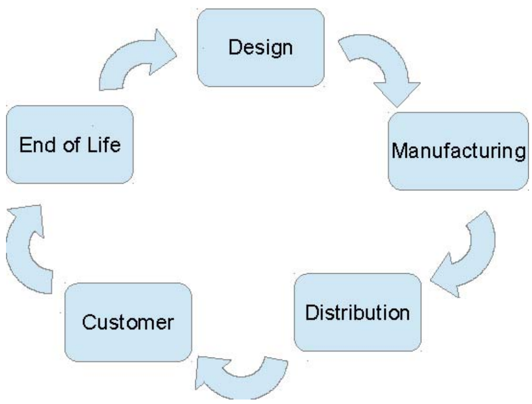


Figure 2-1: Simplified view of a product's lifecycle

2.2 Product Lifecycle Management Systems

A PLM system is an information technology (IT) system or set of IT-systems whose main purpose is to integrate the functions of a company by connecting, controlling, and integrating different business processes through product data with the aid of technology [28].

Currently, different specialized IT-systems are utilized during the product's lifecycle. Each of these systems is very efficient in its own area, for example, CAD systems are extensively used during the design phase. However, some of these specialized systems can cause bottlenecks to appear elsewhere in the data flow. The task of a PLM system is to interconnect these separate IT systems in order to permit a fluent data exchange across all of the different processes [1].

In general, all commercial PLM systems offer similar features and components (see Figure 2-2), such as[56]:

A file vault is a repository of files that contains information about the product at different stages of its life cycle. This information can include CAD designs, requirements documents, and other relevant files.

A metadata base maintains the relationships between the product data produced by the different PLM applications, the structure of the information, and the rules for systematically recording relevant information.

An application provides a user interface to carry out all the PLM functions. Additionally, this application communicates with other PLM applications.

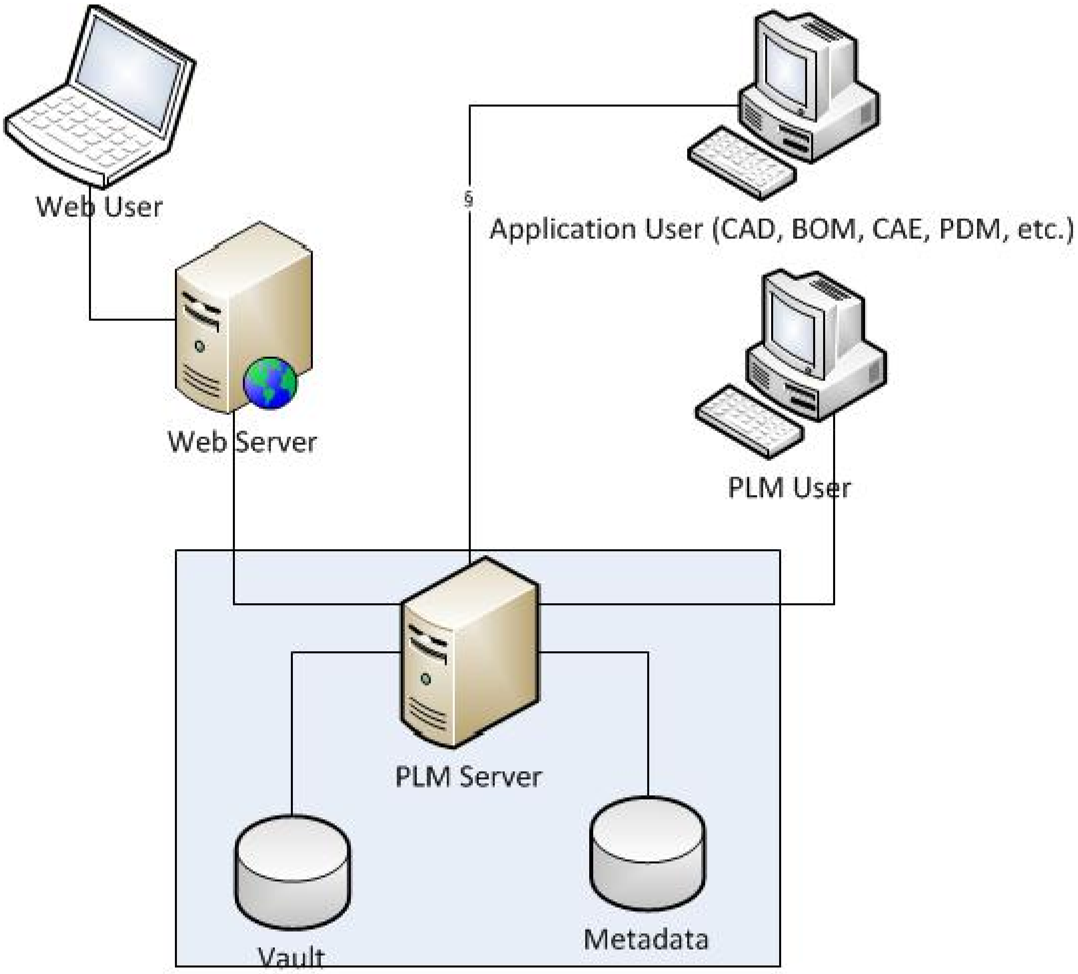


Figure 2-2: Basic PLM system components

2.2.1 PLM Challenges and Requirements

Products are now frequently the result of a collaborative process which involves the efforts of multiple participants; therefore, PLM systems should provide mechanisms that permits data exchange between these participants. Products are also becoming more sophisticated, with more complex designs, more features, and a larger number of parts, thus generating more information, including larger CAD files and inventory data, hence motivating the use of scalable PLM systems. For example, in the aerospace industry, each product has thousands of complex parts, which are designed and manufactured all over the world. As a result there is a great deal of data flowing between offices, partners, and other stakeholders during the whole development, produce, operations, and recycling process.

Data integrity is an important concern in current PLM systems. Outdated or inconsistent data could have a significant negative impact on the process and can result in mismatches between the requirements and the specifications, delays, or even failures of the product (before or after entering service). Each of these impairments can lead to a cost in time and/or money, or in the worst case cause accidents and even deaths.

Security is also very important because some parts of this data are shared not only inside the company's network, but also with partners outside of the company. This is particularly true for some specific industries, such as aerospace or military, as in these domains confidentiality is a very important issue. In other industries, any leak of information concerning new upcoming products to competitors can result in a launch failure, loss of market share, loss of the ability to protect intellectual property of the product being developed, etc.

2.2.2 Future of PLM

As mentioned earlier, the complexity of PLM has increased due to the constant changes in production, including a larger network of providers and more complex products with more parts. Factors such as globalization, global economy, geopolitics, and social demands for even more sustainable products and production have accelerated changes in companies, leading to changes in business models. All of these factors are influencing the future of PLM.

In the case of products, technological advances and economical changes have led to a situation in which consumers increasingly demand advanced products. At the same time, globalization has facilitated companies looking for markets in multiple different countries, hence increasing competition. It is probable that competition will become even more intense in the future and only those companies that are the most efficient and most innovative will survive.

Another aspect that has become more relevant is the protection of the environment and the impact of the industry on it. There are increasingly demanding environmental regulations being applied to production and regulating different aspects during the life cycle of a product. In many cases the lifetime of a product can be many years. In the past, there was little concern about the product after its useful lifetime. However, regulations are changing the way products have to be treated after the product is retired from use. These regulations often impose higher costs, affecting competitiveness - particularly for companies located in countries which stricter regulations [35].

The global economy may be a key factor in the future. For example, in the recent years the combination of a weak economy in Europe and a strong currency has negatively affected many Swedish export companies as the combination has effectively increased production costs. In such scenario, a flexible system that facilitates reducing purchasing costs by sourcing from alternative suppliers could minimize the negative impact of a strong currency. After the crisis of 2009, Scania has reduced the times from order to delivery and consequently increased their flexibility, therefore minimizing the negative impact of this adverse economic situation [54].

The factors introduced in this section imply that in the future organizations will demand greater efficiency and more innovation in order to cope all these challenges. In addition, organizations are becoming more interdependent, leading to more complex networks of suppliers and more advanced products. At the same time, clients will demand advanced products more often and the competition will be fiercer. Another important change is regulations which demand integral product management not only during production but also controlling the parts used in production and extending to the useful lifetime of the product. For all these reasons, PLM will probably become increasingly relevant as a global oriented concept and its application by companies will increase.

2.2.3 Collaborative Product Development

Collaborative Product Development (CPD) can be seen as a business strategy, driven by the growth of the global market. The purpose of CPD is to increase efficiency, reduce costs, and accelerate innovative product development by combining the strengths and expertise of geographically dispersed teams [55]. Various factors, such as globalization and outsourcing, are increasing the number of collaborators in projects, thus CPD could speed up the decision-making of all the different stakeholders [9]. Recently, companies in industries, such as aerospace and automotive, have been combining the capability of specialized partners and their own internal expertise and size to develop innovative business models and processes [10].

CPD can occur in any of PLM phases, requiring the integration of information and processes, particularly in PLM applications such as Product Data Management (PDM), Computer-Aid Design (CAD), and team-work tools. Figure 2-4 illustrates collaboration among different companies in different processes. Another important aspect of collaboration among companies is that their relationships are often temporary and can change according to each company's interests and projects. For example, it is possible for companies to collaborate on a specific project, while being competitors in a different area.

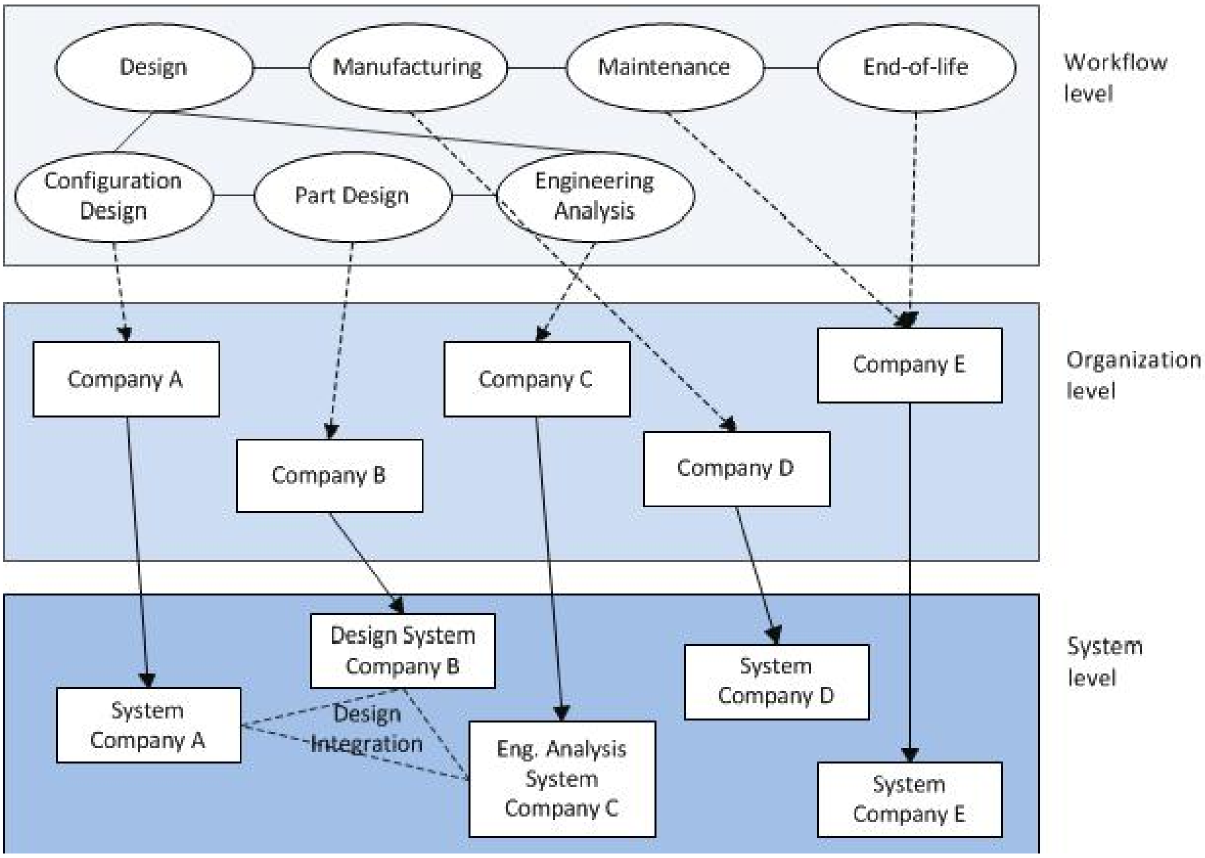


Figure 2-4: System integration for collaborative product development in a distributed environment (Adapted from figure 1 of [46])

It could be seen in Figure 2-4 different companies may collaborate along the product's life cycle in specific processes and phases. A given company can be involved in one or more of these processes. At the same time, there are interdependencies between processes and companies. These interdependencies are dynamic and they can change during the product's life cycle and over time. This figure also shows that different systems of each of the companies support the processes and in some cases these systems need to interact and exchange data, for example, during the design phase.

Collaboration occurs not only in the private sector, but also in the public sector or in a combination of the two. A result of fast paced industries and strong competition is the demand for increased exchange of knowledge and expertise in order to be able to compete. Error: Reference source not found showed an example of a partnership between different institutions who exchange information and knowledge in concrete projects aiming for greater efficiency and more rapid innovative in the European aerospace industry.

2.3 Distributed Systems

A distributed system refers to a set of (software or hardware) components located at physically separated computers interconnected by networks that coordinate their actions through message passing with the principal aim of sharing resources (hardware, software, data, etc.) [11]. Such a distributed system comes with consequences, such as *concurrency*, *no global clock,* and *independent failures* [11]. Examples of distributed systems can be found in multiple applications, including PLM systems.

Depending on the scale and the type of distributed systems there are several challenges within distributed systems to overcome. According to George F. Coulouris, et al. some of these challenges are [11]:

Heterogeneity Distributed systems are typically heterogeneous, having different types of computer hardware, networks, operating systems, programming languages, interfaces, protocols, etc. The challenge is to develop systems that can work well under such circumstances.

Openness All services should be easily accessed by clients, whether the client is local or remote. Systems should provide enough flexibility to support new services or new types of clients.

Security Access to resources by unauthorized users should be prevented, while authorized users should be guaranteed that they can make appropriate use of them. Security has three components [12]:

- Confidentiality refers to the protection of the content against unauthorized users
- Integrity protection protects against data alteration or corruption
- Availability means that resources should be remain accessible.

Scalability Scalability means that a system can increase the number of resources and the number of users while maintaining its cost-effectiveness. Such a scalable system could be deployed in different scenarios: with different amounts of data and different number of users, provide different levels of geographical coverage, and support different sized networks while maintaining the quality of service.

Failure Handling Any failure of the hardware or software can cause unexpected behaviour or even stop the task before it is completed. In the case of distributed systems, failures are mostly partial, which means that only some of the components fail while others continue running [11]. Therefore, the difficulty is properly handling these failure events, as a distributed system cannot simply be restarted as could be done with a single piece of software or hardware.

Concurrency Distributed systems allow resource sharing, thus more than one client may require the same resource at the same time. One approach to handling this situation is to process a single request at a time. However, this can limit the performance of the system. Another approach permits multiple requests to be processed concurrently, but when conflicts occur, it is responsibility of the system to avoid or resolve these conflicts.

Transparency The system complexity should be hidden from the user, thus the system should be perceived as a whole rather than as independent components. There two main goals of transparency [11]:

- Users must be able to access the system in the same manner.
- Users do not need to know the physical location of the resources they request.

2.4 Indirect Communication in Distributed Systems

Indirect communication can be defined as communication between entities without a direct coupling by using an intermediary [11]. In contrast to the client-server model (which is a coupled approach), indirect communication provides greater flexibility due to two important properties: space and time decoupling [11].

Different alternatives for indirect communication are: group communication [13], publish-subscribe systems [14], message queues or forwarding [15], and shared memory [16]. Each of these will be described further in the following paragraphs.

2.4.1 Group Communication

In group communication processes are organized in groups and the communication occurs through messages sent to the group. Each member of the group receives the message. In comparison with multicast communication, group communication facilitates group membership management and adds reliability, guaranteed order, and failure detection. Applications of group communication include areas such as the financial industry, multi-user games, highly available servers, and support system monitoring and management.

Figure 2-6 shows the basic behaviour of group communication. A process sends a message to the group and every group member receives the message. The group membership management maintains a view of the group, updating it whenever a process fails or a new process joins the group.

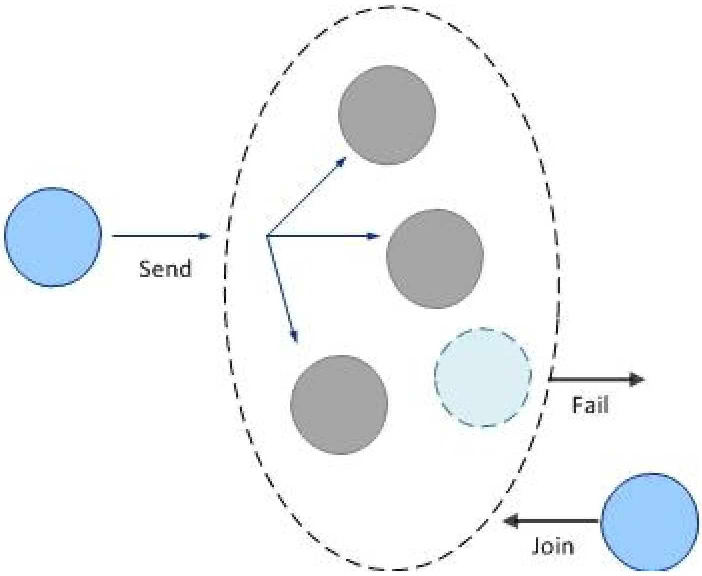


Figure 2-6: Group Communication

2.4.2 Publish-subscribe systems

A publish-subscribe system is a communication technique based on event processing. Subscribers define their interests by subscribing to topics and an event service is responsible for matching events published by publishers with the interests of the subscribers. Publish-subscribe is a one-to-many system since an event can be delivered to more than one subscriber. These types of systems are widely used for large-scale dissemination of events, for example, in support of collaborative work where participants need to be informed of specific events. Another example is monitoring systems.

Publish-subscribe systems have two important characteristics: heterogeneity and asynchrony. Heterogeneity is achieved by using events as means of communication, thus enabling components to interoperate. Asynchrony arises because messages are sent by publishers to an event service which decouples publishers and subscribers.

In addition, there are other desirable characteristics of publish-subscribe systems. Expressiveness refers to the degree to which the interests of the subscribers are captured. Scalability is related to the capacity of the system in terms of the number of users it can support. Modularity is also desirable in order to facilitate implementation and interoperability.

Figure 2-7 illustrates the basic function of the publish-subscribe technique and its basic components. A publisher publishes an event under a topic and the event is delivered by the event system to subscribers to that topic. Subscribers can unsubscribe to topics when they are no longer interested in the topic.

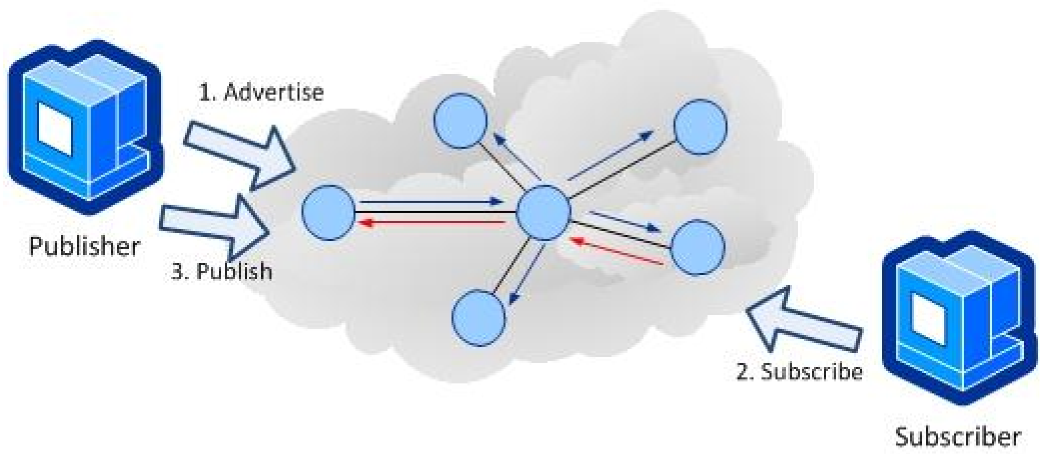


Figure 2-7: Publish-Subscribe systems

2.4.3 Message queuing

A message queuing technique uses a message queue as an indirect means of communication. The mechanism is very simple, a process sends a message to the queue, which stores the message until another process consumes the message. In contrast to a client-server model, message queuing provides asynchronous delivery and uses persistent storage to back up the messages until they are delivered.

Figure 2-8 shows the basic components of a message queuing system. A producer sends a message to a queue and the message is stored until a consumer receives the message. The poll operation permits a consumer to learn if there is a message available and the notify operation permits the message queuing system to inform the consumer when there is a message available.

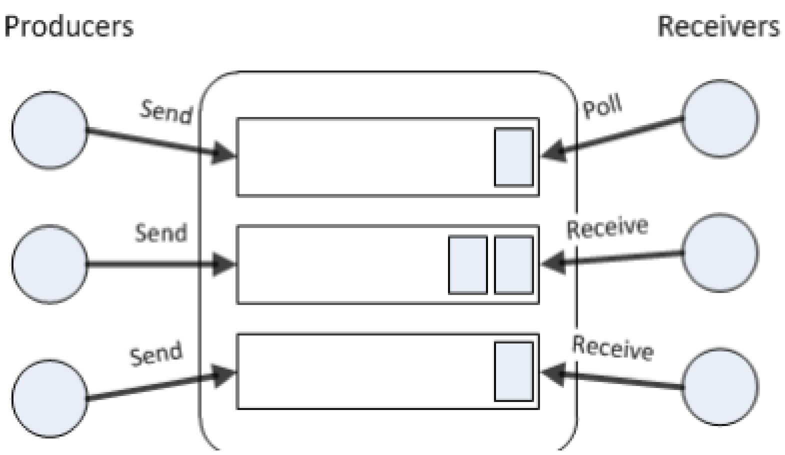


Figure 2-8: Message queueing (image taken from [11]).

2.4.4 Distributed shared Memory

The distributed shared memory (DSM) technique shares data between computers without sharing physical memory. The main characteristics of this technique are that a process can access a distributed distributed memory as if this memory is part of its own address space. This method is mainly used in parallel applications or when shared items need to be accessed directly. The distributed shared memory model is shown in Figure 2-9.

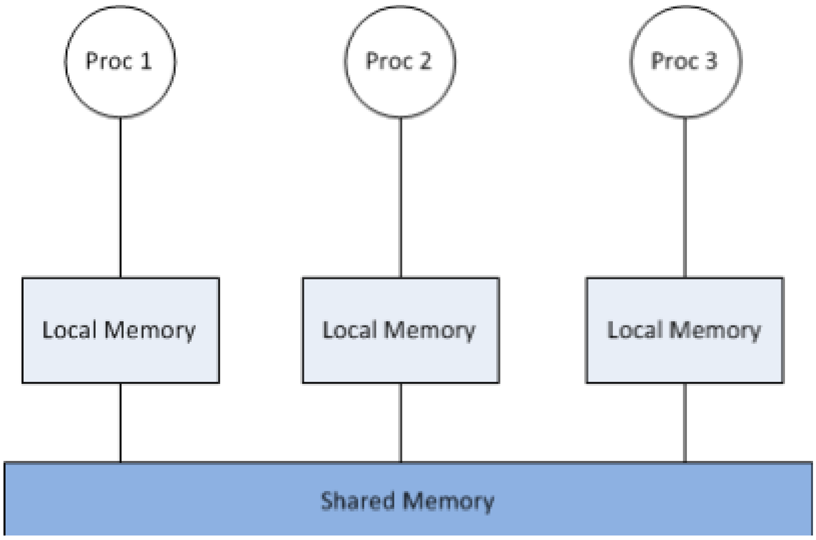


Figure 2-9: Distributed shared memory communication model (image taken from [11]).

2.5 Web Services

Web services have emerged to support interoperability and to extend the functionality provided by systems connected to a network, such as the Internet. Web services permit clients from one organization to interact with servers of a different organization, independent of the underlying platform on which these services are deployed. Another characteristic is that services provided by an organization can be invoked by services provided by other organizations. This capability permits web services to be built on top of other services, making re-use possible across organizations and providing integration among a number of businesses.

There are two common implementations of web services which differ in the data exchange protocols used. These two implementation approaches are based upon:

- Simple Object Access Protocol (SOAP) [20] and
- Representational State Transfer (REST) protocol.

The first approach is based on SOAP, which is an XML language that defines a message-based architecture and message formats. In addition, SOAP provides a description of the operations provided by the service using the Web Services Description Language (WSDL) [18], which is based on XML (see Figure 2-10). WSDL is an XML-based language for describing the functionality provided by a web services. This description, acts like a contract as it describes the input parameters and the format of the data structure which will be returned. This type of web service is based on standards, such as WS-security, WS-orchestration, WS-AtomicTransaction, and other specifications [17], which provide additional functionality. This implementation approach is a good alternative for stateful operations and for integrating separate systems.

XML [19], a W3C standard, is a simple mark-up language readable by both humans and computers and is extensively used in the Internet. The main characteristic of XML is that it does not provide a limited set of tags and elements, but rather developers can define their own tags and elements according to the semantics needed by their application.

Communication between clients and the services is done through messages. A common protocol for web services data exchange is SOAP. SOAP provides a framework for data exchange between services. Data is serialized into XML prior to being transmitted, and then de-serialized at the destination. SOAP is a stateless one-way protocol, with a very simple structure: a header and a body.

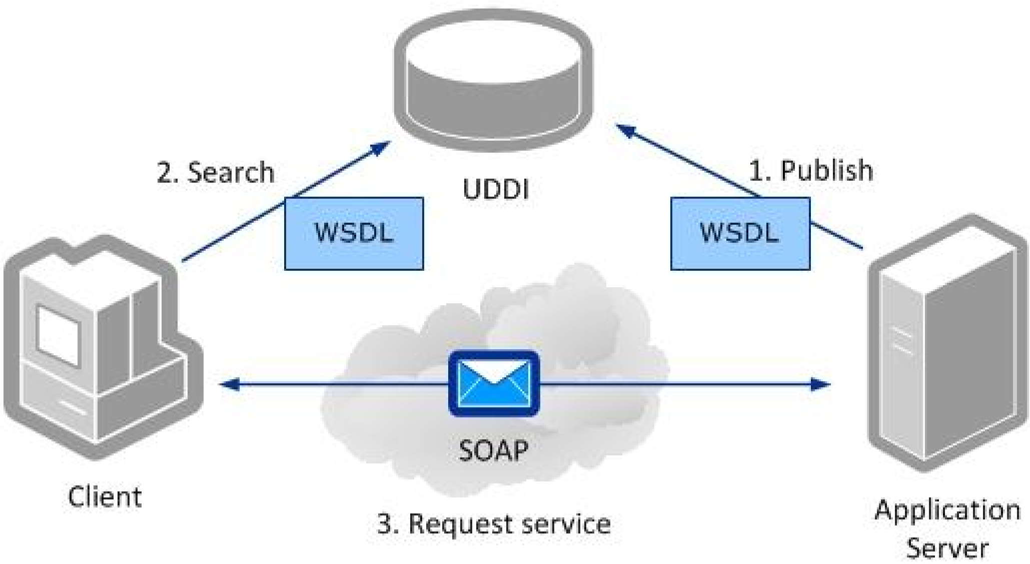


Figure 2-10: Web service based on SOAP

The second approach, based on REST, is better integrated with HTTP as it uses the standard HTTP request and response mechanisms. REST provides a simple alternative for web services since it does not required XML or WSDL descriptions. REST based web services are good for stateless operations and caching situations.

It is difficult to make a direct comparison of both approaches since they have different characteristics and capabilities. REST offers a lighter alternative, while SOAP provides a full set of standards in order to provide extensibility via a set of specifications which add additional functionality.

2.6 Standard for Product Data Exchange (STEP)

STEP is a standard specified in ISO 10303. STEP is designed to support product data modelling. STEP is maintained by ISO technical committee TC 184, Automation systems and integration, sub-committee SC 4, Industrial data. Its main aim is to provide product data modelling mechanisms and enhance data exchange and sharing in an integrated manufacturing environment. Most of its parts are described in the EXPRESS language. EXPRESS is a description language that can be understood by both computers and humans. One of the current applications of STEP is the exchange of three-dimensional geometric product data among different PLM systems.

Two mechanisms for data exchange are provided by STEP: neutral format and shared database. In order to use STEP, there are two required steps: (1) a conversion from the internal format to the STEP standard, and (2) the reverse conversion from the STEP standard to an internal format [21]. Figure 2-11 illustrates how STEP can be used for data exchange between different PLM applications. One of the principal characteristics of STEP is that it can be used independently of the information system being used [22].

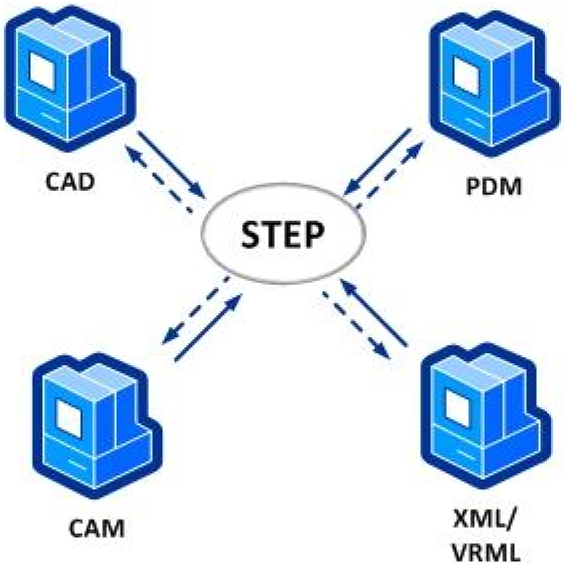


Figure 2-11: Data exchange between different applications using STEP.

2.7 Service-Oriented Architecture

A Service-Oriented Architecture (SOA) is a set of design principals, policies, and practices that enables application functionality to be provided and consumed as a set of services [23]. SOA is built upon the concept of a service, which exists both at a business level and at the technological level. On a business level, a service can be seen as a repetitive business task, such as reserving a ticket or opening a back account. At a technological level, a service refers to a software artefact that provides functionality to the real world and encapsulates reusable business functions [24].

According to SOA, services can be published and discovered, and a service must have an administrator and an owner, who might be a person or an organization. Services are designed to be loosely coupled, and thus capable of interacting with other services which may be provided by other organizations. For this reason, it is desirable that services be interoperable so that they can be used in Business-To-Business (B2B) integration.

The interaction between clients and a service is done through messages, according to the specification of the service. Although SOA can be implemented using different technologies, the most common approach uses web services because they are loosely coupled and provide seamless interoperability.

SOA promotes a mash-up approach for developing new applications, thus a new service can be created by combining other services. Currently, multiple services are offered by companies such as Amazon [25] and Google [26]. These services are available through published interfaces with functionality that can be easily reused. Figure 2-12 shows a basic web services architecture. This figure shows how different data sources can be abstracted through web services. It also shows that web services can be consumed by another web service reusing and combining them for a specific business case.

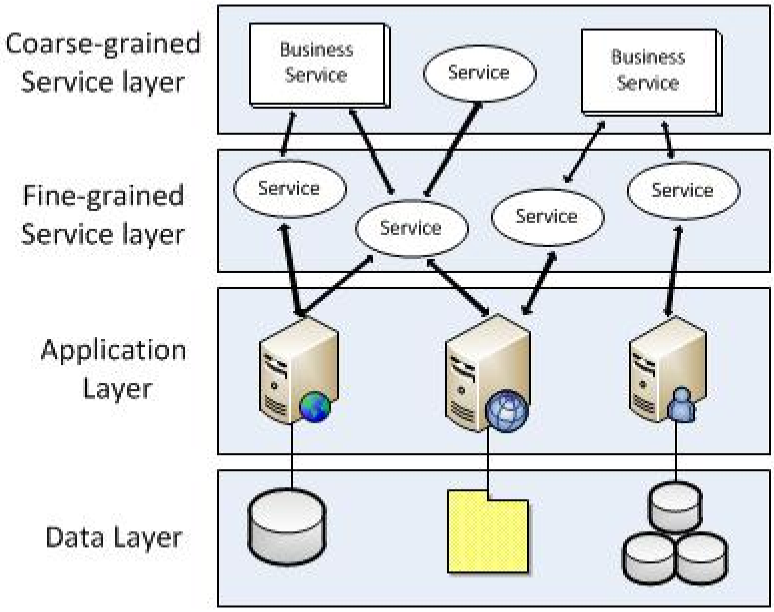


Figure 2-12: Web Services basic architecture

Companies that implement SOA aims to improve their cost-efficiency, agility, adaptability, leverage legacy investments, and interoperability [27]. However, the success of a SOA implementation depends on the strategy adopted. According to Lee, Shim, and Kim [56], companies that have successfully implemented SOA most often balanced factors which combine strategy, organization/management, technological infrastructure, project management, and governance. In addition, it is crucial to define the scope of SOA including the goals and vision combined with an incremental implementation.

2.8 SOA and PLM

PLM has emerged as a strategic approach for integrating systems and data throughout the whole product lifespan. SOA offers as a good means to implement PLM, because SOA provides the flexibility and the mechanisms for business integration as linked services, thus facilitating collaboration between participants during the product's lifecycle and permitting transparent data access from heterogeneous systems.

Figure 2-13 shows a legacy system in which different activities within an overall process access different independent databases. However, if a SOA approach is applied, these databases are decoupled from processes and applications, while at the same time permitting their integration [29].

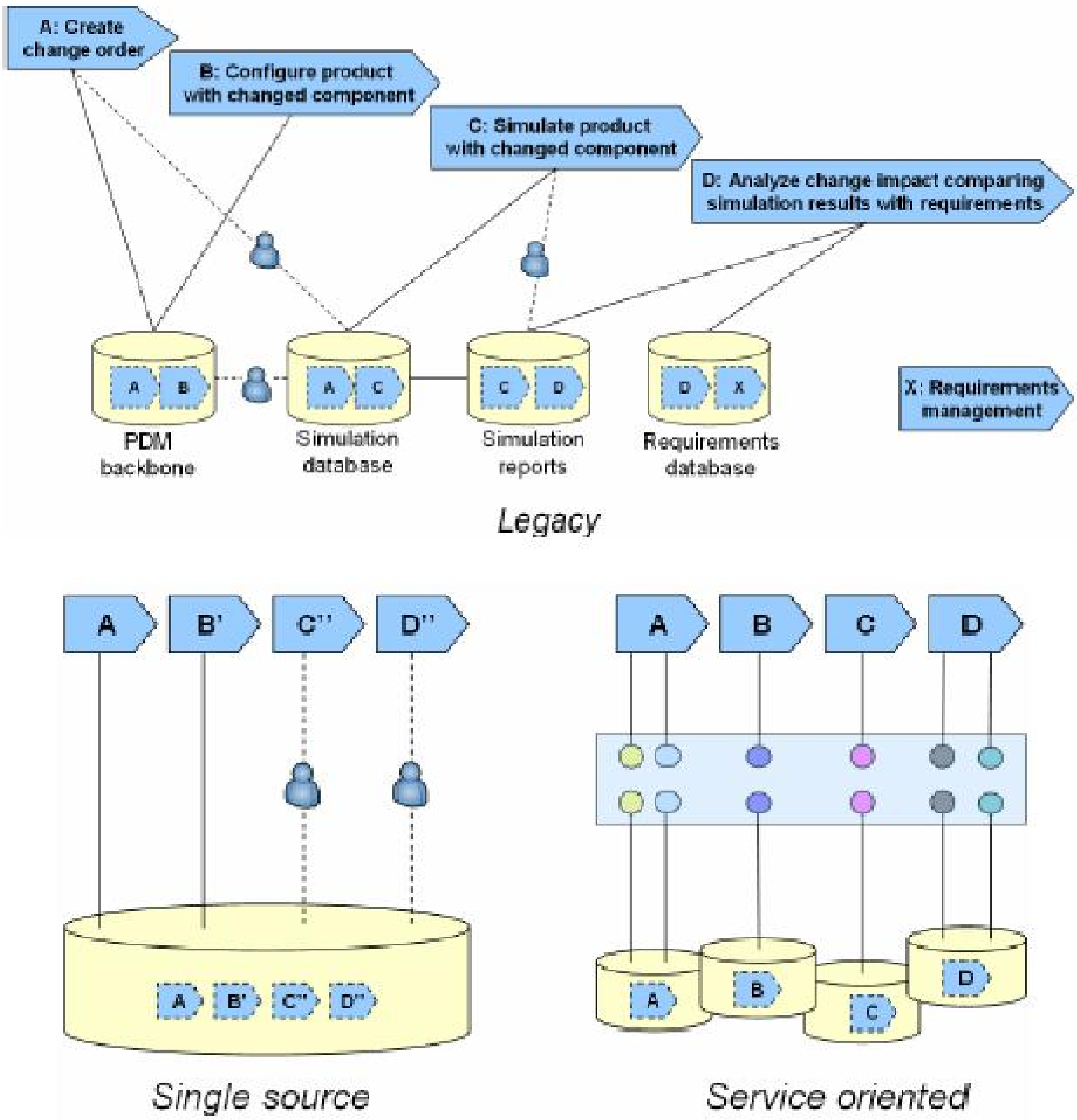


Figure 2-13: Interoperability of legacy systems through SOA

Figure 2-14 shows the SOA environment for PLM with the different layers that interact to provide the collaboration required for product development being provided as services that integrate different PLM applications (such as BOM, CAD, or PDM).

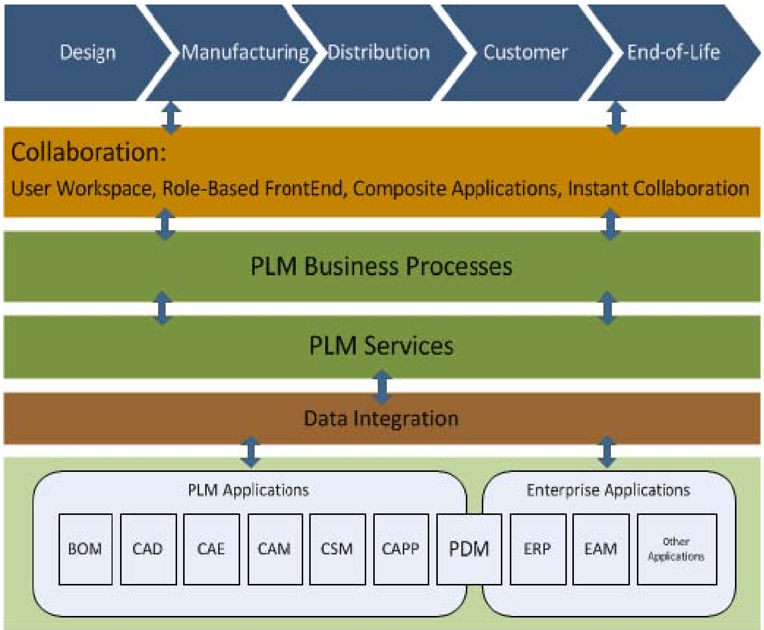


Figure 2-14: SOA for PLM environment [32]

There are several reasons for adopting SOA in an organization. For example, SOA enables agility by providing services that can be dynamically combined to address business requirements [30, 31]. SOA also promotes weak coupling between services which allows transparency and flexibility, in contrast to single source solutions and suppliers' suites [29]. Another advantage of implementing SOA in a company is that it permits the company to customize and standardize their IT environment due to the loose integration of different databases and systems which are presented to the user as a single storage unit [29]. Other benefits of SOA include multidisciplinary integration and collaboration which facilitate product innovation [30].

2.9 Previous Work

A lot of research has been done on PLM and collaborative product development. A recurrent topic is the interoperability across different heterogeneous systems when using SOA and web services. For example in [29], a service oriented PLM architecture is presented for decoupling business logic and processes from the application and data base layer and data. An alternative for flexible processes through a service-oriented architecture is proposed in [31]. A SOA based solution for enterprise integration is presented in [32], which includes integration of systems, processes, and data.

Information exchange among collaborators is a main concern because it facilitates the integration of the different applications and domains under the PLM scope. A framework for integrating multiple CAD and PDM systems converting CAD models to Engineering Bill of Material (EBOM) is presented in [22]. CAD model sharing has been extensively studied, as these models represent a significant part of the data that is shared in terms of volume and number. Parallel CAD data retrieval using web services is introduced in [33].

The use of common open standards can greatly facilitate information exchange and domain integration. Unfortunately, the adoption of open standards is still an issue due to incompatibility of the existent standards and the divergence of interests in the way standards should be defined [34]. STEP, described in section Standard for Product Data Exchange (STEP), facilitates storage and product data exchange, which can enable interoperability across heterogeneous systems [57s].

Although there has been extensive research on PLM and collaborative product development, some aspects require more study. In the current global context, the location of the participants can have a large impact, particularly when large amount of data (such as CAD models) needs to be transferred. For example, Li, Gao, and Wang [36] propose a real-time solution for integrating heterogeneous CAD systems using a client-server model *without* considering the effect of latency between the collaborators.

A heterogeneous PLM systems scenario requires an underlying mechanism that permits them to share data without sharing all of the contents of their own databases. This thesis project aims to propose an alternative mechanism for data sharing which considers the geographical location of the participants. In order to formulate such a proposal, the following related works have been studied to gain different perspectives.

First, we study a concurrency model for data exchange in PDM systems. Next, an algorithm for clustering based on latency is introduced. Next, we present a data exchange service for exchanging CAD designs. After this, the deployment of a distributed PLM for a global collaborative environment is presented. Finally, two commercial applications are presented, a multi-site collaboration feature of Siemens' TeamCenter and an enterprise open-source PLM platform: Aras Innovator.

2.9.1 A concurrency model for PDM systems

Chan and Yu [37] propose a concurrency control model to address the requirements for data distribution and integrity of distributed product development. They reviews three DBMS concurrent control techniques: two-phase locking, timestamp ordering, and flow graph locking, and make an analysis as of each of them as an alternative for PDM systems, concluding that they are not the ideal mechanisms, therefore they propose a different approach.

According to their paper, an optimistic scheme, such as timestamp ordering, has the drawback that some effort may be wasted. For example, if a CAD file is being edited simultaneously by 2 engineers, the one that submits their work later will have to redo this work. On the other hand, a pessimistic scheme would reduce the overall system's throughput by reducing the degree of concurrency.

Their proposal combines granulated locking with data version control, adopting three standard modes of lock (shared (S), exclusive (X), and intention shared (IS)) and two new modes: version (V) and intention version (IV). Figure 2-15 shows the application of locks. The vault stores a set of product projects, with each product project composed of assemblies and subassemblies. When these entities are accessed the IS lock is applied. If the assemblies are directly related to parts, then the IV lock is applied. The parts can utilize S, V, and X lock modes.

This control model has a simple lock mechanism. In order to read an entity, first the ancestor (Figure 2-15) is set to IS lock mode and the entity to S lock mode. To update an entity, the ancestor at level 0 is set to IV lock mode, and after this the entity is locked with the V lock mode.

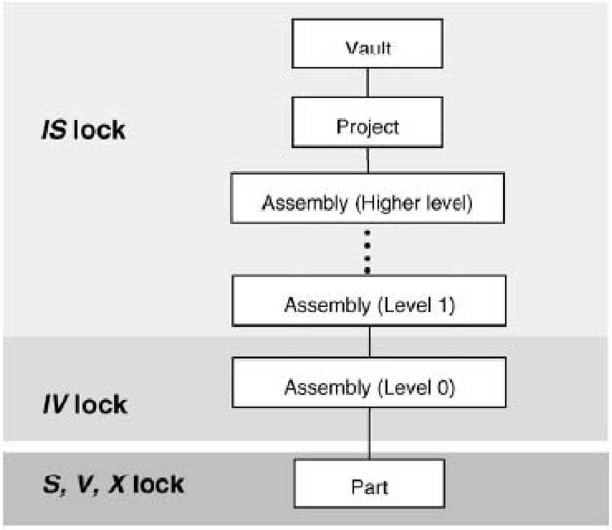


Figure 2-15: Lock granules precedence graph

2.9.2 Latency based dynamic grouping aware cloud scheduling

Malik, Huet, and Caromel[38] proposes an algorithm for grouping nodes based on the network latency in a multi-cloud environment. Their algorithm complexity is O(n^2), the input is the latency values between the nodes, and it requires to setting two configuration parameters: the maximum latency threshold between two node, and a group latency threshold. One important characteristic of this algorithm is that the groups generated are mutually exclusive. Another characteristic is that this algorithm can work with partial knowledge of the latency values.

2.9.3 Web Services for CAD (WSC)

Eugster, et al. [14] propose the WSC process and convert the CAD model data used in the product's lifecycle into the required form to support the integration of different CAD systems operated in collaborative product development. The specification of this WSC is defined using WSDL and the service requesters need to format their requests according to this service specification. When a requester asks for a specific CAD model, WSC retrieves the appropriate CAD file (which may use a proprietary CAD model) and converts this CAD file into an XML-based neutral model and return the results of the conversion to the requester. WSC does not deal with the processing of the XML-based neutral model at the requester.

WSC proposes an extensible and reusable multilayer architecture (Figure 2-16) suitable for integration in scenarios with multiple heterogeneous CAD systems. This is a good example of web services technology applied to PLM and collaborative product development that facilitates integration among different CAD systems.

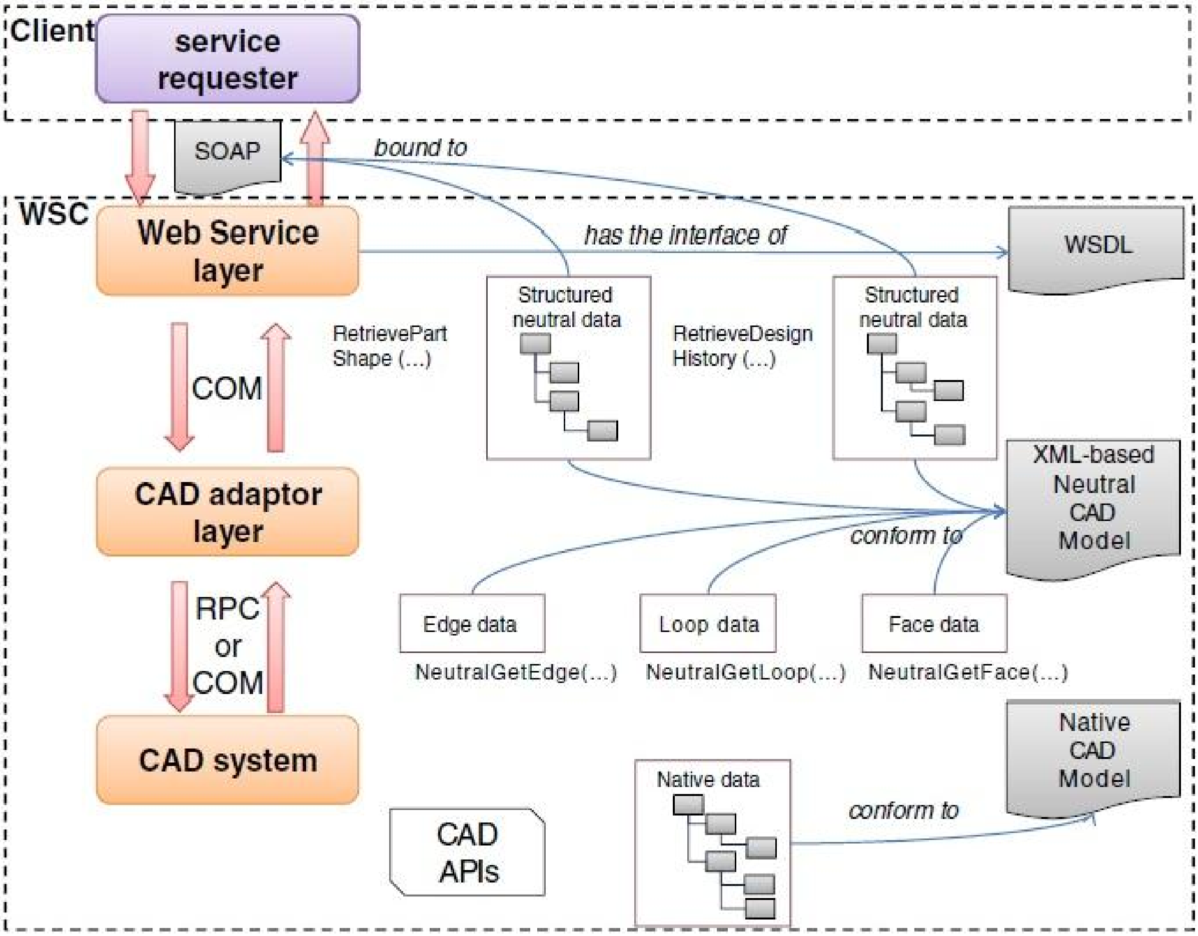


Figure 2-16: Web Services for CAD (WSC) architecture[14]

2.9.4 Distributed Deployment of a PLM system

A methodology to design the system architecture of a PLM platform for a globally distributed enterprise was proposed by Luh, Pan, and Chu in [39]. This methodology analyses the company from four hierarchical perspectives: organization, data, content, and applications views.

The organization view evaluates the geographical distribution of the locations of the company and their network topology. Additionally, the role of each site in the development process is identified, including their collaborative activities. This view gives an overview of both the business structure and the communications infrastructure of the company.

The data view evaluates the amount of data created at each side to determine if a local file vault is necessary or not. This evaluation involves calculating the required bandwidth to and from a remote data vault based upon an estimate of the volume of data to be transferred.

The content view focuses on the data exchanged at each of the company's locations, particularly of CAD files as these are usually the bulk of the data traffic. Moreover, is not always possible to increase the bandwidth at a site, hence, this study uses an approach to reduce the CAD geometry data (known as levels of detail (LOD) concept) [40].

Finally, the application view focuses on studying the usage of PLM functions at each site to determine the proper deployment of applications. It is also important to determine the maximum number of concurrent users in order to determine the performance requirements and the licensing requirements, and thus adjust the hardware and software deployment accordingly.

2.9.5 TeamCenter Multi-site Collaboration

TeamCenter Multi-site Collaboration [41] is a component of Siemens' TeamCenter PLM platform that enables integrated product development among sites and suppliers. It permits each local site to maintain its own data, its own users, and its group management, while sharing only selected information and defining who can access this data.

In order to provide supplier integration, TeamCenter provides two possible configurations: proxy and hub. The proxy configuration (Figure 2-17) permits suppliers to integrate without having direct access to the internal network but rather perform their accesses via a proxy server, which is installed *inside* the firewall of the company. The hub configuration utilizes an intermediate site between the site and the supplier. In this configuration, copies of objects to be shared are placed in a hub to which the supplier has access. The supplier can do the same if they want to share data with their own suppliers, thus creating a network as shown in Figure 2-18.

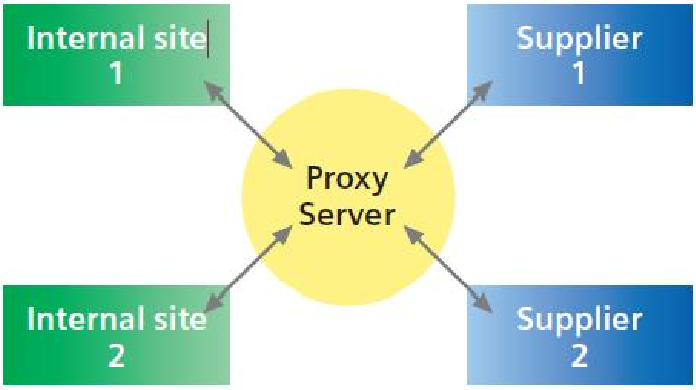


Figure 2-17: Proxy Configuration [41]

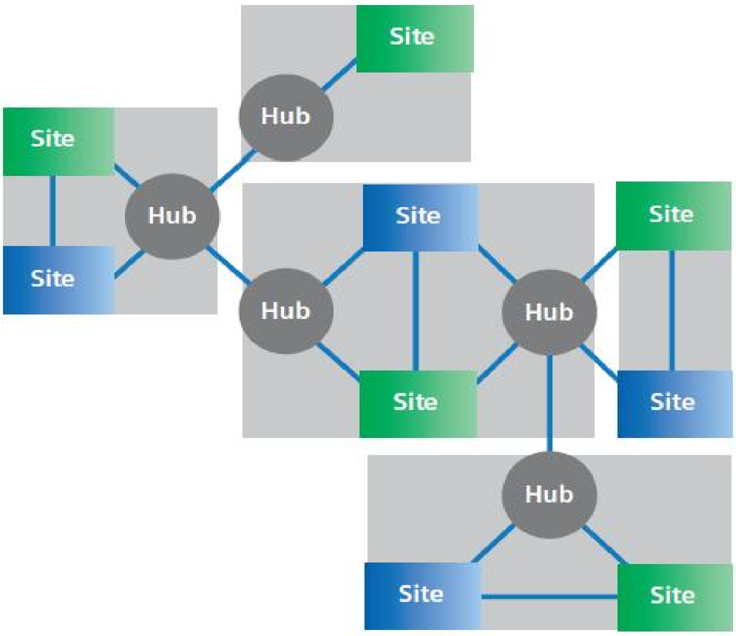


Figure 2-18: Hub Configuration [41]

A replication system allows the company's administrators to define access rules in order to avoid compromising security. This is necessary as preserving adequate security is an important aspect of this inter-company process. The data is replicated on a business level, rather than a raw-data level, which gives the administrators more control and is also more efficient since only the required data is replicated. The technology used for transferring large files such as CAD designs is the File management System (FMS [51]) which delivers a better performance than FTP [41].

2.9.6 Aras Innovator

Aras innovator is a model-based enterprise SOA solution that provides a comprehensive set of applications for PLM. The underlying technology of Aras uses meta-data templates to define each business object that is used for representing the information. This model-based approach is based on a run-time model which makes it possible to make changes in real-time. Additionally, the models are defined and stored as XML templates, which are easy to manipulate and integrate into the SOA Web Services in the Aras platform.

Aras Innovator is a web based, n-tier application built on top of Microsoft technologies. Its service oriented architecture (shown in Figure 2-19) is composed of a web client, application server(s), database(s), and file server(s) that are based on standards, such as HTTP/HTTPS, XML, and SOAP.

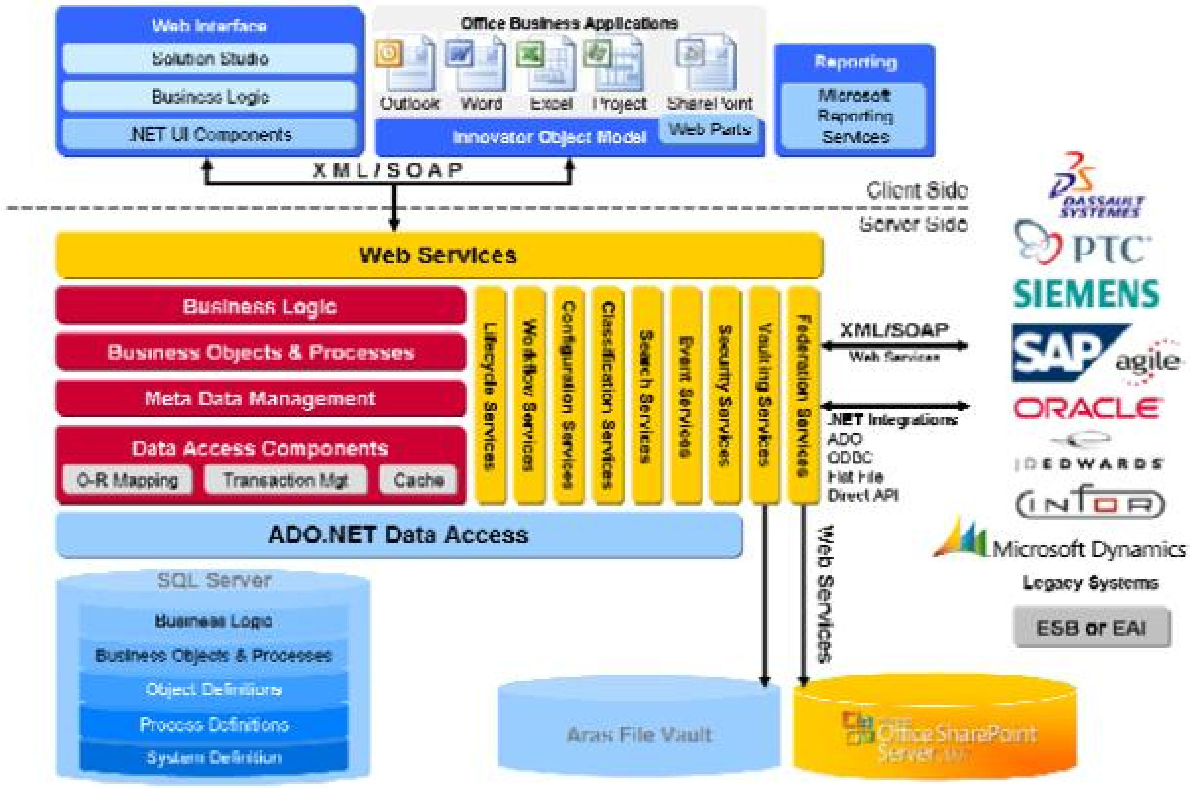


Figure 2-19: Aras Innovator enterprise application framework architecture.

The distribution model of Aras Innovator is based on an enterprise open-source model in contrast to licensed software. Aras solutions offer a process-oriented approach not only for PLM, but can also be used for enterprise quality management and global supplier management. In addition, Aras can be used as a complement of existing legacy ERP, PLM, or

PDM systems through open integration. Aras Innovator can easily be extended thanks to the SOA approach. Additionally, it is available as open source which facilitates customization according to users' needs.

3 Method

In this chapter we introduce the problem and the proposal developed in the context of this master's thesis project. First, a scenario to illustrate the problem is presented. Next, the proposed solution is described, including some details of its functions. The following two chapters will present details of the implementation of this proposed solution and an analysis of the proposed solution in a PLM context.

3.1 Scenario

A typical scenario that shows the benefits of collaborative work is the development of a complex product, such as an aircraft with millions of parts and components, involving the collaboration of several companies located in different countries. From a PLM perspective, such collaborative development can be challenging due to the heterogeneity of the PLM systems used (section PLM Challenges and Requirements). A collaborative product design implies parts and components being designed concurrently, and a permanent data exchange between collaborators (section Collaborative Product Development). For example, a company A is developing a part Pa1 composed of several components, and component Ca1, is also component of part Pb1 developed by company B, therefore, its design data is shared between both companies. The two companies have different PLM systems and any change to Ca1 that company A makes should be synchronized with the PLM system of company B. Although there are standards that facilitate data exchange between heterogeneous systems (section Standard for Product Data Exchange (STEP)), simply exchanging the data does not solve the problem of the consistency of the shared data. This scenario is shown in Figure 3-1.

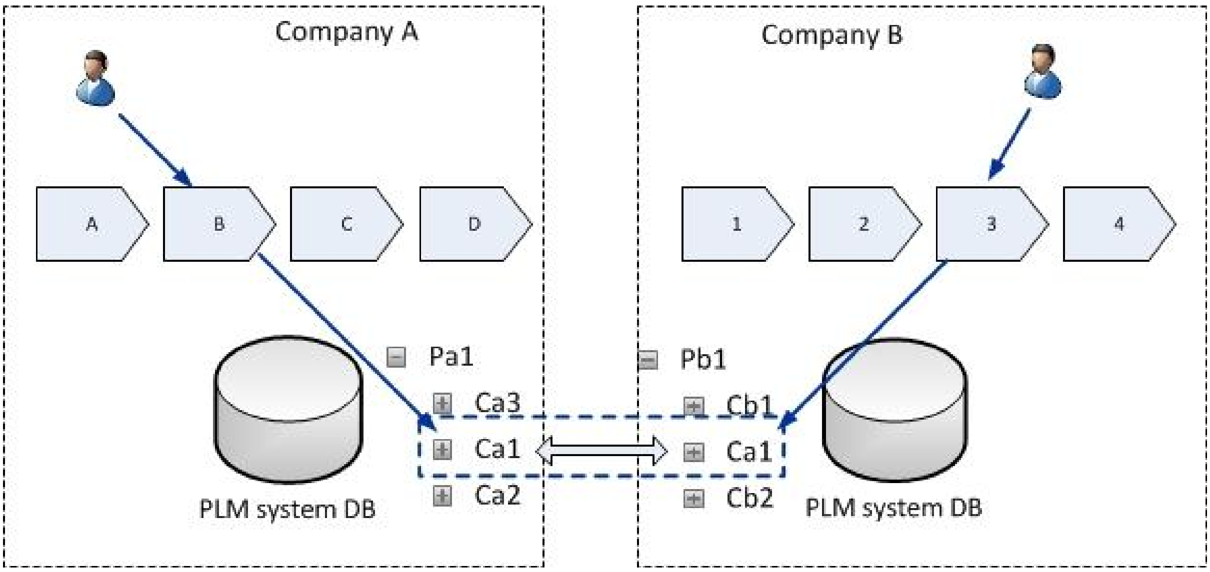


Figure 3-1: Scenario with two companies sharing product data.

Another aspect that can be relevant in a product design collaboration is the location of the different collaborators' sites and the network latency in the communication between these sites. For example, if one collaborator is located on the east coast of the US and the other in Shanghai, the network latency would be around 260 ms (according to measurements reported at [42]). This would limit the throughput of a TCP connection with a TCP receive window

(RWND) size of 64 KB to around 2 Mbps (based upon TCP Throughput = TCP RWND * 8 / RTT [50]). If large files such CAD models are constantly exchanged, network latency becomes an issue – note that this can be addressed by increasing the receive window size and using scaling options in TCP [52].

Figure 3-2 shows a distributed collaboration environment which requires a mechanism to exchange data between different PLM systems. This scenario also requires a concurrency control mechanism that avoids updates being lost, while maintaining data consistency. In order to address these requirements, a proposed solution is described in the next section.

Figure 3-2: Distributed collaborators of a complex project, such as an aircraft.

3.2 Design

This master's thesis project proposes an approach that aims to provide a coordination mechanism for sharing data in a PLM environment with dispersed collaborators. It also aims to minimize the impact of the network latency between distant collaborators' locations.

The solution proposes a mechanism by which collaborators are grouped with respect to their network latency. There are different peer networks that provide a communication mechanism, however, they are constructed without considering the network latency. The idea of grouping closer collaborators is to minimize this effect of latency [53].

Each collaborator is considered as a node in the system and each group elects a leader node which coordinates data exchange inside the group and also interacts with other groups. The concurrency control mechanism adopted is based on the one reviewed in section A concurrency model for PDM systems. In the current solution, the leader nodes act also as the brokers on the publish-subscribe overlay network.

Initially, the participants are grouped using the algorithm introduced in section Latency based dynamic grouping aware cloud scheduling. This algorithm minimizes the inter-node latency while dynamically producing mutually exclusive groups on O(n2). This algorithm also supports the use of partially complete latency information. However, in the current solution it is assumed that the latency between all pairs of sites is known. An example of this grouping is shown in Figure 3-3.

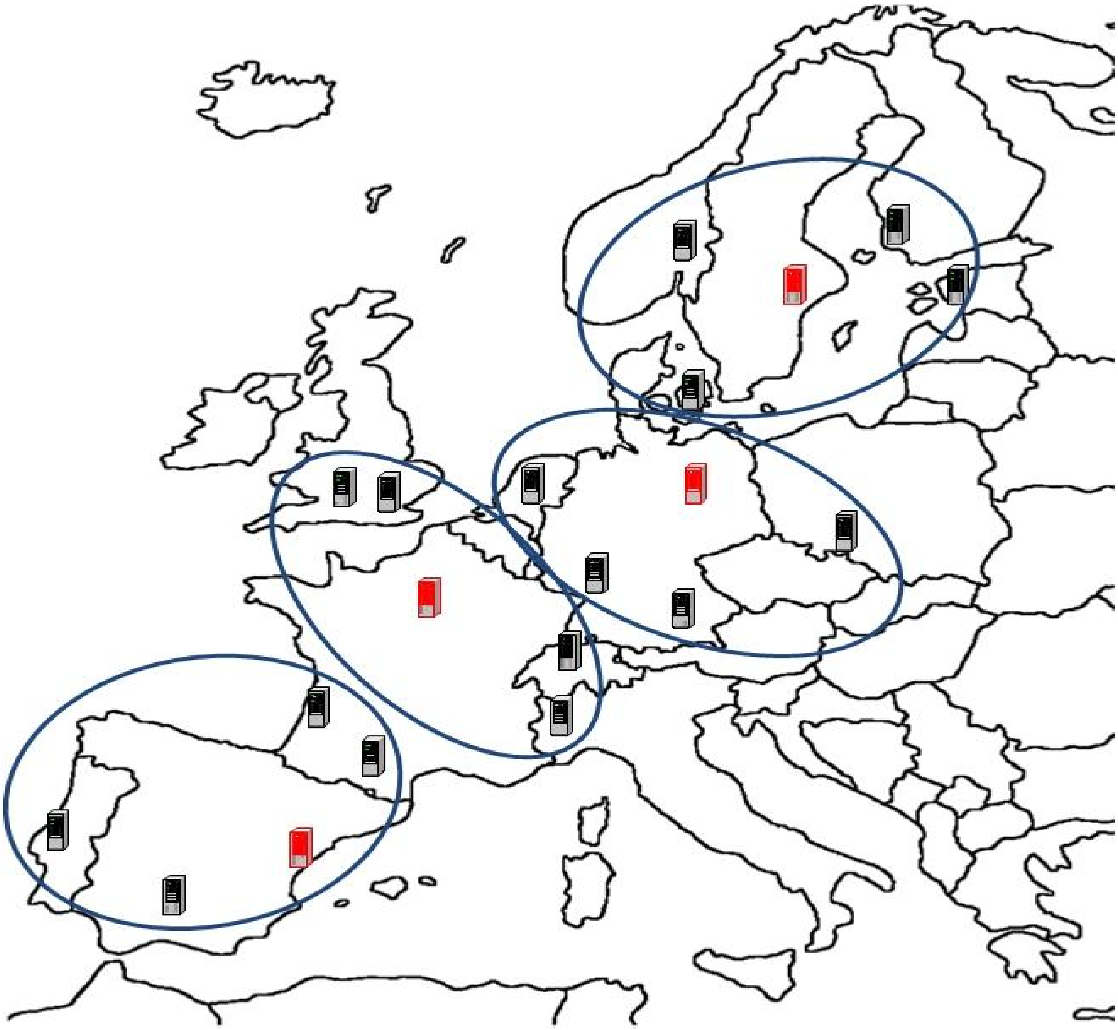


Figure 3-3: Collaborators arranged in groups with respect to network latency.

After the groups are created, each group *Gi* elects a leader node *Li*, which is the node with the lowest average latency to all of the members of this group. The leader coordinates the data exchange within the group and interacts with other groups' leaders. The aim of creating groups is to avoid a centralized solution with a single point of failure, to improve the overall performance in the system, and to provide scalability as groups can be sub-divided when the workload exceeds the leader's capacity. A leader in each group facilitates the concurrency control mechanism and communication between groups.

Each node maintains a local registry *R* of the resources that are shared, including the existing versions of each resource. This registry is only an index of the available shared data and does not include the resource itself, but only a reference to the resource. In addition to this registry, the leader keeps a reference to the nodes that have a local copy of a specific version of the resource.

In order to obtain a copy of a shared resource a node must contact the leader to learn the closest node with a local copy of the latest version. This mechanism works in the following way (see Figure 3-4):

1. The node $n_{i,j}$ requests from the leader L_i the resource.

2. The leader returns a reference to the node which has a local copy and is located close to the requester.

3. The requester obtains a copy of the resource from this node.

4. After obtaining the data, the requester informs the leader that it has a local copy of the resource.

5. The leader adds an entry in its registry which includes a reference to the node that just got a local copy of the resource.

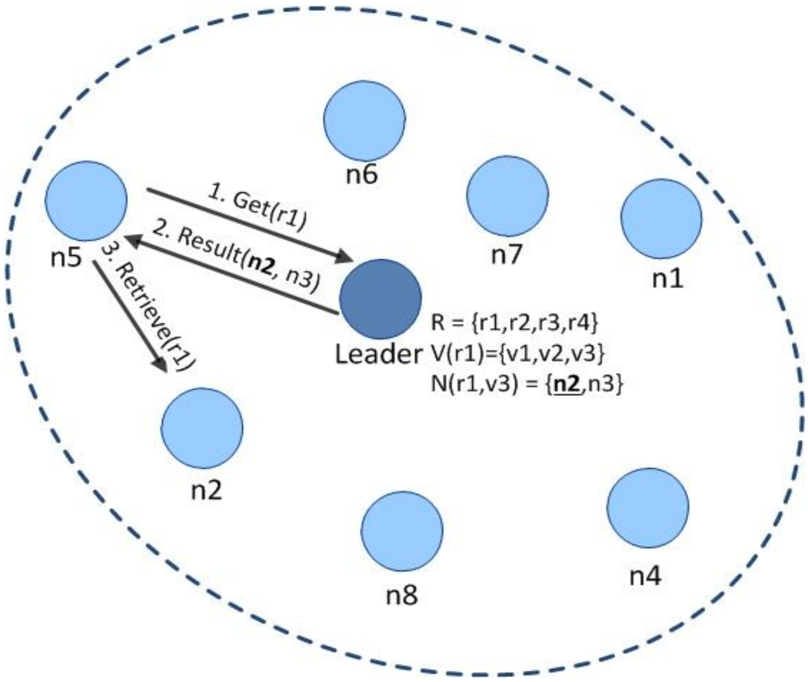


Figure 3-4: Data retrieval mechanism inside a group.

By redirecting the requester to a close node with a local copy of the resource, the data transmission task is distributed to other nodes rather than a central node as would occur in a centralized model. Additionally, by selecting the node with the minimum latency to the requester, the solution aims to minimize the latency in the system.

A node shares a new resource by notifying the leader of its group, which then sends a notification message to the other nodes in the group. After notifying the group's nodes, the leader forwards the new resource event. The events are transmitted to other nodes using a filtering-based routing mechanism using the broker network overlay Indirect Communication in Distributed Systems. This mechanism (Figure 3-5) is described as following:

1. The node n_{ij} contacts the leader L_i to share a new resource.

2. The leader L_i adds an entry to its registry and informs the members of group G_i that a new resource has been added.

3. The leader informs the neighbour leaders that a new resource has been added.

4. Each leader that was informed locally performs steps 2 and 3. If a leader has already been informed, it does not deliver the message again.

The concurrency control mechanism is based on the locking protocol described in section A concurrency model for PDM systems which provides a lock model with a version control. In this protocol the leader of each group coordinates access to the local resources and keeps track of the lock state of each shared resource accessed by the group. This mechanism allows concurrent data reads, while avoiding concurrent data edits. If the leader receives an edit permit request on a resource that is currently being edited, it queues the request until the resource is available.

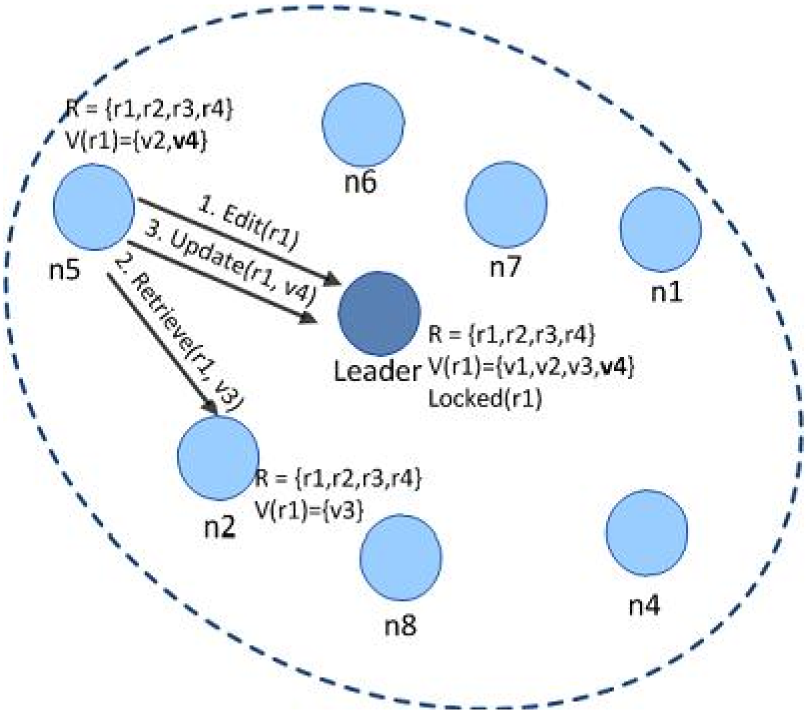


Figure 3-5: Resouce editing with lock control

The communication mechanism is based on a decentralized publish-subscribe messaging pattern, as described in section Publish-subscribe systems using a broker network overlay. For the construction of the broker network, each leader node will act as a broker, mediating between publishers and subscribers from different groups. Whenever an event occurs (indicating a change to the shared data), this event will be published throughout the network. Figure 3-6 shows how the leaders act as brokers following the publish-subscribe pattern.

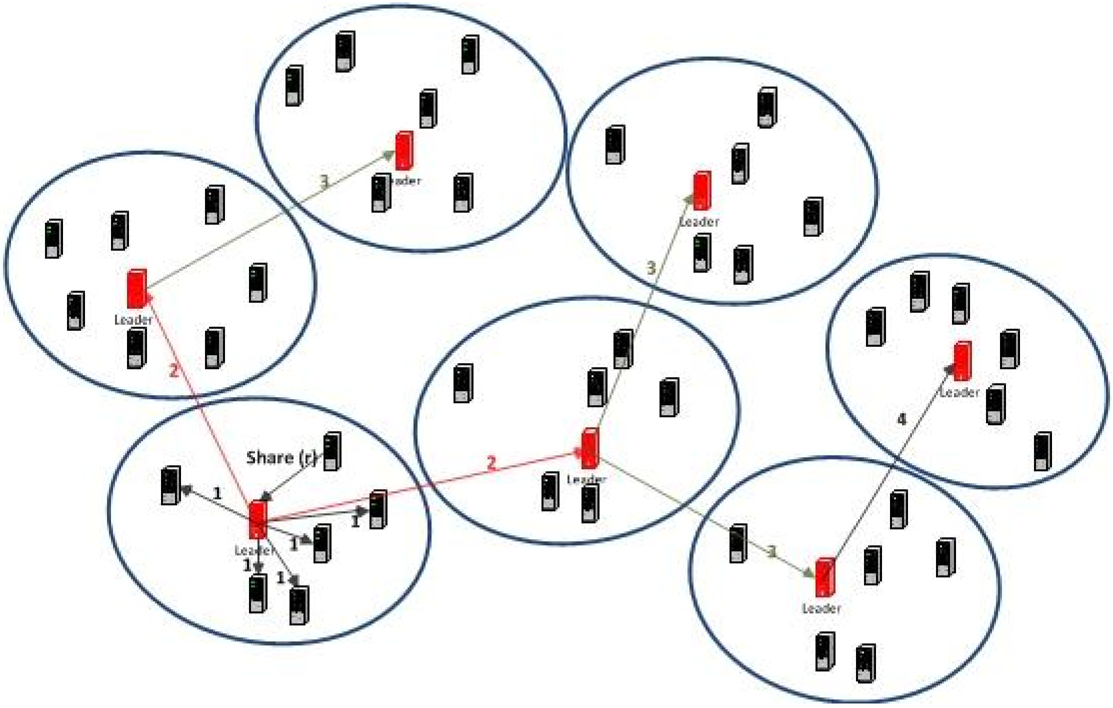


Figure 3-6: Advertising mechanism informing that a resource r has been shared.

The solution presented in this section provides a communication network which is constructed based on network latency in contrast to other approaches (such as that described in section TeamCenter Multi-site Collaboration). Another aspect considered is that the set of collaborators is not open, hence a new collaborator cannot spontaneously join the network, but rather a new collaborator should be added by the administrator responsible for the project. Consequently, groups are semi static, i.e., groups only change after an action by the human administrator responsible for the project. This slow rate of change in collaborators (and hence in groups) makes the network more stable and more controllable.

The publish-subscribe mechanism is well suited to the characteristics of PLM since it is an event based communication mechanism and any change related to a process, the data being shared, or other actions can be treated as an event. For example, if two collaborators are working together on a specific process, both can subscribe to events that would be generated on a topic that corresponds to that process. Another example would be when a component is re-used in multiple parts designed by different collaborators. In this case, all the collaborators can subscribe to an event related to information about that part.

4 Collaboration Module Design

In this chapter we present a solution based on the proposal presented in the previous chapter. The solution is implemented as a module that can be deployed as a separated application and implements services that permit its integration into different PLM systems.

4.1 Components

The solution is based on two main components: *Administration* and *Collaboration* modules. Each of these is described in a subsection below.

4.1.1 Administration module

The *Administration module* enables the creation and administration of collaborative groups and facilitates its own deployment on the collaborators' nodes. This component is used by the owner of the project to define the participants and the nodes' configurations.

In order to develop a functional prototype of this administrative component, the following requirements were identified:

Basic data

- Create, edit, and delete projects
- Add, edit, and remove collaborators

Group administration

- Generate groups
- Move a collaborator to a different group
- Connect and disconnect groups
- Deploy projects

The class diagram of the basic classes of the *Administration component* is presented in Figure 4-1. It can be seen from this figure that the solution can handle multiple projects.

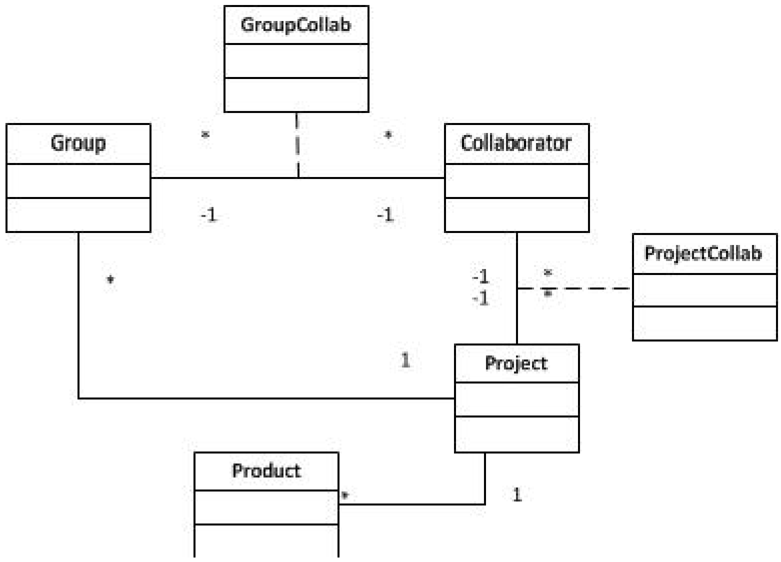


Figure 4-1: Collab_Administrator component class diagram

A graphic tool was considered as part of the implementation of this solution in order to provide a convenient way to access to the features listed above and to facilitate the deployment of the software in the network by setting up all the configurations and the connections between the nodes. Figure 4-2 presents a diagram of the user interface (UI) classes that represent the elements of the groups.

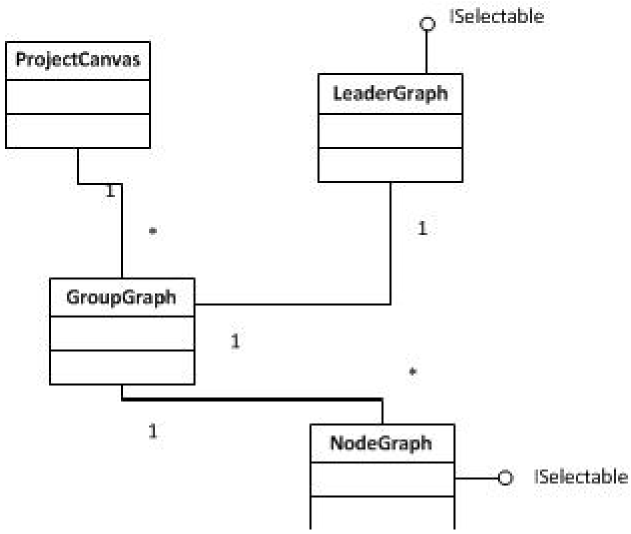


Figure 4-2: Collab_Administrator UI class diagram

The *Administration module* also provides mechanisms for deploying the generated groups. As a requirement for the deployment, the collaborators should have deployed and be running the web services of the collaborator module, which is described in the next subsection. Figure 4-3 presents a flow diagram of the deployment process.

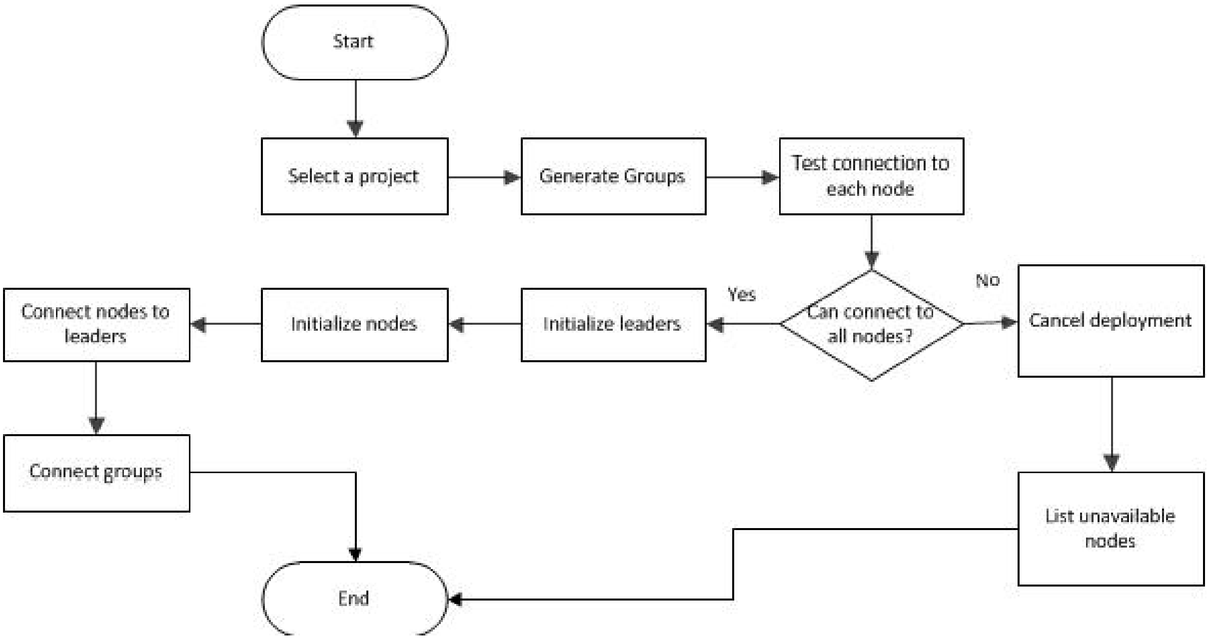


Figure 4-3: Project deployment flow diagram

4.1.2 Collaboration module

The second component is the *Collaboration module* which implements all the logic for sharing data, including the coordination mechanisms and all the functionality that makes communication possible between the collaborators. The requirements identified for implementing the prototype of this component are:

Collaboration service operations

- Share a resource
- Get a copy of a resource
- Subscribe to changes of a resource
- Request to edit a resource
- List resources

Administration service operations

- Test a connection
- Initialize configuration of a node or leader
- Connect to a leader
- Disconnect from a leader

The *Collaborator* class implements the services defined in four different service contracts.

IMessageService	defines the basic functionality for sending and receiving messages.
IApplicationService	defines all the necessary methods for sharing, updating, and editing the resources.
IResolverService	defines methods that permit each node to register and get assigned a unique ID used in the overlay network. In addition, this service permits to obtain the location of a node based on the unique ID.
IDeployService	defines the methods for configuring and starting the application services which can be called from the administration console.

The classes for implementing the Collaborator module are presented in Figure 4-4.

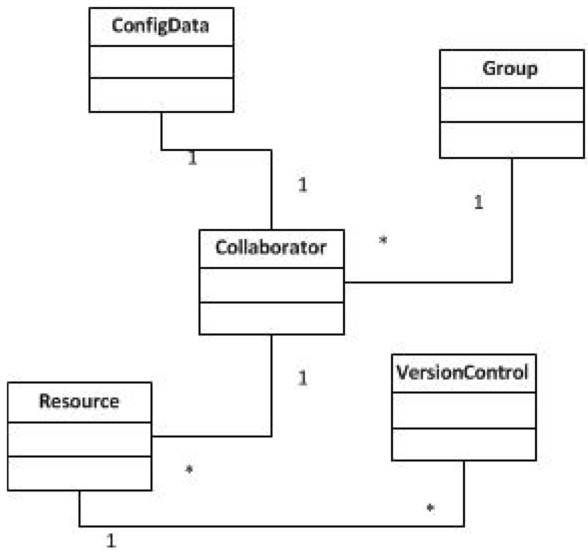


Figure 4-4: Class diagram of the Collaborator module.

The publish-subscribe model permits the nodes to inform participants about events taking place within the system. These events are generated whenever a new resource is shared, a resource is updated, or a resource is removed. For example, if a collaborator shares a new resource, this becomes an event and the collaborator acts as a publisher to publish this event. Whenever a node requests a copy of a resource, the node subscribes to the topic that corresponds to that resource and will be informed of all the subsequent events related to that resource. Figure 4-5 describes this communication model applied to the proposed solution.

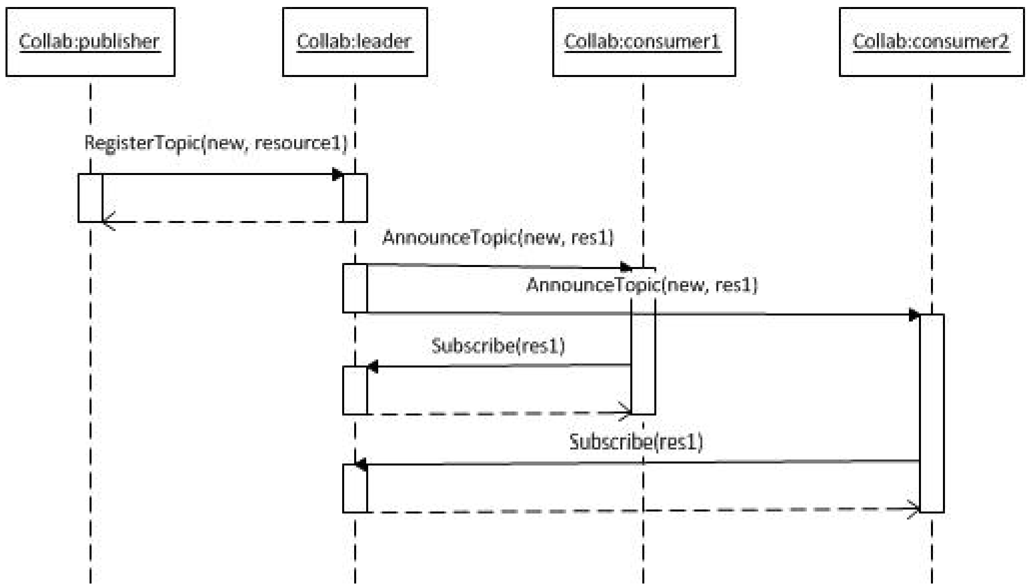


Figure 4-5: Publishing resource sequence diagram

The mechanism for version control and the concurrency model was presented in section A concurrency model for PDM systems. This mechanism provides granularity and a simple lock mechanism that can be implemented by the Collaborator. Figure 4-6 presents a sequence diagram when editing a resource.

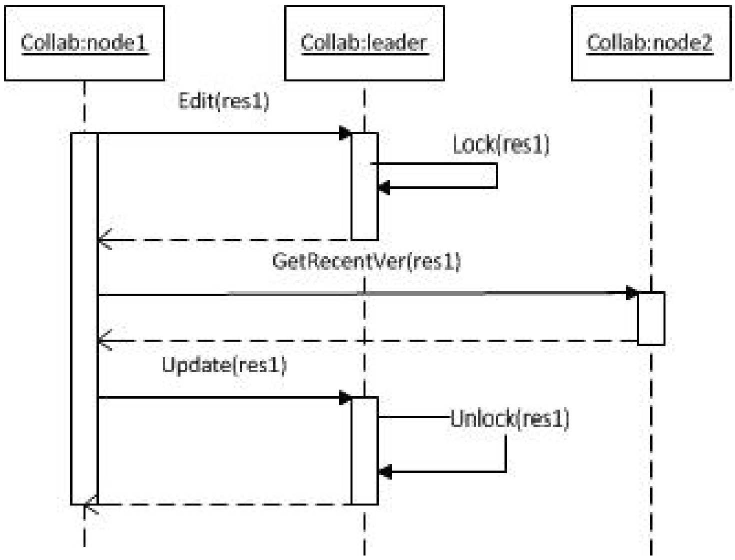


Figure 4-6: Edit resource sequence

As mentioned in the previous chapter, each node keeps a local copy of the resources it requires. The resources are files maintained locally at the server and the node maintains a list of the descriptions of the resources. Additionally, the leader node keeps a registry of the existing local copies in order to locate nodes with local copies when a collaborator requests a copy of a specific resource.

4.2 Deployment

The solution is deployed using the *Collaboration module* which interacts with the collaborator nodes through web services (see Figure 4-7). Each participant deploys a collaboration site which is accessible by the other collaborators. This configuration permits a collaborator to share only the data related to the project, thus it avoids exposing other data of this participant.

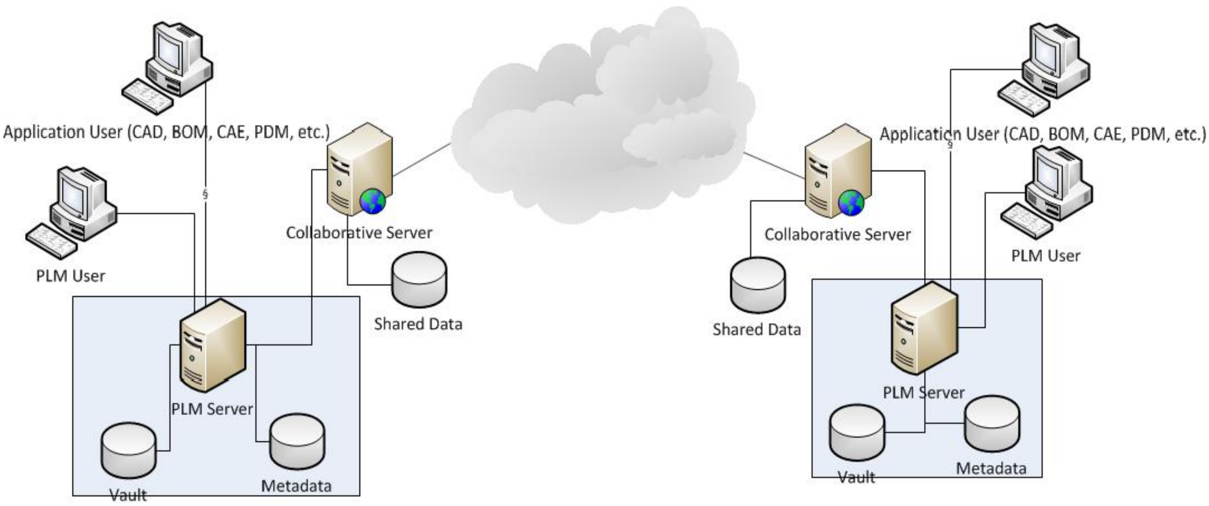


Figure 4-7: Deployment of the collaborative module in a PLM system connecting 2 participants.

4.3 Evaluation and Limits

In the project, a prototype version of the module described in this chapter has been implemented. This prototype will be used for evaluating the approach proposed. For the evaluation of the proposed solution, a distributed collaborative scenario is considered and a simulator is implemented. A simulation was considered a better alternative for the current project compared to an emulation since larger number of events can be generated in shorter time compared to an emulation.

The simulator uses the methods defined in section Administration module and consumes the services defined in section Collaboration module. These services permit to create the groups, the broker overlay network, and the data generation (sharing, subscribing, and updating events).

The evaluation will focus on measuring the latency along the events generation and propagation in the scenario where collaborators exchange information more often among members closer located. The results will be compared with current to a traditional centralized model. The objective of this experiment is to validate the proposed solution which involves organizing collaborators with respect to intra-latency and using a communication protocol based on publish-subscribe as a better approach to mitigate latency compared to a centralized model.

As a result of this evaluation, it will be determined whether or not this approach is feasible for enabling a multi-site collaboration in a global PLM solution. The evaluation provides also information for comparing the proposed solution with a centralized approach. Details of this evaluation will be given in section Evaluation.

5 Results

A prototype was implemented based on the approach proposed in chapter 4. This prototype was implemented using Microsoft's .NET technologies and Windows Communication Foundation (WCF) [43] which provide a complete development platform. The administration module has been implemented as a desktop application. Figure 5-1 presents the main window of the administration tool. Note that in this example, the nodes for the two projects are all running on the same physical computer.

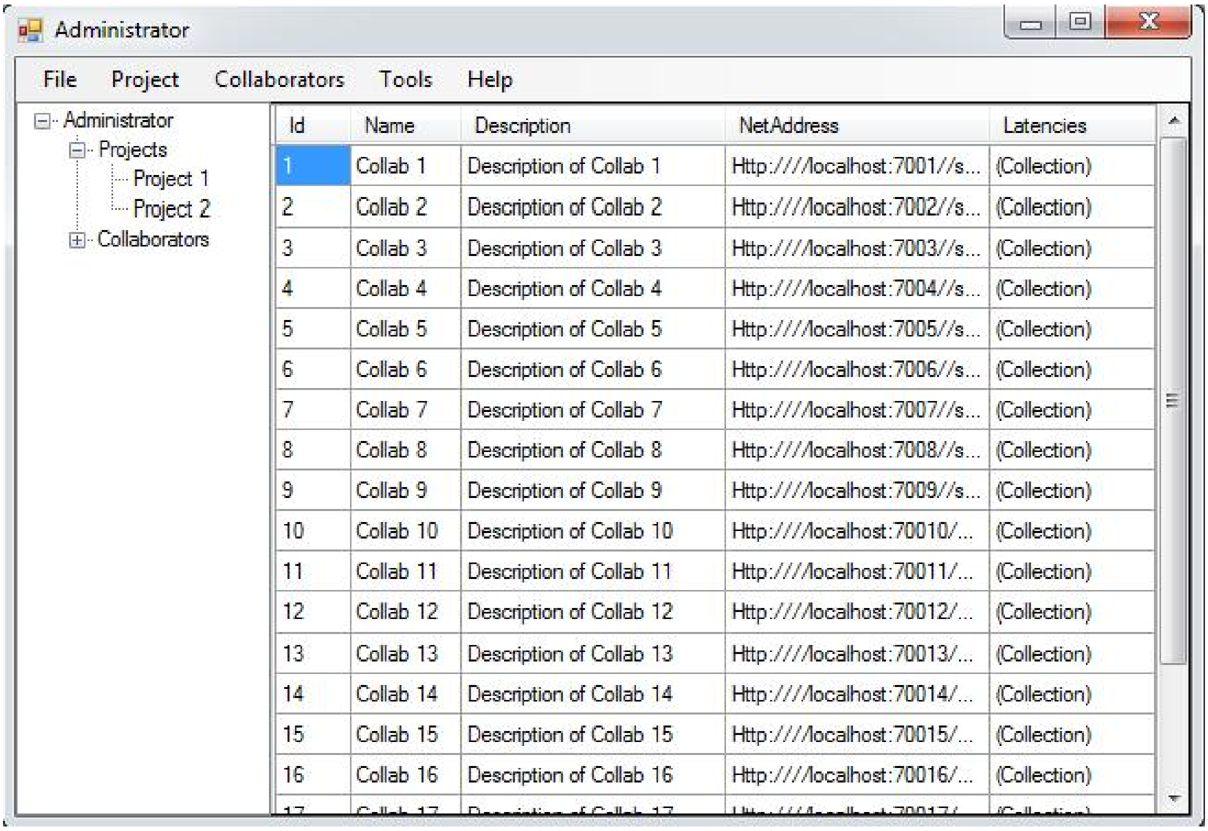


Figure 5-1: Main window of the CollabAdmin tool

One of the main features of the administrator tool is that it provides a visual interface which presents the network as a diagram that can be modified. The tool can generate the groups according to the selected algorithm, modify these groups (adding or removing members), and modifying the connections between groups. Figure 5-2 shows this diagram and with a set of options that permits an administrator to manually add or remove connections between groups and to move nodes to different groups.

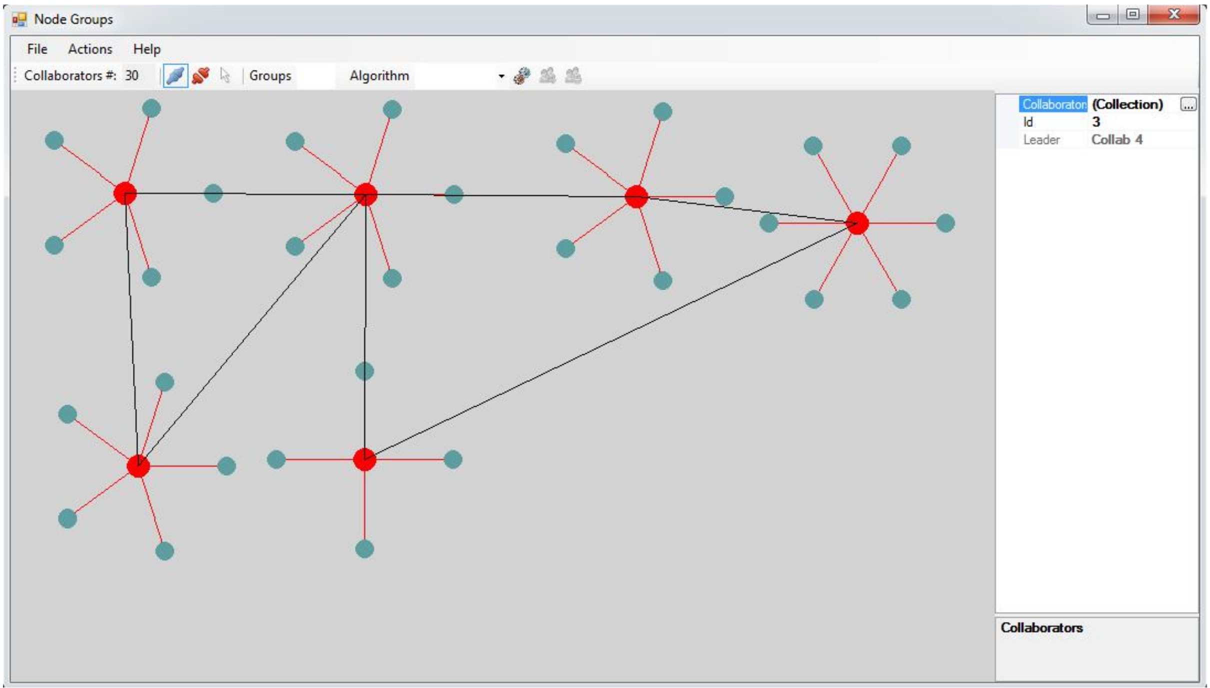


Figure 5-2: Network diagram generated by the tool showing the groups and group leaders of an example collaboration (here collaboration 4)

The deployment of the resulting collaborators network can be done using the tool (see Figure 5-3). As requirement, collaborator nodes must be running the Collaborator module on a server with Microsoft's Internet Information Services (IIS). The Administration module connects to the Collaborator instances through web services.

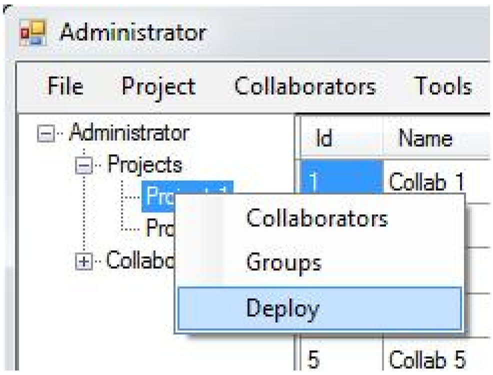
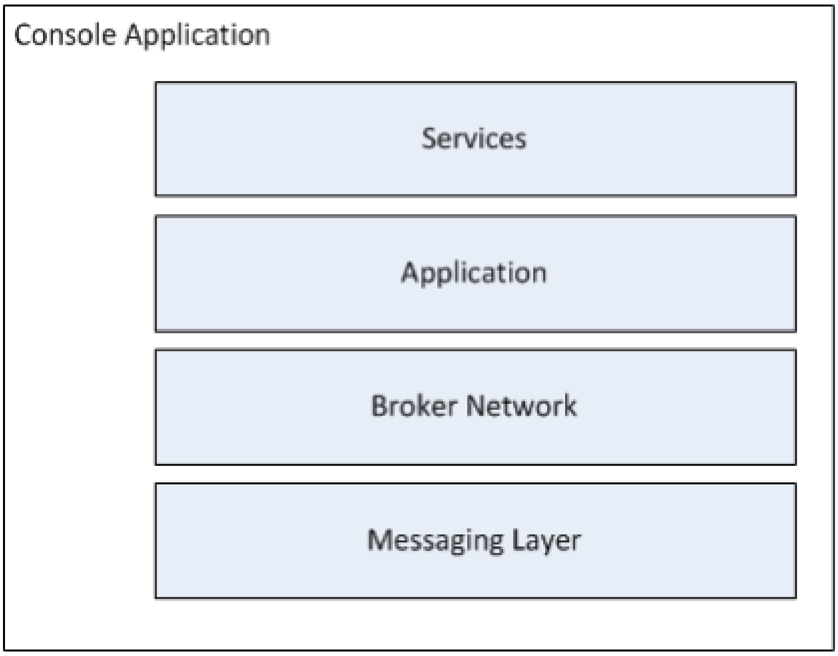


Figure 5-3: Deployment option of the administration tool.

5.1 Collaboration implementation

The implementation of the collaboration module is based on a console application that deploys the services that enable the collaboration based on the publish-subscribe pattern. This console application wraps the classes that implement all the logic for the collaboration module. Figure 5-4 shows the architecture of the application.



5-4: Application architecture.

The current implementation is based on components that are connected through events. The messaging layer implements the *IMessageService* interface using WCF. In case a message is delivered, the messaging layer throws an event *MessageReceived.* The broker handles the *MessageReceived* and according to the routing mechanism implemented at the leaders and throws an event that is handled by the application layer.

WCF supports SOAP and provides interoperability with standards such as REST or JSON. Another reason for utilizing WCF is that the Aras Innovator platform is built on top of Microsoft technologies, therefore, use of WCF facilitates integration with the existing platform. In addition, it provides security features for enabling a secure message exchange. Table 5-1 present the requirements identified on a distributed collaborative PLM and the correspondent alternative proposed in the current project to address them.

5-1. Table: Requirements of a distributed collaborative PLM that the current solution satisfies.

Distributed collaborative PLM	**Solution**
Mitigation of latency	Group collaborators based on their inter-site latency
Data access control and consistency	Concurrency model
Scalability	Publish-subscribe protocol
Protection of proprietary information	Independent deployment with a separated database
Integration with other PLM systems	SOA and web services

5.2 Evaluation

In order to evaluate the current solution, a simulation environment was set up. This simulation environment runs on a single machine with multiple instances of the collaboration service. A data collector has been implemented and attached to each instance, collecting information from the simulation.

The simulation environment was set up considering some characteristics of the development of a complex project, taking as an example the Airbus a380. This product was developed based on a modularity concept in which very specialized partners around Europe designed and developed components and subunits. Each member country of the Airbus consortium was in charge of one of the structural subsystems of the aircraft (fuselage, wing, cabin, or tail) [44]. Each subsystem was developed using the consortium member's own supplier networks and collaborators. Data exchange during the parallel development of subsystems is very important since there are several dependencies between the various subsystems. Although a majority of data exchange occurs between partners within a single country, there is also data exchanged between collaborators located in different countries. This type of project is becoming more common, as specialized companies develop specific components that subsequently are assembled to produce a complete product.

In the current simulation 5 different regions are defined (figure 5-5), each region being in charge of the development of a subsystem. Every region has 6 collaborators, making a total number of 30 collaborators. For example, the European Aerospace Cluster Partnership (EACP) has 41 members [47].

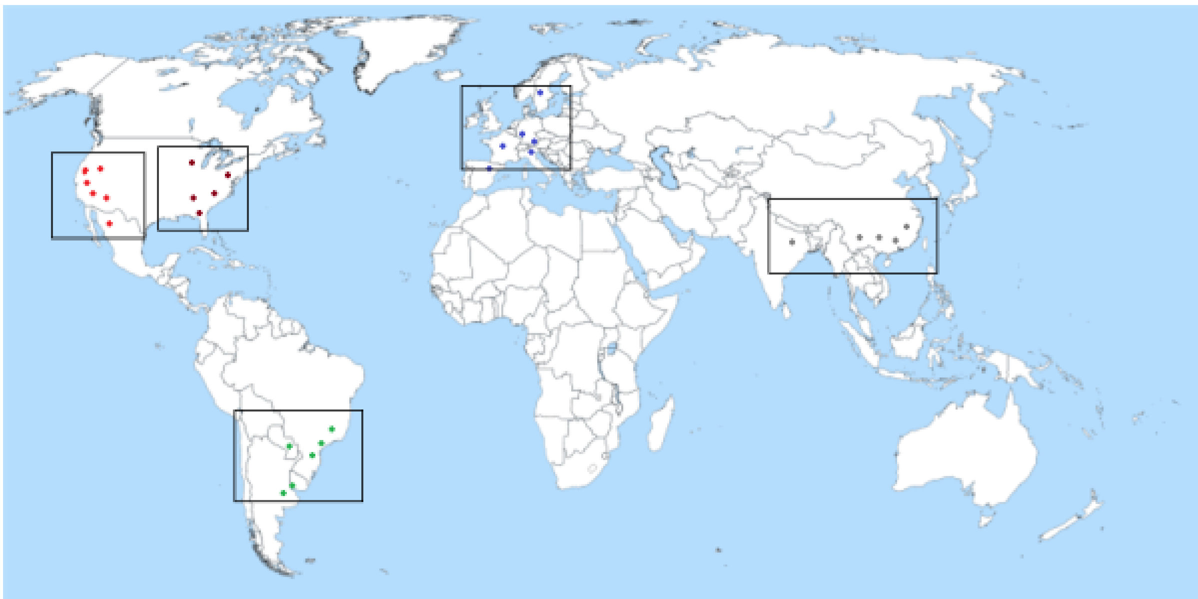


Figure 5-5: Silmulation environment. Region 1 and 2 are located in North America. Region 3 is located in South America., region 4 in Europe and region 5 in Asia.

The simulations were executed on 2 different scenarios. The first scenario presents a centralized system, in which all the clients connect to a single server which keeps track of the subscriptions. The second scenario implements the grouping algorithm presented in section Latency based dynamic grouping aware cloud scheduling. Figure 5-6 shows a centralized system compared to a group based.

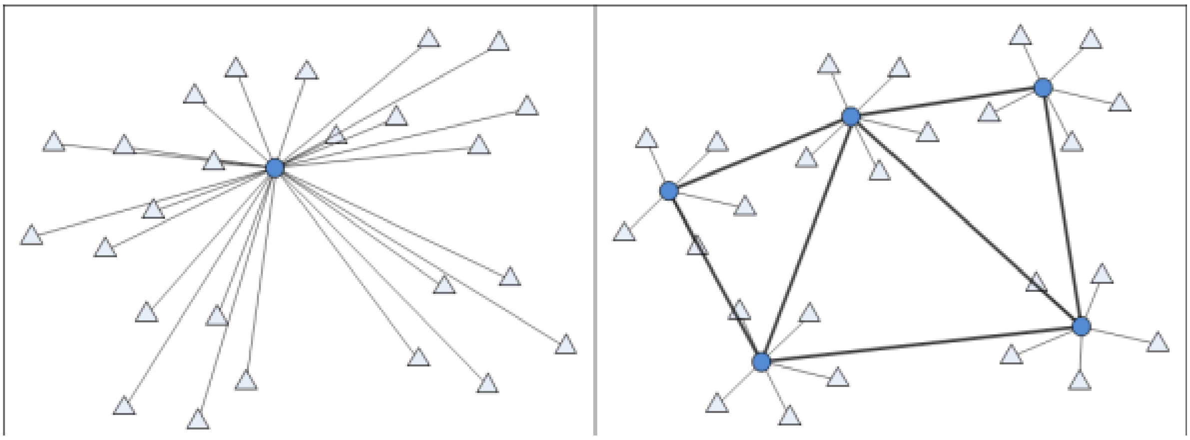


Figure 5-6: On the left the scenario 1 showing a centralized architecture. On the right the scenario 2, with a group based and a broker network that connects groups.

Each participant shares a file and is automatically subscribed to the topic that corresponds to this file. Additionally, every participant is subscribed to all the files shared by collaborators from the same region as it is assumed they work close together to develop the assigned subsystem. In addition, it is considered that some collaborators require resources from other regions.

5-2. Table: Simulation scenario configuration.

	Region 1	Region 2	Region 3	Region 4	Region 5
Clients	c1, c2, c3, c4, c5, c6	c7, c8, c9, c10, c11, c12	c13, c14, c15, c16, c17, c18	c19, c20, c21, c22, c23, c24	C25, c26, c27, c28, c29, c30
Files shared	f1, f2, f3, f4, f5, f6	f7, f8, f9, f10, f11, f12	f13, f14, f15, f16, f17, f18	f19, f20, f21, f22, f23, f24	F25, f26, f27, f28, f29, f30
Additional Subscriptions	(c2,f14), (c5,f21)	(c7,f17), (c10,f5)	(c13,f2), (c15,f22)	(c19,f3), (c20,f9)	(c25,f12), (c29,f7)

The latency between participants is calculated based on the distance and a factor that increases according to the distance. The latency values used for the simulation is presented on table 5-3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	40	106	296	421	13	6	10	6	15	45	40	57	19	58	104	125	114	104	118	336	234	222	234	213	401	470	399	462	393
2	40	0	63	194	351	18	24	25	48	54	9	3	15	19	16	68	72	59	55	77	228	145	135	145	128	331	401	330	392	323
3	106	63	0	201	322	101	103	70	110	130	50	67	48	59	44	10	15	19	10	12	217	147	136	144	145	300	369	303	365	295
4	296	194	201	0	102	228	281	285	310	315	192	193	145	228	147	223	148	136	181	221	23	12	19	15	21	74	136	69	128	68
5	421	351	322	102	0	395	408	406	434	445	343	352	322	385	322	346	288	283	304	336	52	110	118	109	136	9	21	9	18	12
6	13	18	101	228	395	**0**	**6**	**16**	**20**	**21**	25	18	44	18	46	103	116	103	76	116	308	210	198	210	187	375	444	373	436	367
7	6	24	103	281	408	**6**	**0**	**12**	**13**	**17**	29	24	50	17	52	102	120	108	79	116	322	222	210	222	200	388	457	386	449	380
8	10	25	70	285	406	**16**	**12**	**0**	**13**	**25**	27	26	48	9	48	68	108	78	69	100	322	224	211	223	205	385	455	385	448	378
9	6	48	110	310	434	**20**	**13**	**13**	**0**	**15**	52	48	64	22	65	106	130	120	109	120	350	287	234	287	225	414	484	413	476	406
10	15	54	130	315	445	**21**	**17**	**25**	**15**	**0**	62	53	73	45	75	129	149	137	128	143	358	294	280	294	228	425	494	423	486	417
11	45	9	50	192	343	25	29	27	52	62	**0**	**12**	**9**	**18**	**9**	56	60	48	43	65	222	142	131	141	128	322	392	322	385	315
12	40	3	67	193	352	18	24	26	48	53	**12**	**0**	**16**	**21**	**19**	72	75	62	59	101	228	145	135	145	127	332	401	330	393	324
13	57	15	48	145	322	44	50	48	64	73	**9**	**16**	**0**	**27**	**3**	56	53	40	28	64	204	127	116	126	114	301	371	301	364	294
14	19	19	59	228	385	18	17	9	22	45	**18**	**21**	**27**	**0**	**27**	59	74	66	57	69	302	207	194	206	189	364	434	364	427	357
15	58	16	44	147	322	46	52	48	65	75	**9**	**19**	**3**	**27**	**0**	52	50	27	25	60	205	128	117	127	116	301	371	301	364	294
16	104	68	10	223	346	103	102	68	106	129	56	72	56	59	52	**0**	**24**	**40**	**21**	**8**	238	198	184	194	195	324	392	327	388	319
17	125	72	15	148	288	116	120	108	130	149	60	75	53	74	50	**24**	**0**	**12**	**12**	**21**	190	127	116	124	129	228	334	231	331	224
18	114	59	19	136	283	103	108	78	120	137	48	62	40	66	27	**40**	**12**	**0**	**9**	**40**	180	115	104	112	113	224	330	225	325	218
19	104	55	10	181	304	76	79	69	109	128	43	59	28	57	25	**21**	**12**	**9**	**0**	**22**	199	130	119	127	127	283	352	285	347	237
20	118	77	12	221	336	116	116	100	120	143	65	101	64	69	60	**8**	**21**	**40**	**22**	**0**	233	195	182	192	195	314	381	318	378	309
21	336	228	217	23	52	308	322	322	350	358	222	228	204	302	205	238	190	180	199	233	**0**	**27**	**44**	**27**	**57**	43	79	29	73	28
22	234	145	147	12	110	210	222	224	287	294	142	145	127	207	128	198	127	115	130	195	**27**	**0**	**6**	**3**	**16**	78	145	75	138	73
23	222	135	136	19	118	198	210	211	234	280	131	135	116	194	117	184	116	104	119	182	**44**	**6**	**0**	**6**	**16**	105	184	103	147	79
24	234	145	144	15	109	210	222	223	287	294	141	145	126	206	127	194	124	112	127	192	**27**	**3**	**6**	**0**	**19**	77	144	74	137	72
25	213	128	145	21	136	187	200	205	225	228	128	127	114	189	116	195	129	113	127	195	**57**	**16**	**16**	**19**	**0**	125	205	120	195	118
26	401	331	300	74	9	375	388	385	414	425	322	332	301	364	301	324	228	224	283	314	43	78	105	77	125	**0**	**40**	**6**	**27**	**4**
27	470	401	369	136	21	444	457	455	484	494	392	401	371	434	371	392	334	330	352	381	79	145	184	144	205	**40**	**0**	**40**	**6**	**44**
28	399	330	303	69	9	373	386	385	413	423	322	330	301	364	301	327	231	225	285	318	29	75	103	74	120	**6**	**40**	**0**	**27**	**4**
29	462	392	365	128	18	436	449	448	476	486	385	393	364	427	364	388	331	325	347	378	73	138	147	137	195	**27**	**6**	**27**	**0**	**40**
30	393	323	295	68	12	367	380	378	406	417	315	324	294	357	294	319	224	218	237	309	28	73	79	72	118	**4**	**44**	**4**	**40**	**0**

Figure 5-7: Latency between collaborators used for the current simulation.2

As mentioned before, each collaborator instance is running as a local process that exposes the services on a specific port. The following code is used in the simulator to launch the collaborator instances.

```
ProcessStartInfo startInfo = new ProcessStartInfo();
startInfo.FileName = ConfigurationManager.AppSettings["CollaboratorExe"];
startInfo.Arguments = "B \"" + brokerPorts[brokerId] + "\" " + appPorts[brokerId];
System.Diagnostics.Process.Start(startInfo);
```

The code shows the arguments that are passed to the executable, which includes the running mode (a B stands for broker-leader and a C for client) and the ports to be used. The first port permits to access the application services and the second is used for the communication with other nodes.

Each process runs as a Windows console application that wrap the class *AppCollab* which implements the *IApplicationService* interface. This console application starts the *AppCollab* which provides all the services to share data and to subscribe to specific topics.

The simulator creates a service client for each of the collaboration instances and calls directly the services in order to publish a new event (share a new resource event or an update event). In order to simulate the latency of the links, the application obtains the latency value of the correspondent link from the matrix and sets a delay before the message is delivered.

5.3 Simulation results

The simulation was run on a single machine with a processor Intel core i5. The simulator initializes the evaluation scenarios introduced in the previous section. After the scenarios were set up, the simulator generated update events to different shared resources. In order to collect the data, a class *DataCollector* implements the observer pattern collecting the information from the events generated in the application. Every time a message is received, the latency between the sender and receiver is obtained from the message itself which stores the time it was created. The DataCollector additionally keeps a counter of the updates delivered. For this current simulation, the number of messages received by the lead nodes and the network latency involved on delivering messages to the subscribers was measured.

The results are presented in table 5-4. The total latency corresponds to the sum of the difference between the time a message was delivered and its creation time. For the current calculation, the overhead time is not considered since the simulation is executed on the same machine and it would be similar to all the messages. Each node has a data collector which calculates the total latency and the latency average for each message.

Figure 5-8 shows trend lines for the two scenarios proposed including a third additional scenario that corresponds to a case in which collaborators are subscribed to topics randomly, instead of subscribing only to topics within their region. These trend lines were obtained applying a linear regression to the data collected in the simulations.

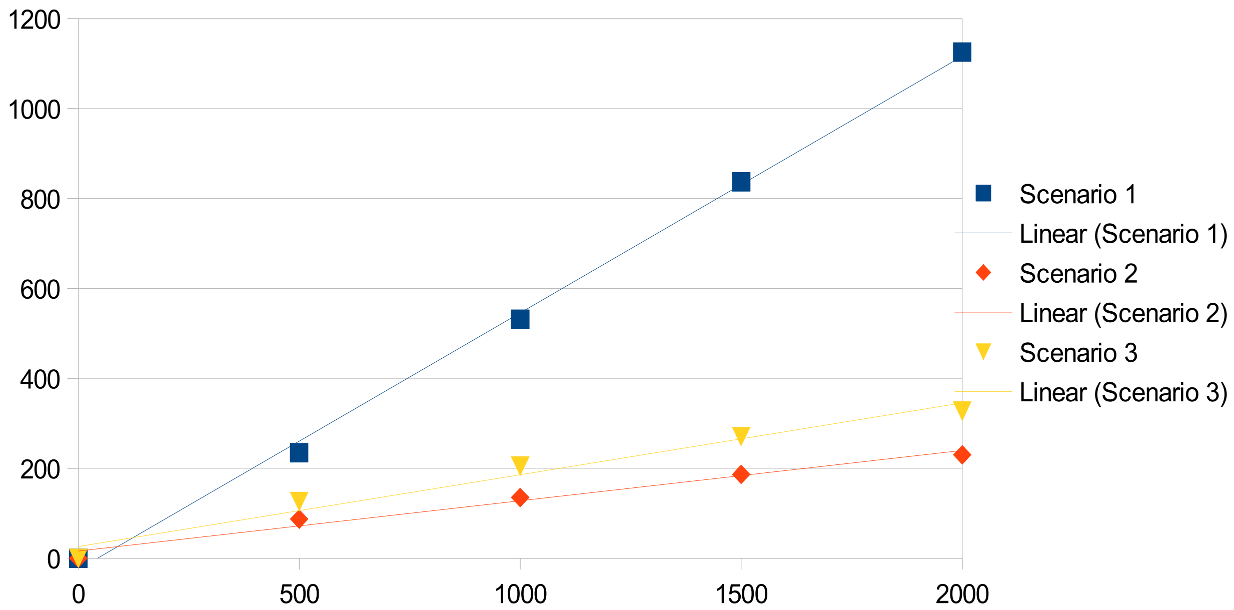


Figure 5-8: Total latency. Scenario 1 is a centralized approach. Scenario 2 is the proposed approach. Scenario 3 is the inter group exchange approach.

The trend lines' equations obtained are presented in table 5-3. These equations can be used to estimate the latency for a higher among of messages. For example, the estimated latency when 10000 events are executed is 2853219.6366 ms and 559672.8271 ms for scenario 1 and 2 respectively, obtaining a latency reduction of 80.39 % in the case of scenario 2 with respect to scenario 1.

5-3. Table: Trend lines equations obtained based on the simulation results

Scenario	Equation
Scenario 1	$f(x) = 285.353x - 310.3634$ $R^2 = 0.9980203265$
Scenario 2	$f(x) = 55.9713x - 40.1729$ $R^2 = 0.9801391743$
Scenario 3	$f(x) = 79.8692x - 53.8214$ $R^2 = 0.9722914391$

Table 5-4 presents the results obtained in the simulation for the 2 scenarios proposed. Scenario 1 corresponds to the centralized approach, in which only one leader is elected (located in region 3). This leader becomes a server node and all the other nodes are the clients. If an event occurs in the system, the server node informs the subscribed nodes. Scenario 2 corresponds to the proposed solution in which nodes are organized based on intra-latency and a leader node is elected on each group.

According to the results obtained from the simulation, it can be seen that the latency increases significantly with the number of updates executed in the case of a centralized architecture. In this scenario, the regions that are closer to the server have a lower latency compared to the regions farther located. On the other hand, in the case of the solution proposed (scenario 2) the latency does not vary significantly between regions and maintains a constant increasing rate significant lower than the generated by the centralized scenario (scenario 1).

The number of messages shown on table 5-4 corresponds to the total number of messages processed by the leader of each region. These are the messages forwarded to neighbor leaders (according to the broker network) and the messages that are submitted to the subscribed nodes of the correspondent region. In the case of a centralized approach (Scenario 1), message forwarding is not required since the server node is connected to all the nodes, therefore, message redundancy is nonexistent. This can be explained due to the overlay network that routes messages between the brokers (leader nodes). One factor that influences the number of messages processed by a leader is the number of connections with neighbor leaders. For example, the leader in region 1 is connected to region 2 and 3 and the number of messages processed is around 39% lower compared to the number of messages processed by leader 3 which is connected to 4 neighbor leaders.

5-4. Table: Results of the simulation. Scenario 1 corresponds to the centralized architecture. Scenario 2 corresponds to the proposed solution.

	Region 1	Region 2	Region 3	Region 4	Region 5
Scenario 1					
100 updates					
Messages # (leader)	0	0	257	0	0
Total latency	11740	5717	4588	14945	23647
Sum avg. latency	1096.94	638.55	309.53	1525.6	2919.6
Avg. latency	632.61	356.23	261.43	912.52	1627.29
Update messages	93	82	87	82	74
	126.522	71.246	52.286	182.504	325.458
500 updates					
Messages # (leader)	0	0	657	0	0
Total latency	52276	36372	18332	75814	104065
Sum avg. latency	636.44	419.64	208.06	847.11	1588.2
Avg. latency / msg	127.288	83.928	41.612	169.422	317.64
1000 updates					
Messages # (leader)	0	0	1157	0	0
Total latency	103893	80401	31335	136198	179490
Sum avg. latency	640.11	448.23	177.48	829.55	1584.45
Avg. latency / msg	128.022	89.646	35.496	165.91	316.89
1500 updates					
Messages # (leader)	0	0	1657	0	0
Total latency	171621	102700	40855	195234	326633
Sum avg. latency	641	386.43	160.76	810.72	1571.08
Avg. latency / msg	128.2	77.286	32.152	162.144	314.216
2000 updates					
Messages # (leader)	0	0	2157	0	0
Total latency	225634	155056	54923	266594	423328
Sum avg. latency	639.81	421.97	158.05	809.24	1562.97
Avg. latency / msg	127.962	84.394	31.61	161.848	312.594
Scenario 2					
100 updates					
Messages # (leader)	428	543	642	531	431
Total latency	3362	3545	3855	3805	3807
Sum avg. latency	189.24	222.15	208.22	213.1	213.04
Avg. latency / msg	37.848	44.43	41.644	42.62	42.608
500 updates					
Messages # (leader)	1144	1440	1816	1420	1155
Total latency	16122	17838	16326	19014	17672
Sum avg. latency	181.71	211.68	205.6	217.92	202.93
Avg. latency / msg	36.342	42.336	41.12	43.584	40.586
1000 updates					
Messages # (leader)	2004	2589	3239	2591	2052
Total latency	26581	27962	25638	30832	24154
Sum avg. latency	146.32	154.84	153.68	175.26	147.85
Avg. latency / msg	29.264	30.968	30.736	35.052	29.57
1500 updates					
Messages # (leader)	2905	3696	4718	3695	2961
Total latency	35064	35930	36475	42960	36018
Sum avg. latency	130.39	138.92	138.17	166.05	147.21
Avg. latency / msg	26.078	27.784	27.634	33.21	29.442
2000 updates					
Messages # (leader)	3805	4712	6330	4758	3870
Total latency	41696	46016	44707	56400	41300
Sum avg. latency	119.16	130.76	131.29	158.47	132.37
Avg. latency / msg	23.832	26.152	26.258	31.694	26.474

5.4 Aras Innovator Integration

Aras innovator (section Aras Innovator) provides a broad set of functionality and integration with different CAD tools, implements standards such as STEP (section Standard for Product Data Exchange (STEP)), security, and is based on a client-server model. The proposed solution can be integrated with Aras through SOAP/XML web services. Figure 6- illustrates the integration between Aras Innovator and the proposed solution.

The current propotype has not been integrated to Aras Innovator due to differences on the .Net framework version (Aras is build with framework 1.1 and the prototype with version 4). However, the new version of Aras innovator is being developed with a compatible framework version.

6 Conclusions

PLM is an effective approach for integrating all the activities related to the product lifecycle and during recent years it has become more relevant due to changes in the model of production (see Chapter Introduction). The increasingly dynamic global context has forced companies to look for alternatives to their traditional centralized production paradigm. Global collaboration seeks to increase the efficiency, reduce the time to market, and reduce costs of product development by taking advantage of the expertise of companies all over the world (section Collaborative Product Development).

Different concepts have emerged for managing the different phases of product development and PLM tries to embrace them in a holistic approach. However, its adoption is not an easy task since PLM systems must integrate heterogeneous systems each with their own data formats and the PLM systems must facilitate fluent data exchange throughout the life cycle of the product (section Product Lifecycle Management Systems). In addition, PLM systems are specialized distributed systems; hence they face the typical challenges of a distributed systems, such as concurrency, lack of global clock, and independent failures (section Distributed Systems). As a result, the current project combined different techniques and technologies as a response to the specific challenges of PLM and the technical challenges of distributed systems.

One of the aspects considered in the proposal is the data access control. Collaboration implies sharing information that is constantly updated (such as design files or requirements documents), demanding a data access control mechanism to maintain data consistency. The proposed solution applies a concurrency model based on a lock mechanism (as described in section A concurrency model for PDM systems).

Another aspect of collaborative product development considered is the geographical distribution of the participants which can affect the inter-site network latency. This network latency can affect the throughput of the communication particularly when large files, such as CAD designs, are exchanged (section Latency based dynamic grouping aware cloud scheduling), hence the proposed solution utilizes a network structured by an algorithm that minimizes the inter-site latency between collaborators.

Although the set of participants in a collaboration project is dynamic, collaborators need to be manually added to the network by the project's administrator. This characteristic makes it easier to maintain the network's group structure using a centralized management entity within each group to control the distribution and sharing of data. A benefit of creating these groups is that it facilitates identifying close nodes with a copy of the resource in order to reduce the time to obtaining a copy of the shared resources. This is possible because each group leader maintains a registry of the local copies of resources distributed within the group.

Another aspect of collaboration considered is the fact that relationships are ephemeral, as collaborators work together only for specific processes or phases of a product's development, which demands a flexible integration mechanism. The proposed solution facilitates this integration through SOA and web services as a means to provide a flexible solution that enables businesses integration (section SOA and PLM).

The solution proposed is based on the publish-subscribe pattern instead of a client-server model. This alternative model decouples the producers and consumers providing scalability. In addition, the results obtained during the simulation showed that the latency can be reduced if a publish-subscribe architecture is implemented considering a scenario in which the information exchange between collaborators is mainly based on groups which are arranged based on a geographical distribution. This scenario can occur, for example, when specialized collaborators located in a country or region develop a specific subsystem of a product [44]. The experimentation was based on a simulation instead of an emulation, and trend functions were modelled using the data generated. These functions can be used by Aras Innovator in order to predict the system behavior both on a centralized and a centralized approach.

Finally, the solution presented achieves the goal of this master thesis by combining different techniques. A concurrency model based on a node election which permits to maintain the consistency of the data exchange. Scalability through a publish-subscribe protocol based a broker network overlay and groups arranged using a clustering algorithm with respect to the intra-group latency.

6.1 Future work

The proposed solution does not consider real-time collaboration that could permit editing of designs simultaneously by multiple collaborators. However, the publish-subscribe model and the latency based approach could be considered for supporting real-time collaboration instead of a client-server architecture as proposed by Smparounis et al. [49]. A future study should evaluate the publish-subscribe model and the latency based grouping mechanism, as an alternative to solutions such as that described by Li, Gao, and Wang [36] who proposes a client-server model.

One aspect that the proposed solution has not considered is node failures, as the proposed solution relies on the collaborators' own infrastructure. However, it a real deployment it is important to provide failure recovery mechanisms, in particular when the group leader node fails. In such a case the nodes should implemented a failure detector and a mechanism for electing a new leader.

The prototype can be further developed by adding some additional features and enhancements and integrated with the new version of Aras innovator. For example, a feature for monitoring the network's status would permit an administrator to identify node failures or other major problems in the network. Statistical information could be collected in order to optimize the network. Another feature would be a simulator that could predict the possible behaviors of the network before it is deployed. A possible architecture of the integration is shown in Figure 6.1.

In the current project we have proposed the use of an algorithm for creating collaborator groups based on the inter-site latency. This solution takes advantage of the close interaction between groups, significantly reducing the latency compared to a centralized solution. In addition, a different communication model could be implemented and compared with the proposed solution which is based on publish-subscribe and message filtering. It would be also interesting to test the current implementation in a real environment rather than a simulation on a single computer.

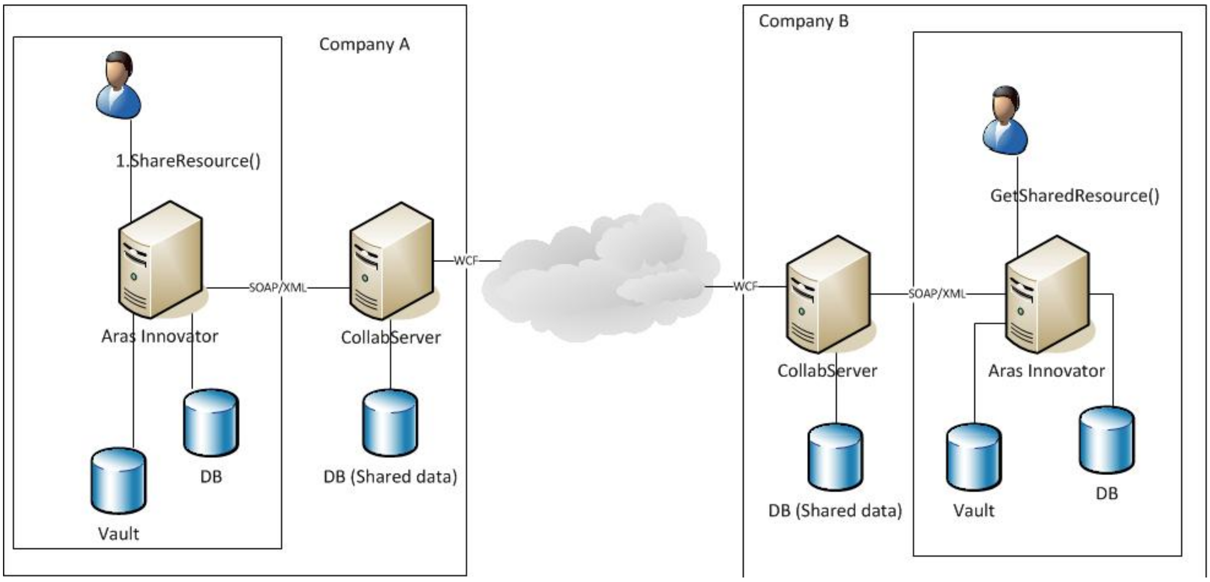


Figure 6-1 Integration of the approach with Aras Innovator.

Finally, the proposed approach can be also extended to other types of collaborative work with shared resources. For example, in areas such as software development, where software outsourcing has become common leading to distributed teams developing different subsystems.

6.2 Reflections

The result of this master's thesis project is a prototype of a module that enables data exchange in a distributed collaborative PLM environment. The proposed solution provides a flexible solution to meet the requirements where collaboration between different companies is becoming more important.

One of the difficulties of the project was to find a C# .Net environment for simulating the scenario and testing the proposed solution. A custom simulator had to be implemented making use of the services provided by the application.

The current prototype provides a solution based on web services facilitating the integration of heterogeneous PLM systems. Open solutions permit businesses to implement PLM without being attached to a specific software provider and building their infrastructure in a way that minimizes costs and maximizes their benefits. This is particularly important for smaller companies which do not have budgets to cover the cost of expensive solutions.

Product development is going on a fast transition to a global collaborative approach where PLM is an important component. Global product development aims to reduce costs and maximize productivity through efficiency and flexibility while enabling innovation. In this context, information exchange between nodes is the core for enabling collaboration. This thesis has proposed a solution to facilitate this collaboration while minimizing latency considering the intra-site latencies.

Finally, society benefits from a more efficient product development. Not only are costs reduced but also the resources used in manufacturing are reduced. The holistic approach of PLM can also benefit the environment since it considers from the beginning all the phases of the product, including the post usage phase.

Bibliography

[1] J. Stark, "Complex and Changing Environment," in Product Lifecycle Management: 21st Century Paradigm for Product Realisation, London: Springer London, 2011, pp. 17–41. [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-0-85729-546-0_2 [Accessed: 27-Dec-2012].

[2] David strom and Jelle Frank van der Zwet, Cloud whitepapers, Interxion, Truth and Lies about Latency in the Cloud, May 2012. Available: http://www.interxion.com/Documents/Whitepapers%20and%20PDFs/Truth-and-Lies-about-Latency-in-the-Cloud-Whitepaper.pdf

[3] W. Y. Chang, H. Abu-Amara, and J. F. Sanford, "Challenges of Enterprise Cloud Services1," in Transforming Enterprise Cloud Services, Springer Netherlands, 2010, pp. 133–187.

[4] Ashish Vulimiri, Oliver Michel, P. Brighten Godfrey, and Scott Shenker. 2012. More is less: reducing latency via redundancy. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (HotNets-XI). ACM, New York, NY, USA, 13-18. DOI=10.1145/2390231.2390234 http://doi.acm.org/10.1145/2390231.2390234

[5] Ashish Vulimiri, Oliver Michel, P. Brighten Godfrey, and Scott Shenker. 2012. More is less: reducing latency via redundancy. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (HotNets-XI). ACM, New York, NY, USA, 13-18. DOI=10.1145/2390231.2390234 http://doi.acm.org/10.1145/2390231.2390234

[6] Dale Littler, Fiona Leverick, Margaret Bruce, Factors affecting the process of collaborative product development: A study of UK manufacturers of information and communications technology products, Journal of Product Innovation Management, Volume 12, Issue 1, January 1995, Pages 16-32, ISSN 0737-6782, http://dx.doi.org/10.1016/0737-6782(94)00025-B. (http://www.sciencedirect.com/science/article/pii/073767829400025B)

[7] J. Stark, "Traditional Pre-PLM Environment," in Product Lifecycle Management: 21st Century Paradigm for Product Realisation, Second., UK: Springer London, 2011.

[8] J. Stark, "Product Lifecycle Management," in Product Lifecycle Management: 21st Century Paradigm for Product Realisation, Second., UK: Springer London, 2011.

[9] S. Jiun-Yan, "Research Issues on Collaborative Product Design and Development," in The Way to Flat Organisation, Supply Chain, Publisher: InTech, Dec. 2008.

[10] V. Srinivasan, "An integration framework for product lifecycle management," Computer-Aided Design, vol. 43, no. 5, pp. 464–478, May 2011.

[11] George F. Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, Distributed systems : concepts and design, 5th ed., International ed. Harlow, Essex: Pearson Education ; Addison-Wesley, ISBN 10: 0132143011, ISBN 13: 9780132143011, 2012.

[12] E. Bertino, L. D. Martino, F. Paci, and A. C. Squicciarini, "Standards for Web Services Security," in Security for Web Services and Service-Oriented Architectures, Springer Berlin Heidelberg, 2010, pp. 45–77.

[13] P. Krzyzanowski, "Group Communication", lecture notes on Distributed Systems, CS 417 Documents, Rutgers University, 2001. Available: http://www.pk.org/rutgers/notes/index.html

[14] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The Many Faces of Publish/Subscribe.” ACM Computing Surveys (CSUR),Volume 35 Issue 2, June 2003, Pages 114-131.
[15] S. Bourchenak and N. De Palma, “Message Queuing Systems.” Department of Computer Science, University of Grenoble I, Encyclopedia of Database Systems, chapter Message Queuing Systems. Springer Verlag, 2008.
[16] B. Nitzberg and V. Lo, “Distributed shared memory: a survey of issues and algorithms,” Computer, vol. 24, no. 8, pp. 52–60, 1991.
[17] The World Wide Web Consortium, “Web Services Architecture current status,” 2013. [Online]. Available: http://www.w3.org/standards/techs/wsarch#w3c_all .
[18] World Wide Web Consortium (W3C), “Web Services Description Language (WSDL) 1.1,” 15-Mar-2001. [Online]. Available: http://www.w3.org/TR/wsdl . [Accessed: 20-Jul-2013].
[19] World Wide Web Consortium (W3C), “Extensible Markup Language (XML),” 26-Nov-2008. [Online]. Available: http://www.w3.org/TR/xml/ . [Accessed: 30-Jun-2013].
[20] World Wide Web Consortium (W3C), “SOAP Version 1.2,” 27-Apr-2007. [Online]. Available: http://www.w3.org/TR/soap/ . [Accessed: 30-Jun-2013].
[21] Y. Oh, S. Han, and H. Suh, “Mapping product structures between CAD and PDM systems using UML,” Computer-Aided Design, vol. 33, no. 7, pp. 521–529, Jun. 2001.
[22] T. Paviot, S. Lamouri, and V. Cheutet, “A generic multiCAD/multiPDM interoperability framework,” International Journal of Services Operations and Informatics, vol. 6, no. 1, pp. 124–137, Jan. 2011.
[23] D. Sprott and L. Wilkes, “Understanding Service-Oriented Architecture.”, CBDI Forum, January 2004. Available: http://msdn.microsoft.com/en-us/library/aa480021.aspx
[24] E. Bertino, L. D. Martino, F. Paci, and A. C. Squicciarini, “Web Service Technologies, Principles, Architectures, and Standards,” in Security for Web Services and Service-Oriented Architectures, Springer Berlin Heidelberg, 2010, pp. 9–23.
[25] V. Jinesh and M. Sajee, “Overview of Amazon Web Services,” Mar-2013. [Online]. Available: http://media.amazonwebservices.com/AWS_Overview.pdf. [Accessed: 08-Jul-2013].
[26] Google Inc., “Google Maps API Web Services,” 11-Oct-2012. [Online]. Available: https://developers.google.com/maps/documentation/webservices/ . [Accessed: 30-Jun-2013].
[27] G. A. Lewis and D. B. Smith, “Service-Oriented Architecture and its implications for software maintenance and evolution,” in Frontiers of Software Maintenance, 2008. FoSM 2008., 2008, pp. 1–10.
[28] A. Saaksvuori and A. Immonen, Product Lifecycle Management. Springer, 2008.
[29] D. Bergsjo, A. Catic, and J. Malmqvist, “Implementing a service-oriented PLM architecture focusing on support for engineering change management,” International Journal of Product Lifecycle Management, vol. 3, no. 4, pp. 335–355, Jan. 2008.
[30] S. Hachani, H. Verjus, and L. Gzara, “Support of product design processes flexibility in PLM systems using a service–based approach,” International Journal of Services Operations and Informatics, vol. 7, no. 4, pp. 313–329, Jan. 2012.

[31] S. Hachani, L. Gzara, and H. Verjus, “A service-oriented approach for flexible process support within enterprises: application on PLM systems,” Enterprise Information Systems, vol. 7, no. 1, pp. 79–99, 2013.
[32] R. Credle, M. Bader, M. Holt, Y. Hyakuna, E. McCarty, L. Mommeja, and M. Novaes, SOA Approach to Enterprise Integration for Product Lifecycle Management. IBM Corp., 2008. Available: http://www.redbooks.ibm.com/abstracts/SG247593.html?Open
[33] B. Kim and S. Han, “Sharing of CAD assembly model data using parallel Web Services,” in 12th International Conference on Computer Supported Cooperative Work in Design, 2008. CSCWD 2008, 2008, pp. 434–440.
[34] S. Rachuri, E. Subrahmanian, A. Bouras, S. J. Fenves, S. Foufou, and R. D. Sriram, “Information sharing and exchange in the context of product lifecycle management: Role of standards,” Computer-Aided Design, vol. 40, no. 7, pp. 789–800, Jul. 2008.
[35] Economic and Social Commission for Asia and the Pacific, “Integrating Environmental Considerations into Economic Policy Making Processes - International commitments,” ESCAP Virtual Conference, 11-Dec-2003. [Online]. Available: http://www.unescap.org/drpad/vc/orientation/M3_lnk_trade.htm. [Accessed: 30-Aug-2013].
[36] M. Li, S. Gao, and C. C. Wang, “Real-Time Collaborative Design With Heterogeneous CAD Systems Based on Neutral Modeling Commands,” J. Comput. Inf. Sci. Eng., vol. 7, no. 2, pp. 113–125, Sep. 2006.
[37] E. Chan and K. M. Yu, “A concurrency control model for PDM systems,” Computers in Industry, vol. 58, no. 8–9, pp. 823–831, Dec. 2007.
[38] S. Malik, F. Huet, and D. Caromel, “Latency Based Dynamic Grouping Aware Cloud Scheduling,” in 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2012, pp. 1190–1195.
[39] Y.-P. Luh, C.-C. Pan, and C.-H. Chu, “A hierarchical deployment of distributed product lifecycle management system in collaborative product development,” International Journal of Computer Integrated Manufacturing, vol. 24, no. 5, pp. 471–483, 2011.
[40] C.-H. Chu, Y.-H. Chan, and P. H. Wu, “3D streaming based on multi-LOD models for networked collaborative design,” Computers in Industry, vol. 59, no. 9, pp. 863–872, Dec. 2008.
[41] Siemens Product Lifecycle Management Software Inc., “Teamcenter Multi-Site Collaboration White Paper,” 2010. [Online]. Available: http://www.plm.automation.siemens.com/es_es/Images/Siemens-PLM-Teamcenter-Multi-Site-Collaboration-wp_tcm52-55446.pdf . [Accessed: 30-Jun-2013].
[42] “Free Website Testing Tools to Help You Improve Website Performance,” Dotcom monitor, 24-Sep-2013. [Online]. Available: http://www.dotcom-monitor.com/WebTools/network_latency.aspx .
[43] Microsoft, “.Net technology guidance.” [Online]. Available: http://www.microsoft.com/net. [Accessed: 14-Sep-2013].
[44] C. Rodríguez Monroy and J. R. Vilana Arto, “Analysis of global manufacturing virtual networks in the aeronautical industry,” International Journal of Production Economics, vol. 126, no. 2, pp. 314–323, Aug. 2010.
[45] Geo Mexico, “The reasons why Mexico is fast becoming a key player in aerospace manufacturing,” September 2011. [Online]. Available: http://geo-mexico.com/?p=4977 .

[46] B. Kim, “Retrieval of CAD model data based on Web Services for collaborative product development in a distributed environment,” International Journal of Advanced Manufacturing Technology, vol. 50, no. 9–12, pp. 1085–1099, 2010.

[47] EACP, “European Aerospace Cluster Partnership members.” [Online]. Available: http://www.eacp-aero.eu/.

[48] “WCF feature details, security overview” Microsoft MSDN, 02-Oct-2012. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms735093%28v=vs.110%29.aspx

[49] Smparounis Konstantinos, Alexopoulos Kosmas, Mavrikios Dimitris, Pappas Menelaos, Xanthakis Vagelis, “A web-based platform for collaborative product design, review and evaluation” Digital factory for human-oriented production system : the integration of international research projects, 2011, Vol.<35-56, pp. 35-56

[50] B. Constantine, G. Forget, R. Geib, and R. Schrage, “Framework for TCP Throughput Testing”, Internet Request for Comments, vol. RFC 6349 (Informational), Aug. 2011 [Online]. Available: http://tools.ietf.org/html/rfc6349

[51] D. Bernhardt and A. Klein, ‘Implementation of a Fault-Tolerant File Management System’, in Fehlertolerierende Rechensysteme, vol. 84, K.-E. Großpietsch and M. Cin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 252–264 [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-69698-5_21. [Accessed: 12-Oct-2013]

[52] J. Wechta, A. Eberlein, F. Halsall, The impact of topology and choice of TCP window size on the performance of switched LANs, Computer Communications, Volume 22, Issue 10, 25 June 1999, Pages 955-965, ISSN 0140-3664, Available: http://www.sciencedirect.com/science/article/pii/S0140366499000626

[53] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. 1999. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (MobiCom '99). ACM, New York, NY, USA, 174-185. DOI=10.1145/313451.313529. Available: http://doi.acm.org/10.1145/313451.313529

[54] Scania Group, “Krona and economic cycle a challenge for Scania,” 11-Sep-2013. [Online]. Available: http://scania.com/investor-relations/scania-value/article-archive/Krona-and-economic-cycle-a-challenge-for-Scania.aspx .

[55] G. Büyüközkan and J. Arsenyan, “Collaborative product development: a literature overview,” Production Planning & Control, vol. 23, no. 1, pp. 47–66, 2012.

[56] J. H. Lee, H.-J. Shim, and K. K. Kim, “Critical Success Factors in SOA Implementation: An Exploratory Study,” Information Systems Management, vol. 27, no. 2, pp. 123–145, Spring 2010.

[57] K. Rouibah, S. Rouibah, and W. M. P. Van Der Aalst, “Combining workflow and PDM based on the workflow management coalition and STEP standards: the case of axalant,” International Journal of Computer Integrated Manufacturing, vol. 20, no. 8, pp. 811–827, 2007.

[58] Banker R, Bardhan I, Asdemir O. Understanding the Impact of Collaboration Software on Product Design and Development. Information Systems Research [serial on the Internet]. (2006, Dec), [cited November 20, 2013]; 17(4): 352-373. Available from: Business Source Elite.

TRITA-ICT-EX-2013:247