



# **Project Report: Intelligent Semantic Oriented Agent based Search (I-SOAS)**

**Author:  
Zeeshan AHMED**

**Contact Information:**

Author:

Zeeshan Ahmed

Phone: 0043 158801 30726

Cell: 0043 699 81302854

Office Address: Getreidemarkt 9/307 1060 Vienna Austria

E-Mail: [zeeshan.ahmed@tuwien.ac.at](mailto:zeeshan.ahmed@tuwien.ac.at)

University advisor:

Univ.-Prof. Dr.-Ing. Detlef Gerhard

Phone: 0043 158801 30722

Email: [detlef.gerhard@tuwien.ac.at](mailto:detlef.gerhard@tuwien.ac.at)

Mechanical Engineering Informatics and  
Virtual Product Development Division (MIVP)  
Vienna University of Technology  
Getreidemarkt 9/307  
1060 Wien  
Austria

Phone: +43 158801 30721

Fax: + 43 158801 3098

Internet: [mivp.tuwien.ac.at](http://mivp.tuwien.ac.at)

## ABSTRACT

Product Data Management (PDM) produced desktop and web based systems to maintain the organizational technical and managerial data to increase the quality of products by improving the processes of development, business process flows, change management, product structure management, project tracking and resource planning. Though PDM is heavily benefiting industry but PDM community is facing some serious unresolved issues in PDM System development. i.e., unfriendly graphical user interfaces and unintelligent search. PDM Systems offer different many services and functionalities at a time but the graphical user interfaces of most of the PDM Systems are not designed in a way that a user (especially a new user) can easily learn and use them. Moreover, PDM Systems contain and manage heavy amount of data but the search mechanism of most of the PDM System is not intelligent which can process user's structured or unstructured natural language based queries to extract desired information.

Targeting above discussed two serious PDM based issues a thorough research is conducted in fields of Product Data Management (PDM), Human Machine Interfaces (HMI) and Semantic Web (SW). Concerning PDM, conducted research discusses PDM desktop and web based systems, PDM system development guidelines and some existing PDM Systems. Concerning HMI, conducted research provide information about HMI Designing including Ideologies, Principles, Patterns, Design Guidelines, Basic Design Goals, Problems and Rich Internet Applications (RIA) for Web based HMI Development. Concerning SW, conducted research provides information about Web based information modeling and processing, Ontologies and semantic oriented information modeling based approaches.

Using conducted research based information and keeping some existing solutions toward targeted PDM community's issues in mind a semantic oriented, information processing and intelligent graphical user interface based solution is proposed. i.e., Intelligent Semantic Oriented Agent based Search (I-SOAS). Describing the proposition in detail the main concept, implementation designs and developed prototype of I-SOAS is discussed in this dissertation.

**Keywords:** Agents, Product Data Management, Human Machine Interface, Information Modeling, Search, Semantic web, Web

## TABLE OF CONTENTS

### Contents

1	Introduction.....	10
1.1	Preface.....	10
1.2	Goals of Research.....	10
1.3	Scope of research.....	11
1.4	Structure of Research.....	12
1.5	Structure of Dissertation.....	14
2	Product Data Management (PDM).....	15
2.1	Introduction to PDM.....	15
2.2	PDM Systems.....	16
2.3	Challenges to PDM.....	16
2.3.1	Static and Unfriendly Graphical User Interface.....	16
2.3.2	Static and Unintelligent Search.....	16
2.4	Related Work.....	17
2.4.1	K2 Mobility SVT: VIF [81].....	17
2.4.2	VIVACE: EU [82].....	19
2.4.3	Promise PLM: EU [84].....	19
2.4.4	Component-based Product Data Management system [63].....	20
2.4.5	Web-based product data management (WPDM) [64].....	21
3	Designing Human Machine Interface.....	23
3.1	Human Machine Interface (HMI).....	23
3.2	HMI Ideologies.....	23
3.3	HMI Design Principles.....	24
3.4	HMI Design Patterns.....	25
3.5	HMI Design Goals and Guidelines.....	25
3.6	RIA for Web based HMI Development.....	26
3.7	HMI: Related Work.....	28
3.7.1	Windchill [85].....	28
3.7.2	CIM Database [86].....	29
4	Web based Information Processing.....	31
4.1	Web.....	31
4.2	Web to Semantic Web.....	31
4.3	Current Challenges of Semantic Web.....	32
4.4	Goals of Semantic Web.....	32
4.5	Semantic Oriented Approaches.....	33
4.5.1	Ontology.....	33
4.5.2	Language Technology.....	36
4.5.3	Semantic desktop: Personal Information Model (PIM) [22].....	38
4.5.4	Reisewissen [11].....	39
4.5.5	Test Application [26].....	39
4.5.6	Meta Data Search Layer [27].....	40

4.5.7	Cultural Heritage and the Semantic Web [38]	41
4.5.8	Semantic Security Web Services (SSWS) [30]	41
4.5.9	Partial Distance Map (PDM) [62]	41
5	I-SOAS Approach	43
5.1	Intelligent User Interface	44
5.2	Information Processing	44
5.3	Data Management	45
5.4	Data Representation	46
6	Implementation	48
6.1	Implementation Design	48
6.1.1	System Sequence Architecture	48
6.1.2	Technological Involvement	49
6.2	Graphical Interface	50
6.2.1	Design Requirements	52
6.2.2	Draft - Graphical Interface Design	53
6.2.3	Internal Work Flow	55
6.2.4	System Sequence Design	56
6.3	Processing and Modeling	56
6.3.1	Design Requirements	57
6.3.2	Internal Work Flow	58
6.3.2.1	Grammatical View	59
6.3.2.2	Ontological View	60
6.3.3	System Sequence Design	61
6.4	Repository	62
6.4.1	Design Requirements	63
6.4.2	Relational Design	64
6.4.3	I-SOAS Repository: System Sequence Design	68
6.5	Knowledge Base	69
6.5.1	Design Requirements	70
6.5.2	I-SOAS Knowledge Base Categories	71
6.5.3	Knowledge Base Cycle	71
6.5.4	System Sequence Design	72
6.5.5	Internal Work Flow	73
6.6	Process Presentation Layer	74
6.6.1	Design Requirements	75
6.6.2	System Sequence Design	75
6.6.3	Internal Work Flow	75
6.7	I-SOAS Process Database Layer	76
6.7.1	Design Requirements	77
6.7.2	System Sequence Design	77
6.7.3	Internal Work Flow	77
6.8	Process Knowledge Layer	78
6.8.1	Design Requirements	78
6.8.2	System Sequence Design	79
6.8.3	Internal Work Flow	79
7	I-SOAS Prototype	81

7.1	I-SOAS Desktop Application – Information Processor .....	81
7.1.1	Information Processor.....	82
7.1.2	Create Database .....	84
7.1.3	Open Database .....	85
7.1.4	Connect Database .....	86
7.1.5	File Configuration.....	87
7.1.6	Relation Builder.....	89
7.1.7	Query Builder .....	90
7.1.8	Result Visualizer.....	94
7.2	I-SOAS Web Application – Graphical Interface & Search .....	95
7.2.1	Main Page .....	96
7.2.2	Graphical User Interface.....	97
7.2.3	Search.....	100
8	Summary.....	103
9	References.....	105
10	Related Publications .....	112
10.1	Related BS & MS Thesis.....	114
11	Appendix.....	115
11.1	Grammar Used in I-SOAS .....	115

## LIST OF FIGURES

Figure 1: Structure of Research	13
Figure 2: K2 Mobility SVT [81]	17
Figure 3: The architecture of CPDM [63]	21
Figure 4: The architecture of WPDM [64]	22
Figure 5: Windchill by PTC [85]	29
Figure 6: CIM Database [86]	30
Figure 7 Ontology development activities [20]	34
Figure 8: Intelligent - Semantic Oriented Agent based Search (I-SOAS)	43
Figure 9: Intelligent User Interface	44
Figure 10: Information Processing	45
Figure 11: Data Management	46
Figure 12: Data Representation	47
Figure 13 Implementation Architecture	48
Figure 14: System Sequence Architecture	49
Figure 15 Involved Technologies	50
Figure 16: Implementation Architecture - Graphical Interface	51
Figure 17: Manual Physical Sketch Description	54
Figure 18: Graphical Interface Internal Flow	55
Figure 19: Graphical Interface System Sequence Design	56
Figure 20: Implementation Architecture – Processing and Modeling	57
Figure 21: Process & Modelling Internal Work Flow	59
Figure 22: I-SOAS Lexer and Parser Grammar	60
Figure 23: I-SOAS Ontology	61
Figure 24: Process and Modelling System Sequence Design	62
Figure 25: Implementation Architecture – Repository	63
Figure 26: I-SOAS Database Organizational Data (Part 1)	67
Figure 27: I-SOAS Database System Data	68
Figure 28: I-SOAS Database System Data	69
Figure 29: Implementation Architecture – Repository	70
Figure 30: Knowledge Base Categories	71
Figure 31: Knowledge Base Cycle	72
Figure 32: Knowledge Base System Sequence Design	73
Figure 33: Knowledge Base Internal Work Flow	74
Figure 34: Process Presentation Layer System Sequence Design	75
Figure 35: Process Presentation Layer Internal Work Flow	76
Figure 36: Process Database Layer System Sequence Design	77
Figure 37: Process Database Layer Internal Work Flow	78
Figure 38: Process Knowledge Layer System Sequence Design	79
Figure 39: Process Knowledge Layer Internal Work Flow	80
Figure 40: Information Processer	82
Figure 41: Information Processer Description	83

Figure 42: Create new database	85
Figure 43: Open database	85
Figure 44: Connect database	86
Figure 45: File Configuration	88
Figure 46: Relation Builder	89
Figure 47: Query Builder	91
Figure 48: Query Table	92
Figure 49: Query Saver	93
Figure 50: Result Visualizer	94
Figure 51: Main Page	96
Figure 52: Default Graphical Interface – Without User login	97
Figure 53: Graphical Interface –After User login	99
Figure 54: SQL based Search	101
Figure 55: Natural Language based Search	101



## LIST OF TABLES

Table 1: Information Processing	84
Table 2: Create Database	85
Table 3: Open Database	86
Table 4: Connect Database	87
Table 5: File Configuration	89
Table 6: Relation Builder	90
Table 7: Query Builder	92
Table 8: Query Table	93
Table 9: Query Saver	94
Table 10: Result Visualizer	95
Table 11: Main Page	97
Table 12: Graphical Interface Page – User Login	100
Table 13: Search Page	102

# 1 Introduction

In this chapter I introduce the domain, research interest and application area. Moreover I present major identified and targeted challenges of selected research area, scope, goals and structure of my research and this dissertation.

## 1.1 Preface

Most of the technological companies are consists of many departments like management, marketing, accounts, production, quality and engineering etc. Every department has its own rules, regulations, data and information. No doubt every department is important and expected to play a vital role in the progress of the company but most of all is the engineering / technical department, which is more responsible for the main product's production and rest of all other departments of company are dependent on it. To successfully run the technical department; required hardware and software is deployed, process are initiated and implemented, required number of technical staff is hired to produce product under the implemented process and by using provided resources.

Till now everything sounds nice, most of the time problems initiate and start growing as company grows due to the rapid increase in data, lack of presence of required in time information and lack of project and resource management. As the result company can face unnecessary additional increase in costs, delays in product completion, loss in quality and waste of time.

In the past, there were no as such systems available to store, track and manage all the related product data. This doesn't mean that there was no system for data management; there were some systems to store the information about product, personnel involved in organization and financial details but there was no as such comprehensive system to manage technical data. To cope with the problem of organizational technical data manager a new field was introduced. i.e., Product Data Management (PDM).

PDM is a digital way of maintaining organizational data to improve the quality of products and followed processes. PDM products mainly manage information about design and manufacturing of products including technical operations and running projects.

## 1.2 Goals of Research

Though PDM systems are heavily benefiting industry but at the same time PDM community is facing some serious unresolved issues .i.e., enterprise spanning PDM system deployment, static and unfriendly user interfaces and unintelligent search. In my opinion, at the moment PDM community is in need of a new approach which can be very helpful in implementing the concepts of Product Data Management in the form of an intelligent knowledge based software application, which must be capable of intelligently handling user's structured and unstructured digital and natural language based requests, process and model the information for fast,

optimized and efficient search mechanism, provide options to store, manage and extract heavy information in more optimized and better way, provide a compact installation process to deploy PDM based application in real time environment.

Goals of my research work is to support web based platform independent Product Data Management based Application development which should be capable of

- Intelligent handling of user requests.
- Process and model unstructured data.
- Store, manage and extract desired information from repository.

### 1.3 Scope of research

In this research dissertation I am focusing, targeting, addressing, discussing and giving solution to tow major PDM problems. i.e.,

#### 1. Static and Unfriendly Graphical User Interface

In my opinion if a product is very productive and with lots of bifacial functionalities but if it is not easily usable then in most of the cases it become flop in industry. Designing and implementing an intelligent and user friendly human machine interface for any kind of software or hardware oriented application is always be a challenging task for the designers and developers because it is very difficult to understand the psychology of the user, nature of the work and best suit of the environment. Normally PDM Systems offer different many services and functionalities at a time but the graphical user interfaces of most of the PDM Systems are not designed in a way that a user (especially a new user) can easily learn and use them e.g. Windchill by PTC [85] and CIM Database [86] etc. In this research work I am targeting the problem of static and unfriendly graphical user interface and trying to support PDM community by proposing a new intelligent and user-friendly graphical user interface.

#### 2. Static and Unintelligent Search

Designing an intelligent application capable of reading and analyzing user's structured and unstructured text based requests and then extracting desired concrete and optimized results from knowledge base is still a challenging task for the designers because still it is very difficult to completely extract the Meta data out of data. There is a search record mechanism in almost all of the PDM Systems, but none of the PDM System is providing an intelligent and dynamic search which can process user's natural language based queries e.g. Windchill by PTC [85] and CIM Database [86] etc. In this research work I am also targeting this PDM's unresolved issue as well and trying to support PDM community by proposing a new intelligent semantic based information processor.

## 1.4 Structure of Research

The planned and decided structure of research as shown in Figure 1 starts with the introduction to the main domain and the field of interest. i.e., Product Data Management (PDM). Then after making in depth analysis of the field, current existing major challenges are identified, goals and the scope of research is decided. According to the scope and goal of research I decided to target two major existing problems to the PDM community. i.e., static & unfriendly graphical interface and static & unintelligent search. Targeting selected problems three main fields are selected to be explored and researched. i.e., PDM System Development, Human Machine Interface and Semantic Web. Then keeping goals and scope of research in mind these three selected fields are investigated and discussed in detail. After the deep analysis of state of the art I proposed a new solution. i.e., *Intelligent Semantic Oriented Agent based Search (I-SOAS)*. and discussed the proposed concept, implementation designs and prototype in detail in this dissertation.

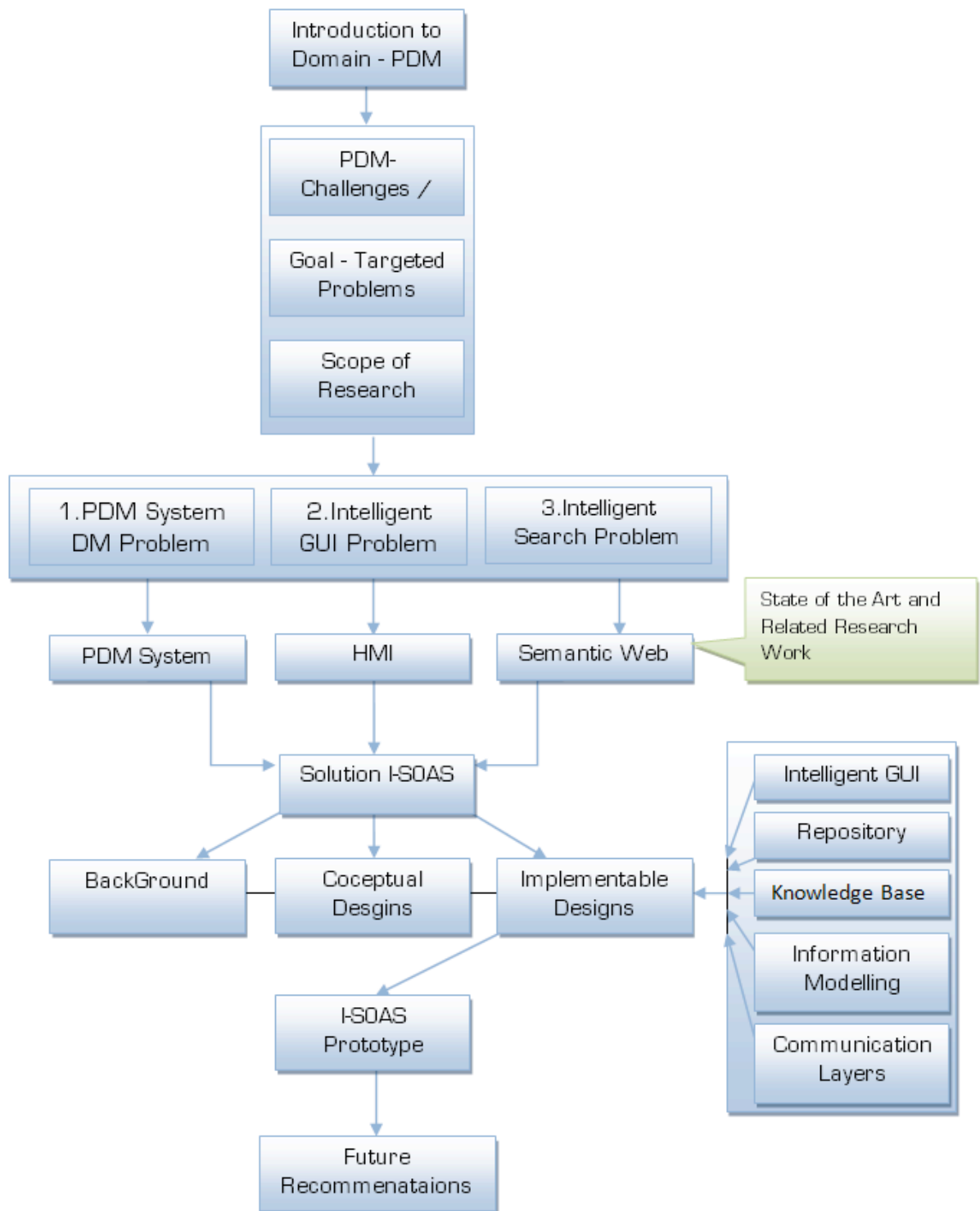


Figure 1: Structure of Research

## 1.5 Structure of Dissertation

*Chapter 1: Introduction* contains the information about selected field of research, major identified and targeted field's challenges, scope, goals and structure of our research.

*Chapter 2: Product Data Management (PDM) System Development:* This chapter provides the detailed information about Product Data Management (PDM), PDM Systems and their currently faced challenges. Moreover this chapter also discusses some PDM based proposed and implemented approaches as the part of related work to my research work.

*Chapter 3: Designing Human Machine Interface:* This chapter provides the detailed information about Designing Human Machine Interface (HMI) including ideologies, design principles, design patterns, design guidelines.

*Chapter 4: Web based information modeling and processing:* This chapter provides the detailed information about Web based information modeling and processing including information about Web, Semantic Web, Ontology and some Semantic Oriented information modeling based Approaches.

*Chapter 5: Semantic based Solution towards PDM Challenges:* This chapter presents a solution to provide comprehensive support in implementing a semantic and agent based solution towards the problems of static and intelligent graphical user interface and unintelligent search. Moreover this chapter also presents the implantation architecture of proposition with the brief explanation of each and every component of main implementation architecture and discusses most suitable and available tools and technologies for the real time development.

*Chapter 6: I-SOAS: Design Implementation:* This chapter provides detailed design based information about major components of I-SOAS system design including the information about the theme, design requirements, system sequence and internal work flow designs of each component.

*Chapter 7: I-SOAS Prototype:* This chapter provides detailed information about currently implemented prototype version of I-SOAS.

*Chapter 8: Summary:* This chapter summarizes and presents the discussions and conclusions of overall researched work; moreover this chapter also presents future recommendations for the followers.

*Chapter 9: References:* This chapter provides used and consulted reference based information.

*Chapter 10: Appendix:* This chapter discusses a performed case study regarding machine interface design.

## 2 Product Data Management (PDM)

This chapter provides the detailed information about Product Data Management (PDM), PDM Systems and their currently faced challenges. Moreover this chapter also discusses some PDM based proposed and implemented approaches as the part of related work to my research work.

### 2.1 Introduction to PDM

In early 1970s there was no as such system to automate the process of data management, then in 1980s Computer Integrated Manufacturing (CIM) was introduced but not seemed to be successful in product data management [87]. Product Data Management (PDM) is a digital way of maintaining organizational data to improve the quality of products and followed processes. PDM products mainly manage information about design and manufacturing of products including technical operations and running projects. PDM is also renowned as Engineering Data Management (EDM) and Engineering Document Management Systems (EDMS), because it provides better management and control over engineering data, activities, and changes related to design and manufacture of product. Like EDMS Product Lifecycle Management (PLM) is another acronym of PDM to manage the entire development life cycle of the product by integrating people, data, processes and business systems [88].

PDM provides a backbone for the controlled flow of engineering information throughout the product life cycle by using engineering data, such as CAD, ERP and field service. Moreover PDM also supports product teams and techniques by providing Concurrent Engineering in improving engineering workflow. PDM systems address issues such as control, quality, reuse, security and availability of engineering data. PDM performs five main functions to integrate and manage all applications, information, and processes during the associated product life cycle i.e., Data vault and document management, Workflow and process management, Product structure management, Parts management and Program management (defined by CIMDATA) [61].

The major objectives of PDM are to reduce the cost of engineering, reduce effort in product development life cycle, reduce time in engineering change handling and new product development, improve the quality and services of product, deliver and support products at the time, improve team coordination, increase customization of products, maintain product configuration based information, manage large volumes of data generated by computer based systems, reduce engineering environment based problems, provide better access to information, provide better reuse of design information, provide common data ware housing, secure engineering data's originality, prevent error creation and propagation and make a strong effect on market shares [59]. Moreover PDM is also supposed to handle business process workflows, change management, revision control, product configurations, product structure management, project tracking and resource planning [60].

## 2.2 PDM Systems

With emergence of CAD technologies PDM Systems are introduced and used to manage engineering data, activities and process through better control of engineering data, activities, changes and product configurations. PDM Systems consist of basic components which are .i.e., data repository, information management module controls to access, store, integrate, secure, recover and manage information, basic networked computer environment based infrastructure, interface module to support user queries, menu and form driven inputs and report, information and workflow structure definition modules to manage resources, events and responsibilities, information structure management module to produce exact structure based on maintained information in the system, workflow structure definition flow and content of engineering activities, workflow control module to control and coordinates the engineering process and system administration module to set up administration and maintain the configuration of the system. To develop a PDM System consisting of at least above mentioned basic components some guidelines/steps are need to be followed e.g. before starting the PDM System it is necessary to understand the currently ongoing engineering processes, working environment, major and minor business objectives, user's system requirements and functionalities of the currently deployed systems in targeted company, collect detailed organizational information, develop information model, define deliverables, identify the most important factors and business metrics for management to reduce lead times and improve product quality, try to identify illogical, wasteful, unnecessary and duplicate activities.

## 2.3 Challenges to PDM

PDM is contributing a lot in industry but also facing some major technical challenges which are .i.e., PDM System implementation, deployment and reinstallation, static & unfriendly graphical user interface, static & unintelligent search, security and non standardized development framework etc. But during this research work I am only focusing on two of these challenges .i.e., static & unfriendly graphical user interface and static & unintelligent Search

### 2.3.1 Static and Unfriendly Graphical User Interface

The graphical user interfaces of available PDM Systems are unfriendly and due to this it's hard for a new user to learn it in short time. Moreover existing PDM System's graphical interfaces are static, nonflexible and unintelligent so then it can automatically learn and redesign itself according to the ease and need of the user.

### 2.3.2 Static and Unintelligent Search

In every PDM System, there is a search mechanism required and implemented to retrieve the user's needed information. Unfortunately there is no as such intelligent search mechanism available that can process user's dynamic request based queries and can extract the most optimized results in minimum possible time in return.



## 2.4 Related Work

In the domain of PDM many products are available and many several methodologies have been introduced by many researchers that are providing lots of values to PDM domain. During detailed review, I put emphasis on the validity of the proposed solutions in literature and presented methodologies. Moreover I am also presenting the relevancy of chosen, reviewed and described related work to my research and development work.

### 2.4.1 K2 Mobility SVT: VIF [81]

K2 Mobility SVT is the abbreviation of K2 Mobility Sustainable Vehicle Technologies, an ongoing in process multinational and industrial organizational supported research project to develop integrated product and process models. K2 Mobility SVT's innovative goals is to provide information modeling, virtual product development, future development of information systems that will allow for continuous engineering with maximum flexibility and to create a central multidisciplinary networking based approach to facilitate holistic vehicle optimization with respect to specific and occasionally conflicting criteria. Moreover K2 Mobility SVT is propose to provide a platform for holistic vehicle optimization, integration of data, information and knowledge from the relevant development disciplines throughout product development and production planning, continuous access to reliable product data and information throughout the product life cycle and optimize human decision making. K2 Mobility SVT's Sub project "System Design & Optimisation" is divided in to seven different and important research project areas .i.e., Multidisciplinary concept framework, Integrated CAE Framework, Multidisciplinary Optimization and Robust Design, Integrated Mechatronic Vehicle, Future PLM, Digital Manufacturing Process Management and a Strategy Project – Semantic Integration Platform as shown in Figure 2.

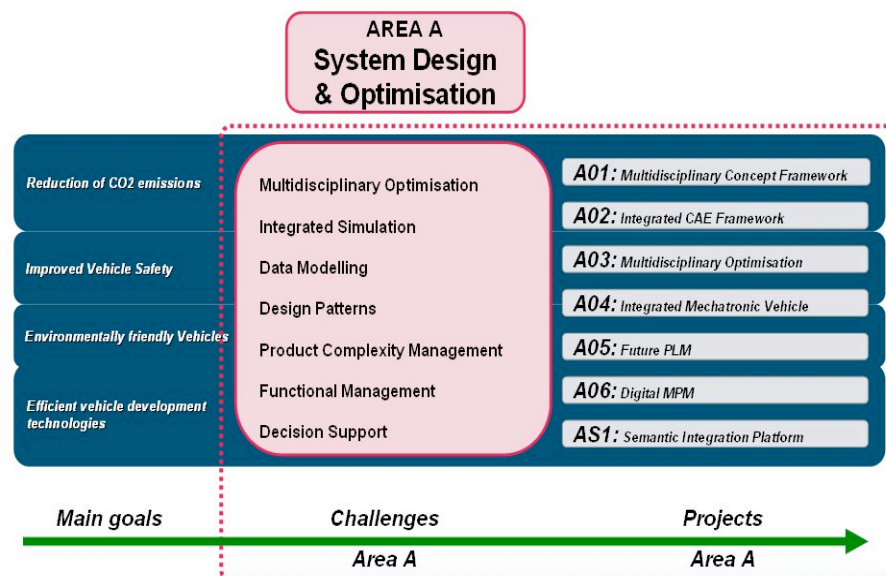


Figure 2: K2 Mobility SVT [81]

*Multidisciplinary concept framework* is proposed to meet several main demands i.e., ever decreasing timeframes for development in the automotive industry, steadily increasing diversity of products and product complexity, and the demand for the earliest possible predictions of the product based on a complete simulation model. The main goal of Multidisciplinary concept framework is to enable the assessment and comparison of the complete vehicle alternatives in the early phases of product development.

*Integrated CAE Framework* is proposed to integrate different processes, resources and data flows in a common and consistent data model, manage simulated data from various CAE disciplines and supply a set of functional and behavior oriented features.

*Multidisciplinary Optimization and Robust Design* is proposed to increase future virtual product development.

*Integrated Mechatronic Vehicle* is proposed to increasing proportion of electronically controlled actuators that cover the desired functions in an appropriate and preferably failure-free manner. Furthermore major expected jobs of Integrated Mechatronic Vehicle is to refine the simulation methods pertaining to the difficult and complex mechanical electrical couplings, increase the consistency of existing software and hardware connections and provide a holistic view of the vehicle at each moment of its development.

*Future Product Lifecycle Management (PLM)* is proposed to manage the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal.

*Digital Manufacturing Process Management (MPM)* is proposed to use digital methods to address the complex set of problems inherent to manufacturing planning. The major expected jobs of Digital MPM are to increasing time efficiency in manufacturing planning and assembly and to improve manufacturing planning process regarding design of manufacturing.

*Strategy Project – Semantic Integration Platform* is proposed to implement core topics, technologies and standards for the lifecycle oriented integration of CAx, PLM and E/E data sources through semantic technologies.

As this is ongoing research project, so the long term goals of K2 Mobility SVT are to improve the use of existing information sources and building machine readable knowledge base expert systems.

K2 Mobility SVT's Sub project "System Design & Optimization" has many relevancies to my research work like information modeling, information systems development, data integration, involvement of semantics and management of the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal which are also the key interests of my research and development work. Apart from all relevancies I cannot evaluate the weaknesses and strengths of this project because it's in the state of proposition not implementation and experimentation.

### 2.4.2 VIVACE: EU [82]

VIVACE is the abbreviation of “Value Improvement through virtual Aeronautical Collaborative Enterprise”. VIVACE is an in process ACARE’s (Advisory Council for Aeronautics Research in Europe) formulated project for fund cutting-edge research in the Aeronautical sector. VIVAC is an Aeronautical Collaborative Design Environment with associated Processes, Models and Methods to design an aircraft provide aeronautic supply chain in an extended enterprise, virtual products with all requested functionality and components in each phase of the product engineering life cycle [83]. VIVACE is aiming to define future European Aeronautical Collaborative Design Environment by producing processes, models and tools available for use in this environment in the second half of this decade. The major objectives of VIVACE are to reduce cost of air craft development by 5 percent, reduce lead time by 30 percent and reduce cost of development of new derivative gas turbine by 50 percent by re-engineering and optimization of the entire design process by modeling and simulation in an advanced concurrent engineering environment.

Basically VIVACE consists of three technical sub-projects .i.e., Virtual Aircraft, Virtual Engine, Advanced Capabilities. *Virtual Aircraft* is a specific global product work area which develops different elements of the aircraft and works around the products for design, modeling, interfacing and testing. *Virtual Engine* is a specific global product work area which develops different engine modules of the aircraft propulsion system and key areas of multidisciplinary optimization, knowledge management and collaborative enterprises. *Advanced Capabilities* is a key integrating work area which develops common tools, methodologies and guidelines to be shared amongst the developments of Virtual Aircraft and Virtual Engine.

Although VIVACE is PLM based product but not much relevant to my research work, the only thing which appeals to me for which I have added this as the part of my related research work is the process modeling & simulation in concurrent engineering environment and advanced capabilities. But still unfortunately not much explicit methodological based information about VIVACE is provided by authors, to have better look and analysis of their used methodologies.

### 2.4.3 Promise PLM: EU [84]

Promise PLM stands for Promise Product Life Cycle Management and Information Tracking Using Smart Embedded Systems. Promise-PLM gathers, processed and delivers relevant product information during the complete product life cycle. Promise-PLM also addresses existing gaps in the information flow, creates better understanding of total life cycle events and calculates total costs of products over the lifecycle. Moreover Promise-PLM allows information flow management to go beyond organizational barriers, securely to take advantage in effectively closing the product lifecycle information loops using the latest Information and Communication Technology (ICT) and seamlessly transforms that information into knowledge.

Promise –PLM consists of four main components .i.e., Product Embedded Information Device (PEID), Middleware, Product Data Knowledge Management (PDKM) and Decision Support

System (DSS). *PEID* is a storage device which stores product based information captured during each even of lifecycle. *Middleware* is device which provides management and communication between one or more PEIDs and existing enterprise backend software, moreover it also acts as networking layer of the Promise-PLM systems. *PDKM* is a device which integrates and manages information from all lifecycle phases of the product. *DSS* is a device which provides decision support for predictive maintenance, diagnosis and analysis of use patterns.

There are several many benefits of using Promise-PLM .i.e., lowers manufacturing costs and gives product a new life, provides real-time decision support through the internet, improves product design, reduces time-to-market and obtains accurate information about value parts and material.

Promise PLM has many relevancies to my research work like Promise PLM includes information gathering & processing, addresses existing gaps in the information flow and involves Knowledge Management and Decision Support System (DSS) which are also some of the key interests of my research and development work. But still unfortunately not much detailed explicit methodological based information about Promise PLM is provided by authors, to have better look and analysis of their used methodologies.

#### 2.4.4 Component-based Product Data Management system [63]

Authors have proposed adequately available, secure, stable, and reliable platform independent Web based Product Data Management System called Component-based Product Data Management system (CPDM) to take advantage in decreasing the cycle time for introducing new technology and products. The architecture of CPDM is based on Component Based Development (CBD), implemented using J2EE and consist of consists of three tiers .i.e., *Presentation tier*, *Business logic tier* and *Data tier* as shown in Figure 3. *Presentation tier* is designed to access the system through a Web browser, *Business logic tier* is implemented to handle the core PDM functionality and *Data tier* is composed of a database and vault for the physical files. CPDM is developed using Rational Unified Process (RUP) in a next-generation military system for a local military company. Developed CPDM is successfully deployed and several advantages in terms of productivity and competitiveness .i.e., reducing design hours, eliminating re-work due to design inconsistency, reducing time to market, and reducing parts inventory are obtained. CPDM architecture is implemented using following technologies .i.e., HTML, JSP, Servlets, EJB and JDBC as shown in Figure 3.

CPDM has relevancy with respect to the implementation designs of my research work because CPDM is based on three tier architecture which can be a good solution to adopt while implementing a PDM System. Used tools and technologies in the implementation of CPDM architecture can also be considered during the design and implementation phase of my research work. Moreover the goal of CPDM is to provide stable, reliable, platform independent Web based Product Data Management System which also somehow matches to my research goals.

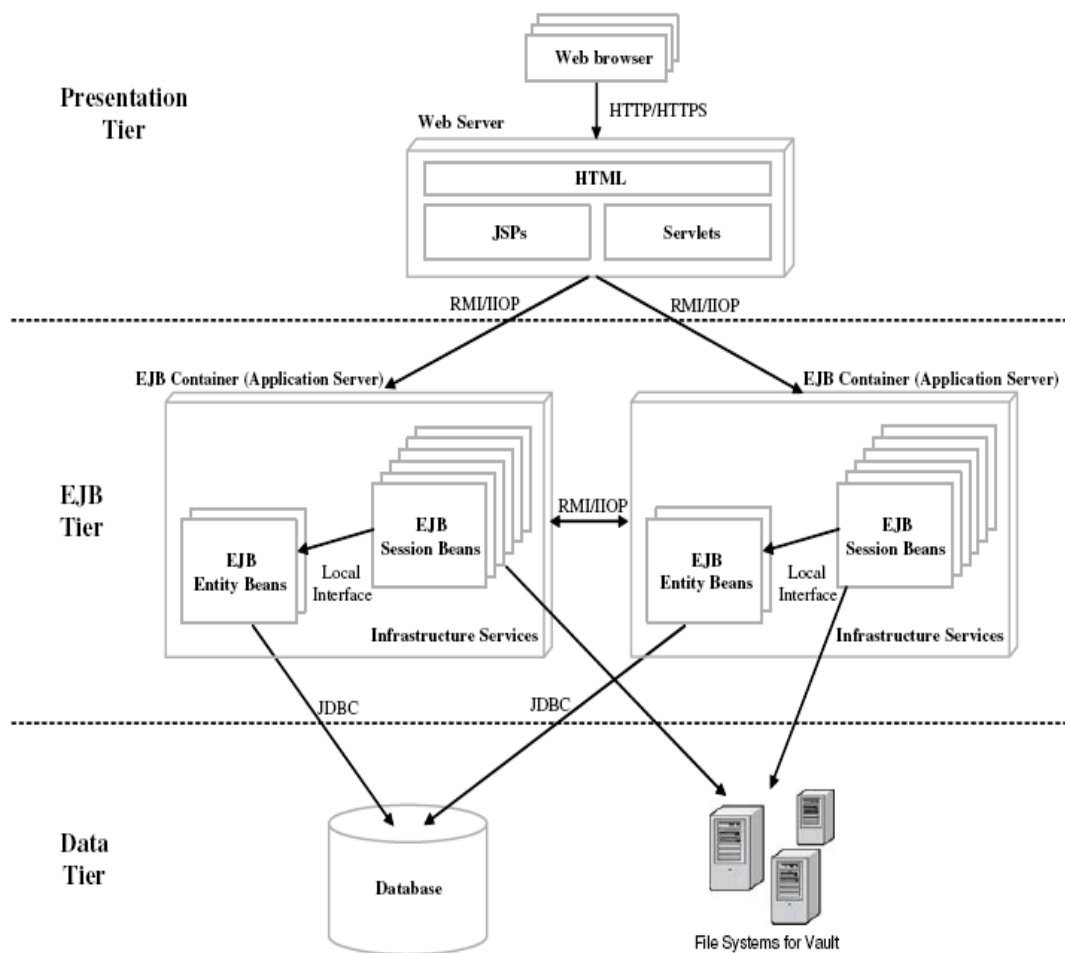


Figure 3: The architecture of CPDM [63]

#### 2.4.5 Web-based product data management (WPDM) [64]

Authors have proposed an easily extendable Web-based product data management (WPDM) to provide product database management through internet, access to users at different locations, especially those who are on different networks, optimize product cycle time and platform independency. The architecture of WPDM is based on three-tier client/server system i.e., Back-end database, Middle layer and Front-end design as shown in Figure 4. *Back-end database* is like a data vault used to store and manage product attribute data, documentary information, relationships between data., *Middle layer* centered control between back-end database and front-end user and *Front-end* design uses a web browser based front end to access WPDM. The designed architecture of WPDM is implemented using Java Servlets, JSP and JDBC and capable of handling user input and managing data in database.

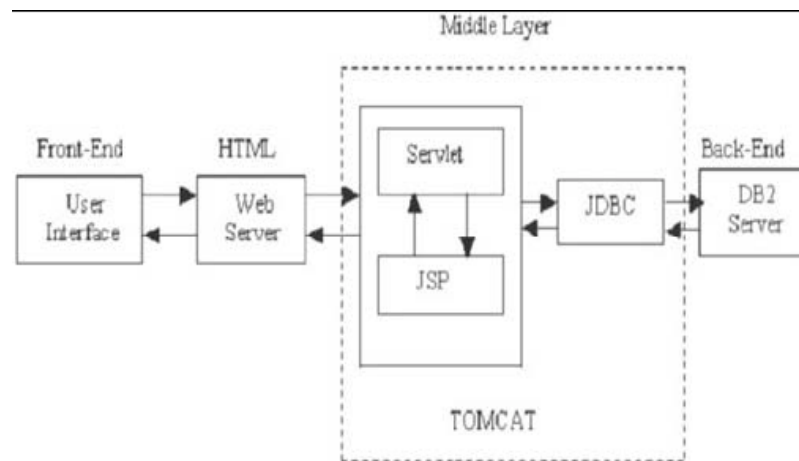


Figure 4: The architecture of WPDM [64]

WPDM has relevancy with respect to the implementation designs of my research work because likewise CPDM WPDM is also based on three tier architecture which can be a good solution to adopt while implementing a PDM System. Used tools and technologies in the implementation of WPDM architecture can also be considered during the design and implementation phase of my research work. Moreover the goal of WPDM is to provide platform independent web based product database management which also somehow matches to my research goals.

## 3 Designing Human Machine Interface

This chapter provides the detailed information about Designing Human Machine Interface (HMI) including HMI Ideologies, Design Principles, Design Patterns, Design Guidelines, Design Goals, Design Problems and RIA for Web based HMI Development.

### 3.1 Human Machine Interface (HMI)

Human Machine Interface (HMI) also renowned as Human Computer Interaction (HCI); is the study of design, evaluation and implementation of interactive computing systems for human use [39]. Designing High quality HCI design is difficult to implement because of many reasons [5] .i.e., market pressure of less time development, rapid functionality addition during development, excessive several iterations, competitive general purpose software and human behavior analysis.

Designing human computer interaction interface is an important and a complex task, but it could be simplified by decomposing task into subcomponents and maintaining relationships among those subcomponents. Task decomposition is a structured approach, applicable in both Software Engineering and Human Computer Interaction (HCI) fields depending on specific processes and design artifacts. Using design artifacts applications could be made for analysis and design by making the hand draw sketches to provide high level of logical design based on user requirements, usage scenarios and essential use cases. To design hand draw sketches there are some strategies to be followed .i.e., planning, sequential work flow, and levels of details.

### 3.2 HMI Ideologies

HMI mainly consists of three ideologies. .i.e., Design idea, Design activity, Design learning.

*Design Idea* is the tradition of arts where as *Design Activity* is the action performed by artifact(s). *Design learning* is the main and important ideology comprises of three important elements. .i.e., iDeas notebook, the iDeas blog, and iDeas wall. [40].

*iDeas notebook*: iDeas notebook extends the traditional notebook design by merging the physical and electronic input in to sketches. The main activities of iDeas notebook are electronic file based activities that are automatically created by notebook page photographs and while retaining the physical aspects of the idea log. There are some advantages of iDeas notebook that they are rapidly capturing rich amounts of data, ready at hand and familiar interaction, permitting designers to focus on tasks rather than tools, and accompanying users in the field and wherever, whenever design happens.

*iDeas blog*: iDeas blog serves as a digital store of collected and generated information. There are two sources of inspiration for iDeas blog .i.e., traditional blog and shared electronic portfolios. Traditional blog are primarily text based and require lots of interactions to add visual information where as shared electronic portfolios follows highly formalized way to

explicitly support visual and textual information. The iDeas blog is the extension of lightweight, automatic integration and archival of iDeas notebook inputs.

*iDeas wall*: iDeas wall provides an interactive surface to present and create ideas and general purpose contents. It is a vertical display surface with direct manipulation capability that can afford collocated group interactions, including the presentation interaction style and the whiteboard interaction style of brainstorming sessions. The iDeas wall provides three methods for users to create and import contents .i.e., can sketch and write on the wall as they would on a whiteboard, can import content from the iDeas blog and can bring up an iDeas notebook page directly by using the pen as a command device.

### 3.3 HMI Design Principles

There are four major Human Machine Interface design principles. i.e., Cooperation, Experimentation, Contextualization and Iteration [41] [42].

*Cooperation*: Cooperation plays a vital role in software project development. The most important and primitive principle of design process is the cooperation between both developers and the end users. Because in the design process with respect to the participatory design point of view there exists an uncommon principle. i.e., presenting the same issues with completely different perspectives and dimensions.

*Experimentation*: Generally experimentation is performed in the middle of recently acquired possibilities and the currently existing conditions. To assure that the present conditions are in conjunction with new ideas and supported by two primitive principles .i.e., concretization and contextualization of design, Principles are in associated with the above mentioned visions performing experiments with visions and hand on experience.

*Contextualization*: Design hooks its initial point with a particular configuration in which new computer based applications put into practice. Participatory design emphasizes on situations based on the implementation of iterative designs. The design composition of use is tied up with numerous social and technical issues. Generally participatory design of the development will specifically includes different kinds of participants i.e. Users, Managers and the design developers.

*Iteration*: In design process, I should hang on to some issues which are not yet revealed, which are visioning the future product from design point of view and the construction of work from use point of view. But participatory design puts a controversial statement in accomplishing the same by making use of artifacts i.e. Prototype. Designers with cooperation will make use of the artifacts as a source for delegation of work. Participatory design also ends up with a controversial statement for trivial division of work in the process of development, which pleads overlap among the members of analysis, design and realization groups.



### 3.4 HMI Design Patterns

Like software engineering design patterns there are also some graphical user interface design patterns [43] .i.e., Window Per Task, Interaction Style, Explorable Interface, Conversational Text, Selection, Form, Direct Manipulation, Limited Selection Size, Ephemeral Feedback, Disabled Irrelevant Things, Supplementary Window and Step-by-Step Instructions [44]. These patterns help designers in analyzing already designed graphical interface and designing a user friendly and required on demand graphical machine interface e.g. helps in organizing the complete graphical user interface into different screens (Window Per Task), develops user machine interaction style (Direct Manipulation), minimizes the cost by identifying exceptions (Explorable Interface), provides textual input information of designed interface's commands (Conversational Text), describes interaction style to choose options from provided list of options (Selection), describes discrete structures on screen (Forms), structures set of selections (Limited Selection), provides the information about the natural flow of the interface (Ephemeral Feedback), guides in identifying and removing irrelevant interface elements (Disable Irrelevant Things), provide information about supplementary windows (Supplementary Windows), helps designer in sequencing set of actions (Step by Step Instructions). [45, 46, 47]

### 3.5 HMI Design Goals and Guidelines

A successful design interface has three main primary design goals. i.e., High resolution, Clean screen and Fluid interaction. [89]

*High Resolution:* An interactive design must display required quantitative material including images and application windows at an appropriate resolution for the user standing at the board (a resolution comparable to workstation monitors). To provide a smooth wall like environment, it needs to be flat, continuous surfaced and without physical seam interruption. In the future most probably large high-resolution based flat panels will be available.

*Clean Screen:* There is a striking difference between the visual look of a paper based project and a standard GUI screen. The GUI intersperses the user's content with a profusion of visual "widgets" for control: window boundaries, title bars, scroll bars, tool bars, icon trays, view selection buttons, and many more. A whiteboard or project wall, on the other hand, provides a uniform blank surface whose content is the user's marks or posted pieces of paper. To capture the feel of a wall we want to reserve the visual space for content, with an absolute minimum of visual distraction associated with interaction mechanics. We organize our display as an arbitrary collection of opaque and translucent "sheets," which can be created, drawn on, and moved independently. Sheets have only content on them, without visual affordances for actions.

*Fluid Interaction:* Along with the visual clutter of GUI interfaces, there are continual interruptions to the flow of activity. Dialog boxes pop up, windows and tools are selected, object handles of various kinds appear and are grabbed, and so on. To some degree this is inevitable to provide complex functionality. The traditional workstation GUI is oriented toward facilitating the speed of highly differentiated actions done by experienced users whose

attention focus is entirely on the current computer activity. But users of a wall display are often engaged in simultaneous conversation with people in the room, so their focus on the board is episodic. They do not have the additional cognitive capacity to cope with complex state differences, or have a high tolerance for having their attention distracted to the interaction with the board instead of with the people.

A successful design interface can be implemented using the following guidelines like design should be presentable, principles should be applied, prepared according to the projected proposal, specified according to the requirements, should be evaluated, assessed by mental work load, mock-ups should be implemented, should be flexible enough to adopt rapidly prototyped changes and modifications, design should be iterative and use case modeling should be used with the identification of user interface elements etc.

### 3.6 RIA for Web based HMI Development

The term "Rich Internet Application" was introduced in a white paper of March 2002 by Macromedia. Rich Internet Applications (RIA) are web applications those have the features and functionalities of traditional desktop applications as well as the web applications. Traditional web applications centered all activities around client server architecture with a thin client where as RIA typically transfer the processing necessary for the user interface to the web client but keep the bulk of the data (i.e., maintaining the state of the program) back on the application server.

RIA share one characteristic; introduces an intermediate layer of code, often called a *Client Engine*, between the user and the server. This client engine is usually downloaded as part of the instantiation of the application, and may be supplemented by further code downloads as use of the application progresses. The client engine acts as an extension of the browser, and usually takes over responsibility for rendering the application's user interface and for server communication. Using Client Engine RIA become more rich, more responsive, balanced, Asynchronous and efficient. [48]

*Richer:* Available user interface behaviors are not obtainable using only HTML widgets available to standard browser based Web applications. This richer functionality may include anything that can be implemented in the technology being used on the client side, including drag and drop, using a slider to change data, calculations performed only by the client and which do not need to be sent back to the server.

*More responsive:* The interface behaviors are typically much more responsive than those of a standard Web browser that must always interact with a remote server. The most sophisticated examples of RIA is, it exhibits a look and feel of a desktop environment level. Using a client engine can also produce other performance benefits.

*Balanced:* The demand for client and server computing resources is better balanced, so that the Web server needs not to be the working horse like in traditional Web application. This frees server resources and allows the same server hardware to handle more client sessions concurrently.

*Asynchronous:* The client engine can interact with the server without waiting for the user to perform an interface action such as clicking on a button or link. This allows the user to view and interact with the page asynchronously from the client engine's communication with the server. This option allows RIA designers to move data between the client and the server without making the user wait. Perhaps the most common application of this is pre-fetching data, in which an application anticipates a future need for certain data and downloads it to the client before the user requests it, thereby speeding up a subsequent response. Google Maps uses this technique to load adjacent map segments to the client before the user scrolls them into view.

*Efficient:* The network traffic may also be significantly reduced because an application-specific client engine can be more intelligent than a standard Web browser while deciding which data needs to be exchanged with servers. This can speed up the individual requests or responses because less data is being transferred for each interaction, and overall network load is reduced. However, over-use of asynchronous calls and pre-fetching techniques can neutralize or even reverse this potential benefit. Because the code cannot anticipate exactly what every user will do next, it is common for such techniques to download extra data, not all of which is actually needed, to many or all clients.

There are some RIA based technologies. i.e., FLEX (Adobe), AJAX, OpenLaszlo and Silverlight (Microsoft).

*Flex* is a highly productive, free open source framework for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops, and operating systems. While Flex applications can be built using only the free Flex SDK, developers can use Adobe Flex Builder™ 3 software to dramatically accelerate development [49]. Adobe Flex is a collection of technologies released by Adobe Systems for the development and deployment of cross platform rich Internet applications based on the proprietary Adobe Flash platform [50].

*AJAX* is a free framework for quickly creating efficient and interactive Web applications those works across all popular browsers. AJAX stands for Asynchronous JavaScript and XML. AJAX is a type of programming become popular in 2005 by Google. It is not a new programming language, but a new way to use existing standards. A primary characteristic is the increased responsiveness and interactivity of web pages achieved by exchanging small amounts of data with the server "behind the scenes" so that entire web pages do not have to be reloaded each time, there is a need to fetch data from the server. [51]

*OpenLaszlo* is an open source platform for the development and delivery of rich Internet applications. It is released under the Open Source Initiative-certified Common Public License. Laszlo applications can be deployed as traditional Java servlets, which are compiled and returned to the browser dynamically. This method requires that the web server be running the OpenLaszlo server. OpenLaszlo was originally called the Laszlo Presentation Server (LPS). [57]

*Microsoft Silverlight* is a web browser plug-in that provides support for rich internet applications such as animation, vector graphics and audio-video playback. Silverlight competes with products such as Adobe Flash, Adobe Flex, Adobe Shockwave, JavaFX, and Apple QuickTime. Version 2.0 brought improved interactivity and support for .NET languages and development tools.

Silverlight provides a retained mode graphics system, similar to WPF and integrates multimedia, graphics, animations and interactivity into a single runtime. It is being designed to work in concert with XAML and is scriptable with JavaScript. XAML can be used for marking up the vector graphics and animations. Textual content created with Silverlight would be more searchable and indexable than that created with Flash as it is not compiled, but represented as text (XAML). Silverlight can also be used to create Windows Sidebar gadgets [56].

### 3.7 HMI: Related Work

In the domain of human machine interface design many graphical machine interfaces for product data management based products are designed and in use. At the moment I am considering the some of the most successful PDM product interfaces as the part of relevant literature.

During detailed review, I put emphasis on the validity of the proposed solutions in literature and presented methodologies. Moreover I am also presenting the relevancy of chosen, reviewed and described related work to my research and development work.

#### 3.7.1 Windchill [85]

Windchill is a web based product data management application by PTC, providing services in content and process management of organizational technical and managerial data, as shown in figure [85]. Windchill is capable of decreasing product development time through efficient collaboration, reducing errors by automating processes and driving conformity to corporate standards, reducing scrap and rework by automatically sharing product data with downstream manufacturing systems and engineers, increasing efficiency by enabling engineers to quickly find and manage multiple forms of digital product content, eliminating mistakes associated with duplicate data, incomplete data, or manual data transfers via a single, secure repository for all product content and lowering your total cost of ownership (TCO) and technology risk by reducing the number of systems and databases to maintain and administer.

With respect to my research and development point of view all provided features in the Windchill are very helpful in writing design requirements for a PDM product but in my opinion there are some features which are still required to be improved like graphical machine interface and search mechanism.

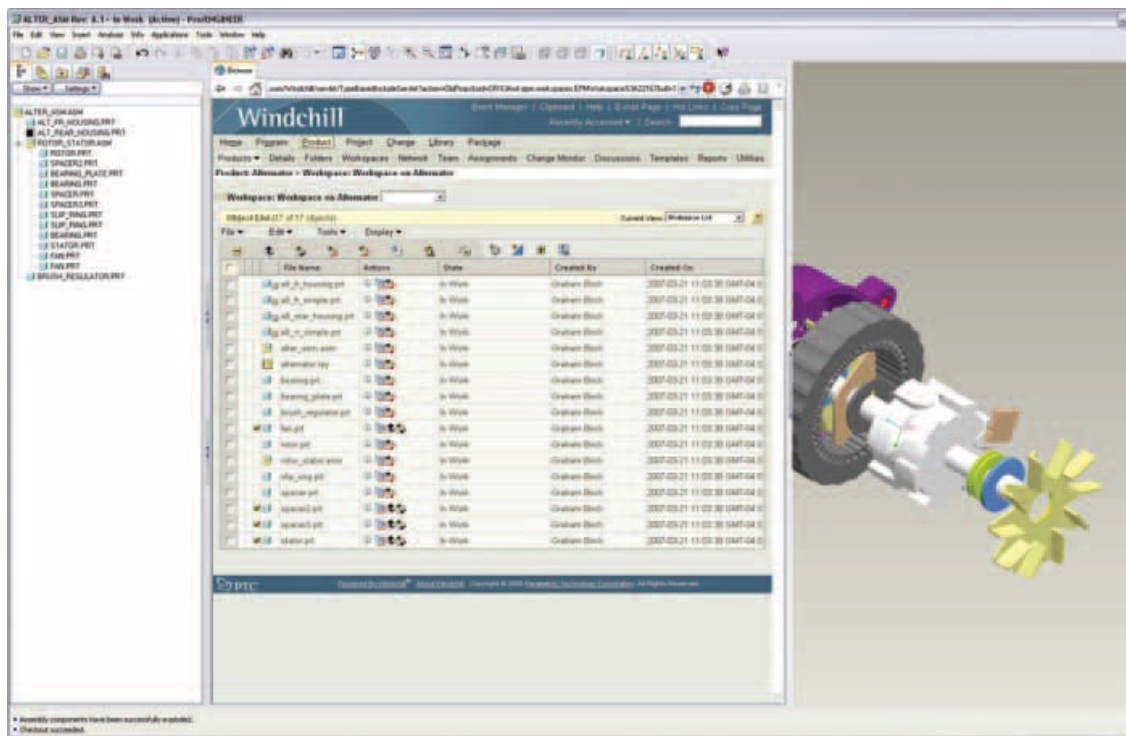


Figure 5: Windchill by PTC [85]

### 3.7.2 CIM Database [86]

CIM Database is a desktop and web based high performance PDM system to manage data and support product creation and process implementation. Moreover CIM Database is a complete and protected data management system for a range of centralized functions like selectable search, CAD systems, product and organizational data management and electronic data interchange (EDI).

With respect to my research point of view all provided features in the CIM Database are very helpful in writing design requirements for a product data management based product but in my opinion there are some features which are still required to be improved like graphical machine interface and search mechanism.

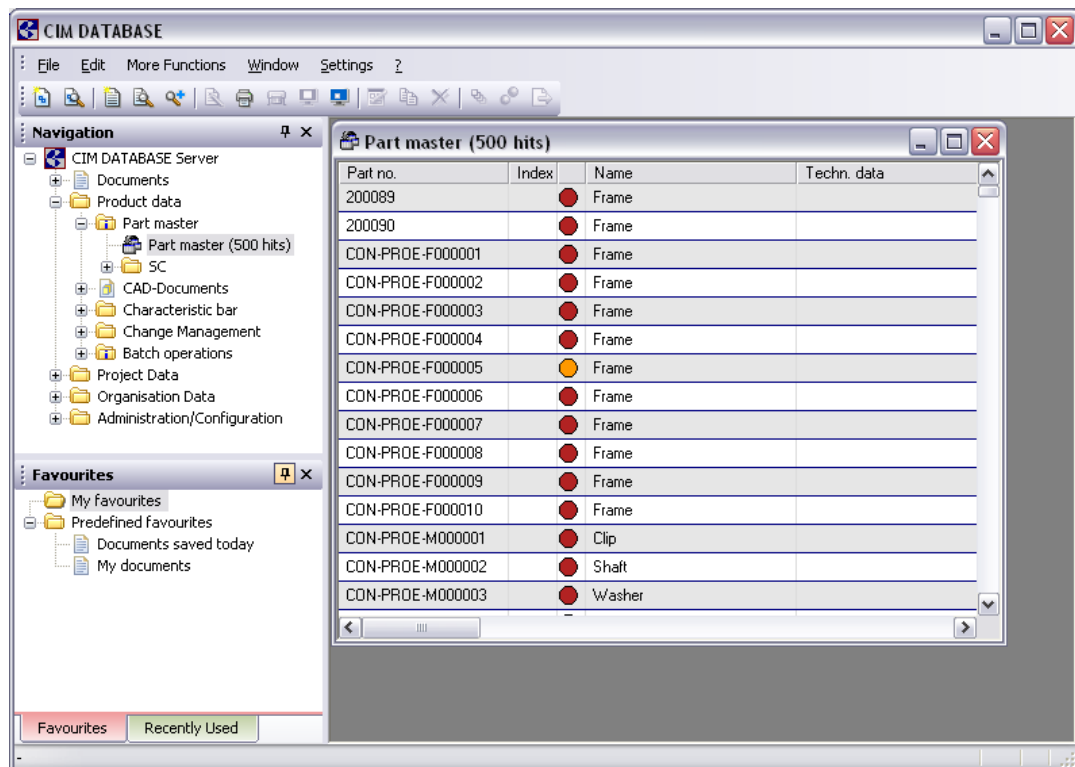


Figure 6: CIM Database [86]

## 4 Web based Information Processing

This chapter provides the detailed information about Web based information modeling and processing including information about Web, Semantic Web, Ontology and some Semantic Web based information processing Approaches.

### 4.1 Web

World Wide Web [2] is a global information sharing and communication system made up of three standards Uniform Resource Identifier (URL), Hypertext Transfer Protocol (HTTP) and Hypertext Mark-up Language (HTML) by Tim Berners-Lee to effectively store, communicate and share different forms of information. The Information is provided over the web in text, image, audio and video formats using HTML, considered unconventional in defining and formalizing the meaning of the context.

Most of the information is structured only inside the available databases over the web and due to this it is quite easy to go for scattered extensive information by looking into bookmarked web pages but quite difficult to extract a piece of needed information. Although some search engines and screen scrapers are invented, search engine uses full text query to search information but can only return unstructured contents not the actual structured information stored in database on web where as screen scrapers extracts and repurpose fragments from web pages but insufficient in creating a rich multi domain information environment [28]. Most of the search engines are not satisfactory because they requires excessive manual preprocessing e.g. designing a schema, cleaning raw data, manually classifying documents into a taxonomy and manual post processing e.g. browsing through large result lists with too many irrelevant items [29]. To increase the integration and interoperability over the web the concept of “Web Service” was introduced. Due to the dynamic nature web services became very famous in industry in short time but with the passage of time due to the heavily increase in number of web services end-to-end service authentication, authorization, data integrity and confidentiality problems were identified which are still alive and not handled by existing web technologies [30].

HTML based documents are formatted in way that these cannot be processed because these are only available in readable format. This deficiency leads to the problems of searching, extracting, maintaining, uncovering and viewing the knowledge-based information over the web. More over this format deficiency becomes the major cause of some semantic based problems and the need of some other approach which will publish data over the web in not only the readable but also in process able format. Because, if data will be available in Meta data [6] (read and process able data format) both read and process able formats, then it will improve the process of search, extraction and maintenance of data over the web.

### 4.2 Web to Semantic Web

To cope with the currently existing web problems. i.e., Information filtration, security, confidentiality and augmentation of meaningful contents in mark-up presentation over the web

a semantic based solution “Semantic Web” was proposed by Tim Berners Lee [5]. Semantic Web is a mechanism of presenting information over the web in a format so that human being as well as machines can understand the semantic of context. Semantic web is a mesh of information that can be linked up in a way, so that it can easily be processed by machines [1] and aim to produce technologies capable of reasoning on semi structured information [9].

The semantic web is an intelligent incarnation and advancement in World Wide Web to collect, manipulate and annotate information independently by providing effective access to the information. Semantic web provides categorization and uniform access to resources, promotes the transformation of World Wide Web in to semantically modeled knowledge representation systems and common framework which allows data to be shared and reused [31]. Semantic web also gives the concept of semantic based web services to provide solutions to the problems of dynamically composed service based applications.

Currently, semantic web is standing on a very important building block Ontology [3]. Moreover semantic web aims of providing information in machine processable semantic models which assigns information resources to classes whose meaning is defined in ontologies [32], a collection of interrelated semantic based concepts. Ontology is the explicit representation and description of already available finite sets of terms and concepts used to make the abstract model of a particular domain, described in detail in Chapter 2 Product Data Management. Moreover, along with the processing ability semantic web agent is capable of communicating, receiving and transferring information to different sources (agent or human).

### **4.3 Current Challenges of Semantic Web**

Currently it is not possible to completely search and extract desired concrete results using full text queries, because there is no as such mechanism exists which can extract the actual semantic from full text query and then look into the data warehouse for the particular knowledge based information. Currently two kind of searching techniques exists; full text search and unambiguous search. Full text search processes natural language based query to retrieve information like Google [10] where as Unambiguous search is based on data whose semantic is already defined in the system, for example we are looking for some person on web and system ask to enter person’s personal information like name, age and address and then using this particular information searching is performed like Reunion [8]. This probability of getting more concrete results from unambiguous search is very high as compared to full text search because there is no such mechanism existing which can extract the actual semantic from full text search query and then look into the warehouse for the particular information based on extracted meaning. To full fill this current need and implement the new idea of data formatting Semantic web is introduced [7].

### **4.4 Goals of Semantic Web**

The main and currently not achieved goal of semantic web is to structure the meaningful contents of unstructured published data over web to take advantage in improving the search process [5] and to involve knowledge management in making some more advanced knowledge modeled management systems.



No doubt semantic web using ontology have contributed the in progress of web but still there are some limitations and due to those semantic web is currently not succeeded in attaining the actual goal of completely structuring the information over the web which can be processed by machines and making advanced knowledge modeled system. The need is to enhance the concept of ontology with respect to development point of view because all the theories can be fruitful if the implementation is possible.

## 4.5 Semantic Oriented Approaches

In the domain of semantic web many methods have been introduced by many researchers, which not only provides the values to web application but also to desktop applications. So I am considering the most relevant to our research and development work as the part of our related work.

During detailed review, I put emphasis on the validity of the proposed solutions in literature and presented methodologies. Moreover I am also presenting the relevancy of chosen, reviewed and described related work to my research and development work.

### 4.5.1 Ontology

Ontology is a main building block of Semantic Web to provide the information in machine processable semantic models and produce semantically modeled knowledge representation systems. Ontology is playing a vital role in solving the existing web problems by producing semantic aware solutions. Ontology makes machines capable of understanding the semantic languages that humans use and understand by producing the abstract modeled representation of already defined finite sets of terms and concepts involved in intelligent information integration and knowledge management [33].

Ontology is basically categorized in three different categories. i.e., *Natural Language Ontology (NLO)*, *Domain Ontology (DO)* and *Ontology Instance (OI)* to provide relationships between generated lexical tokens of statements based on natural language, knowledge of a particular domain and to generate automatic object based web pages [14]. Ontologies are constructed and connected to each other in a decentralized manner to clearly express semantic contents and arrange semantic boundaries to find out required needed information [34].

Ontology construction is a highly relevant research issue that depends on the extraction of information from web and emergence of ontologies. Natural language based information is treated as the input to the ontology construction process, which parses the text in nouns and verbs. Nouns are represented as “Classes” and verbs as “Properties” containing values, relationships with other properties and some constraints. Classes are further divided in main and sub class categories maintained in taxonomy hierarchy. The size of ontology varies due to the increase in number of classes and instances.

Ontologies can be made manually from scratch, by extracting information from web and by merging already existing ontologies in to new ontologies. But this manual process sometimes becomes very complex and time consuming especially when dealing with the large amount of

data. Moreover, to support the process of semantic enrichment reengineering for the building of web consisting of metadata depends on the proliferation of ontologies and relational metadata. This requires high production of metadata at high speed and low cost.

So in these cases machine learning approaches can be very helpful in generating ontologies automatically because it provides real time schemes like classification rules, instance based learning, numeric predictions, clustering, Bayesian networks and decision trees which can be very helpful in the generations of ontologies.

Ontology development is an iterative process based on six main activities .i.e., Determine Scope, Enumerate Terms, Classify Ontology, Define Classes, Define Properties and Create Instances as shown in Figure. 8. In the beginning of ontology development process it is very important to determine the scope otherwise it will be very time and effort consuming. Then enumerated terms should be identified to classify ontologies within their respective types. Classes and their respective properties along with their relationships and constraints are defined using identified enumerated terms. In the end only the instances are created and used. To implement ontology development process some experience, a powerful user friendly ontology supporting tool and communication between domain experts and developers is required [20].

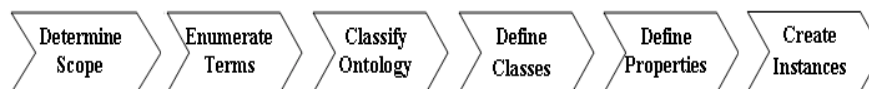


Figure 7 Ontology development activities [20]

First step in building ontologies is to create the nodes and edges, once the concepts and relationships of a graph based ontology are constructed then next step is to quantify the strengths of semantic relationships [29]. Ontologies can be constructed manually and automatically by using some ontology supporting languages. i.e., XML (eXtensible Mark-up Language) [16], RDF (Resource Description Framework) [1] and OWL (Web Ontology Language) [17] offering ways of more explicitly structuring and richly annotating Web pages.

XML (eXtensible Markup Language) [16], is one of the fundamental contributions towards middleware technologies [13]. It is a markup Meta language that allows sharing of information between different applications through markup, structure and transformation. As the major contribution towards semantic web, XML using Data Type Definitions (DTD) and depending on data types, attributes, both internal and external elements structure documents and provide syntax serialization and abbreviation for data modeling [4]. The XML schema restricts the syntax to be only used for the structured documents, because of this XML has two main problems in process of information extraction; first it is without semantic and second arbitrary naming and structuring of elements.

RDF (Resource Description Framework) [59], a URL based syntax data representation, provide a secure and reliable mechanism for the exchange of metadata between web applications. RDF process metadata by making an abstract data model based on three object

types attributes Resource, Property and Statement. Resource is an expression, Property is an attribute to describe a resource where as the statement is a resource having some property and value. RDF use three containers Object bag, Sequence and Alternative to keep multiple available and alternative values arranged in an order in resources and properties. Bag contain resources, Sequence contain resources along with their properties having single or multiple values arranged in order and Alternative contain resources having alternate value(s) of a property [12]. RDF provide syntax serialization and abbreviation for RDF data modeling. Serialized syntax express the full capabilities of data modeling in a very regular fashion and abbreviated syntax include additional constructs to provide a more compact form in representing a subset of the data model. RDF is more useful than XML in ontology construction because it provide semantic based features for data including domain independency, vocabulary and privileges in defining terminologies used in schema language. Furthermore it also provide syntax based on reification (statements about statements), data types, attributes, nesting, elements, element types, element container and no restrictions in structuring document like XML. RDF has its own grammar but not complete, rely on the support of XML to full fill the need. Moreover RDF modeling mechanism is insufficient in expressing various logical statements [4].

OWL (Web Ontology Language) [17], derived from American DARPA Agent Markup Language (DAML) [18]. OWL is based on ontology, inference and European Ontology Interchange Language (OIL) [19], claim to be an extension in RDF in expressing logical statements. It is rich in vocabulary because it not only describes classes and properties but also provide the concept of namespace, import, cardinality relationship between the classes and enumerated classes. OWL has some limitations like only one Namespace per project is allowed, Import is not currently supported, no database backend and Multi-User support and a few OWL Language features are missing, [20].

The development ontology driven applications is slowed down due to some limitations and principal problems that are

- Natural language parsers used to parse the information to construct the ontologies are limited because they can only work over a single statement at a time [13].
- Existing methodologies of structuring ontologies are in sufficient and need to be improved because now it is quite impossible to define the boundaries of ontology based particular domain's abstract model and automatically handle the increase in size of ontology due to the increase in number of classes and instances.
- Creating ontologies manually is a time consuming process which becomes very complex when there is a large amount of data to create large number of ontologies. To take advantage in creating large number of ontologies by reducing the complexity and time an automatic ontology creation mechanism is required. Some mechanisms are already proposed and implemented to create ontologies automatically but those are insufficient and less qualitative. While creating nouns based classes using existing automatic ontology creation mechanism, it is quite impossible to identify the possible existing relationships between classes to draw the taxonomy hierarchy [35].

Furthermore it is also quite impossible to perform automatic emergence of ontologies to create new ontologies [37].

- Currently available ontology validators are restricted and not capable of validating all kind of ontologies e.g. based on complex inheritance relationship [30].
- Domain specific ontologies are highly dependent on the domain of the application and because of this dependency domain specific ontology's contained specific senses are not possible to find in general purpose ontology [36].
- The process of semantic enrichment reengineering for web development consists of relational metadata required to be developed at high speed and in low cost depending on proliferation of ontologies, which is currently also not possible.
- Handle the dynamically raised calculations caused by the comparison of big complexities of similar ontologies is also not possible [37].

#### 4.5.2 Language Technology

Language Technology is a linguistic based field of computer science, also called as Human Language Technology or Natural Language Processing [69]. Moreover Language technology is about to make machine capable of read, listen, understand and analyze human (natural) language.

The main objective of Languages technology is to teach machines, how to communicate and help humans by communicating (listening and speaking) with them [70].

Although existing Language Technology based desktop, web and semantic web applications are not able to complete listen, understand and analyze human (natural) language based instructions by using human language based knowledge. But still the community of Language Technology is struggling in producing such intelligent system which will be based on natural language interface capable of communicating with human in natural language by listening, understanding and analyzing the context and semantic.

There are many ongoing language technology based research projects contributing in cross lingual Information, knowledge management, multilingual document production and multilingual natural communication like TWENTYONE, MULINEX, MIETTA, OLIVE, PARADIME, Whiteboard, GETESS, TG/2 and TEMSIS etc.

*TWENTYONE; Disclosure and dissemination of documents on sustainable development* is a European Commission sponsored project for environmental organization by locating and automatic translation of information [71].

*MULINEX; Multilingual Indexing, Navigation and Editing Extensions for the WWW* is a European Commission sponsored project for the efficient use of multilingual online

information by providing the combinations of the newest Information Retrieval Technology with advanced Language Technology to improve search and navigation in the WWW [72].

*MIETTA; Multilingual Tourist Information on the World Wide Web* is a European Commission sponsored project for providing flexible cross-lingual access to tourist information in the web, combination of class-based and free text search, Simultaneous access to heterogeneous data sources, presentation of information in different languages through machine translation and multilingual generation and Advanced localization and web-technology for simplified maintenance [73].

*OLIVE; Retrieval of video material based on speech-recognition* is a European Commission sponsored project for providing Video-retrieval on the basis of indices constructed from transcribed speech, three step retrieval process from textual index terms, cross-lingual retrieval on the basis of document translation and query translation, additional background material integrated for improved speech recognition and for further disclosure of video contents [74].

*PARADIME; Intelligent Extraction of Information from On-line Documents* is a European Commission sponsored project for providing intelligent retrieval of information from German documents, management of vast sources of linguistic knowledge, new functions with machine learning processes and integration of graphic visualization techniques, server architecture and access to the Internet [75].

*Whiteboard; Multilevel Annotation for Dynamic Free Text Processing* is Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMB+F) sponsored project for designing, implementing, investigating and evaluating a new system architecture that facilitates the combination of different language technologies for a range of practical applications [76].

*GETESS; German Text Exploitation and Search System* is Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMB+F) sponsored project for the development of an intelligent work tool for researching information [77].

*TG/2; Practical generation of natural language text*, TG/2 stands for a new generation of template-based generators and designed to organize a classical production system, separating the generation rules from their interpreter. TG/2 can provide solutions for limited sublanguages that are tuned towards the domain, can quickly be accommodated to new tasks, can be integrated smoothly with 'deep' generation processes, can reuse generated substrings for alternative formulations and can be parameterized to produce the preferred formulations[78].

*TEMSIS; Transnational Environmental Management Support & Information System* is a European Commission sponsored project for increasing availability of up-to-date data in the information age is of limited use without adequate presentation. TEMSIS can provide On-line access to current measurements, can select Information according to the user's requirements, can generate environmental reports in the user's language (German, French), can Build data

and parameter dependent structure, can combine extemporaneous and fixed text components and can represent, comprise and summaries collected data [79].

*COSMA: Automated Appointment Scheduling by E-Mail* provides concepts of automatic planned and manage Appointments using software agents to save Time and money savings during the arrangement of business appointments by communicating with a number of partners via e-mail, distributing client/server solution in the Internet, robusting dialogue control with intelligent failure handling and Shallow-Parsing techniques for the analysis of the German language [80].

Language Technology aims to teach machines, how to communicate and help humans by communicating (listening and speaking) with them which is one of the key interests and very important goal of my research and development work. Moreover the followed or proposed mythologies by the Language Technology community in fulfilling their goals of research can also be very helpful in my research and development work.

#### 4.5.3 Semantic desktop: Personal Information Model (PIM) [22]

Authors promoted the idea of stepping into user's mental model by implementing Personal Information Model (PIM). PIM is designed to improve the process for the identification of documents and retrieval of no unnecessary document. The design is based on ontologies and classes, the relationships of classes and ontologies are predefined and the information can be accessed using RDF graphs. Four rules based on forward changing principle are defined to retrieve the information, this information is divided in three parts Author (single or team), Relevant project, and Relevant solution. The system works in the following way. First query runs with aiming to find out the project and if project is found then it is moved to find out the related documents of the project.

The proposed architecture mainly consists of three main components Receiver, Interpreter and Analyzer. Receiver is used to provide index services and obtain the information about the structure of indexed files with the help of so-called brainFiller<sup>1</sup>. Interpreter first retrieved Information (structure / unstructured) using full text search and then uses so called LiveLink<sup>2</sup> to structure the contents of obtained information with the help of manual annotation and meta data, to store and retrieve contents based on their properties and preferences. As the last step Analyzer queries using Jena inference engine<sup>3</sup> on created RDF models to infer runs and also use F-Logic<sup>4</sup> to integrate rules.

To take the advantage of proposed approach by sharing the information from search, four case scenarios are designed: Local search, group search, closed community and open community. Local search scenario only deals with the search mechanism can only be applied to a personal

---

<sup>1</sup> Creates term vector for file and cluster documents with in their respective place (folder).

<sup>2</sup> A database driven web based Knowledge Management system [1.23]

<sup>3</sup> An open source RDFS and OWL based framework for building Semantic Web applications [1.24]

<sup>4</sup> An extension of classical predicate logic, based on object oriented framework [1.25]

desktop, group search can be applied with in a particular network domain, closed community consists on number of users having different roles but same topic where as open search consists on the users with different roles and different topics.

PIM is designed to improve the process for the identification of documents by improving the search mechanism (information processing) which is one of the key interests of my research and development work as well.

#### 4.5.4 Reisewissen [11]

An approach is proposed to identify potential relevant knowledge sources and provide quality services by semantically connecting, organizing and sharing the currently isolated pieces of information in an online portal to anticipating customer behavior. Proposed approach is implemented using semantic web technologies in a project Reisewissen, a hotel recommendation engine and travel information system. The design of Reisewissen is composed of three main components Data Connectors (DC), Evaluation Framework (EF) and Evaluation Engine (EE). Data connectors are used to provide transparency to data sources and transformation of data from heterogeneous to common data format (RDF and Java objects), more over it also provides the caching and fetching of data. Evaluation Framework is a workbench to test the quality of data and rules by providing functions and filters to map resources and return result in decisive format (Boolean or float value) and Evaluation Engine combines individual filters to rank and filter information by their weighting and yielding an overall score.

Information is obtained using simple object access protocol (SOAP) based web services and stored in both RDF and non RDF formats, which then matched to find out the desired result. Data stored in RDF format is based on developed ontologies mapped between database and RDF triples<sup>5</sup>. More over Reisewissen uses Prolog to capture expert's knowledge which can be formalized and generate new data by implementing the customer request in evaluator encapsulated rules. Data is matched semantically by combining data properties to Ontology and similarities between two concepts are determined by distance reflecting their respective positions in hierarchy and as the result list of selected results are generated to customer.

Reisewissen is designed to provide quality services by semantically connecting, organizing and sharing the currently isolated pieces of information which is one of the key interests of my research and development work as well.

#### 4.5.5 Test Application [26]

An approach based on the semantic web concepts for publishing information on web without inserting into relational database by making a flexible reasoning Ontology based system and to take advantage in improving already existing semantic search mechanism is proposed.

---

<sup>5</sup> A triple consists of a subject, predicate, and an object [1.21].

Each web object is referred as Unified Resource Identifier (URI) and to provide context in machine interpretable way every key principle is implemented in RDF and OWL format. Information is first converted in to ontologies, then using a translator application transformed into machine understandable format and then saved in to RDF files. These RDF files can easily be published on web as well as retrieved by any search engine based on digital assistants. The proposed approach is implemented in real time software application called Test Application. Test Application is developed to publish the information in to machine interpretable form for mobile phones.

Test Application is designed to provide quality services by semantically connecting, organizing and sharing the currently isolated pieces of information which is one of the key interests of my research and development work as well.

#### 4.5.6 Meta Data Search Layer [27]

Authors have discussed a successful process for metadata search based on three questions for information extraction: What user needs?, Where it lies? and How it can be achieved?. The targeted objective is to identify the location from set of locations contained by a document and avoid looking into non-specific document. Scalability and efficiency of approach is determined using simulation of document metadata keywords, location pointers, node connections and node knowledge. The whole process of identifying target location and search consists of nine procedural steps.

- Select target document from network having at least one keyword.
- Use keywords contained in document for the construction of search query.
- Start with free node (not already containing any target document).
- Make a record of start node.
- Start node's knowledge treated as base knowledge for the selection of other sub nodes.
- If number of nodes is equal to forwarding degree select those nodes.
- If number of nodes is less than forwarding degree select additional nodes.
- If number of nodes is more than equal to forwarding degree selects subset of nodes.
- For each selected node, if node contained target document update connectivity and if doesn't then continue search using node.

Authors have explicitly mentioned that this search mechanism is good but still there is a room for improvement in examining the path length of searches for different and same users



characterized by their different query distributions. Moreover time to converge to a stable network can be an ambiguity and need to have more realistic simulation using parameters and distributions.

Metadata Search Layer is designed to identify target location by searching using simulation of document metadata keywords, location pointers, node connections and node knowledge, moreover the nine stepped Metadata Search Layer methodology looks quite impressive to see and adopt implementing search mechanism in my research and development work.

#### **4.5.7 Cultural Heritage and the Semantic Web [38]**

Cultural Heritage and the Semantic Web is ontology based proposed approach and implemented in a tool to semantically annotate existing cultural contents by supporting the annotation process by an intelligent editor. Proposed approach is limited to some extent but still provides the process of semantic navigation, intelligent search, 3D visualization and methodology to publish and exploit content on the Semantic Web.

Cultural Heritage and Semantic Web aims of providing semantic navigation, intelligent search and 3D visualization which are some of the key interests of my research and development work as well.

#### **4.5.8 Semantic Security Web Services (SSWS) [30]**

SSWS is an ontology based solution towards end-to-end service authentication, authorization, data integrity and confidentiality problems introduced by web services. SSWS is capable of improving currently available concept of web services by adding semantically meaningful, declarative and machine process able descriptions of security. SSWS is developed using semantic web technologies OWL, XML, context of DARPA Agent Mark-up Language (DAML) and some existing web's supporting technologies SOAP, WSTK, Apache Tomcat Web, HTML, JSP, and Servlets.

SSWC aims of providing end-to-end service authentication, authorization, data integrity and confidentiality which can be are which can be important aspects to be considered in future in my research and development work.

#### **4.5.9 Partial Distance Map (PDM) [62]**

Authors have proposed an interactive high dimensional indexing scheme, based on Partial Distance Map (PDM), to speed up the process of information retrieval especially from Chinese calligraphic character database. According to authors, due to the high level complexity of Chinese characters no as such efficient technique exists to retrieve and index large Chinese calligraphic character databases.

Authors have used Approximate Point Context (APC) to search characters, proposed Partial Distance Map to established link between semantic-level concepts of characters and low-level shape features through the user relevance feedback and during information retrieval process

Pruning Distance Table (PDT)<sup>6</sup> is used and dynamically updated. Authors developed a pseudo search algorithm based on Hyper Center Reallocation (HCR), an Approximate Minimal Bounding Hypersphere and Clustering based method, used to find out First Nearest Neighbor (1-NN). The pseudo algorithm works in two steps, first it calculates 1-NN using HCR based on submitted query and then start search with a small radius and increase the size of radius step by step to form a bigger query sphere iteratively. Once the number of candidate characters is become larger than  $k$ , the  $(|S|-k)$  characters which are farthest to the query one are identified and removed from  $S$ . In this way, just the  $k$  nearest neighbor characters is returned.

Propose techniques is also evaluated by authors in an extensive performance study, which was resulted with the information that PDM is superior to iDistance and NB-tree in terms of both I/O and CPU cost.

Partial Distance Map aims to speed up the process of information retrieval and searching characters that is one of the key interests of my research and development work as well.

---

<sup>6</sup> A sequence of pairs which contains the corresponding PDs of the different characters

## 5 I-SOAS Approach

This chapter presents a solution to provide comprehensive support in implementing a semantic and agent based solution towards the problems of intelligent graphical user interface, Meta data extraction, modeling and information retrieval over the web for PDM systems.

Keeping eye on above already discussed four major and very important aspects to be considered before and during the development of most of the software products .i.e., Human Machine Interface, Meta Data Extraction, Information Processing Management and Data Presentation I have proposed a new approach *Intelligent Semantic Oriented Agent Based Search (I-SOAS)*. The proposed architecture of I-SOAS is consists of four main sequential iterative components .i.e., Intelligent User Interface, Information Processing, Data Management and Data Representation as shown in Figure.

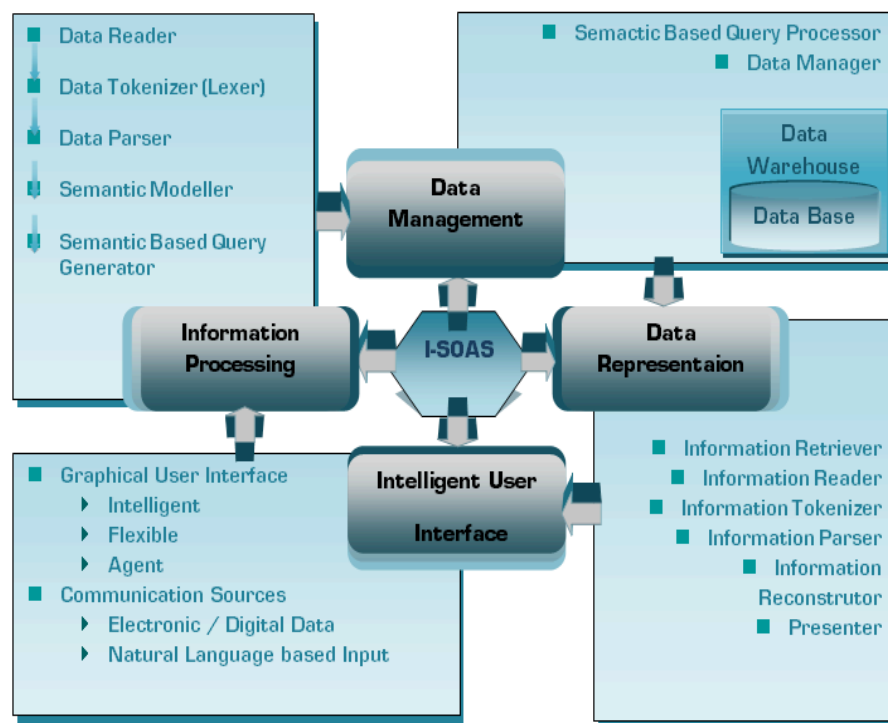


Figure 8: Intelligent - Semantic Oriented Agent based Search (I-SOAS)

Every product whether is software or hardware has a user interface for user system communication, here in our proposed architecture of I-SOAS the first component is also the User Interface .i.e., Intelligent User Interface. I have tried to improve concept of User Interface to Intelligent User Interface. Considering the concepts of compiler construction of any software programming language I have tried to introduce a new concept of information processing and semantic modeling. Then following the already available well matured data warehousing concepts to store and manage data I have designed our third component for data

management. Then to reconstruct the final output of the system in user's understandable format, I have designed the last component of I-SOAS.

## 5.1 Intelligent User Interface

Intelligent User Interface is responsible for the intelligent communication between user and the system. Intelligent User Interface is proposed as an intelligent dynamic user interface which is be capable of first analyzing the source of input, forwarding inputted data for further processing and responding back to user with end results in the same input format. Moreover Intelligent User Interface is supposed to be flexible enough so then it can be redesigned by user itself.

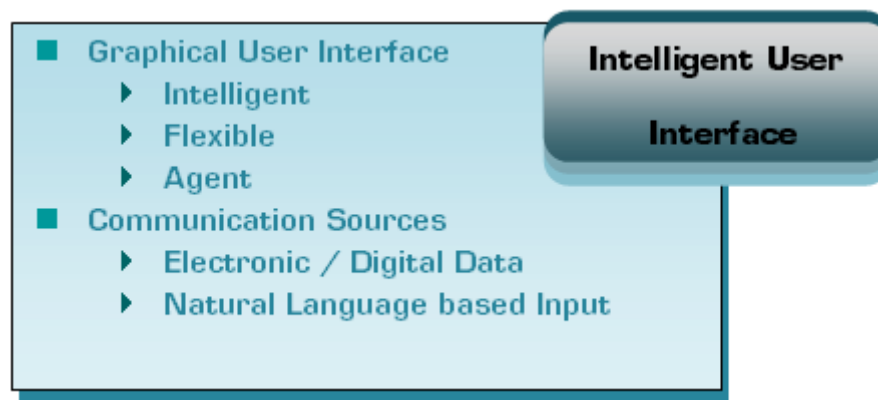


Figure 9: Intelligent User Interface

To implement the Intelligent User Interface system architecture, Intelligent User Interface is divided into two main categories .i.e., Graphical User Interface and Communication Sources. In Communication Sources first the corresponding user is identified to enable the correct communication mode. If it is a digital system then electronic data communication mode is enabled and if it is natural system then natural language based communication mode is enabled. Whereas the Graphical User Interface is consists of the concept of three sub categories .i.e., Intelligent, Flexible and Agent to intelligently handle the user's requests, provide multiple options to redesign the graphical user interface according to the ease of the user by user itself and perform internal architectural component's agent based communication.

## 5.2 Information Processing

Information Processing unit is the most important component, the quality of its performance depends upon the accuracy in the results of this component. The overall job of Information Processing is divided into five main iterative sequential steps .i.e., data reading, tokenization, parsing, semantic modeling and semantic based query generation as shown in Figure.12. Each step requires intensive and intelligent effort in development. The main concept behind the organization of these five steps is to first understand the semantic hidden in the context of natural or digital set of instructions and generate a semantic information process able model for the system's own understanding and information processing.

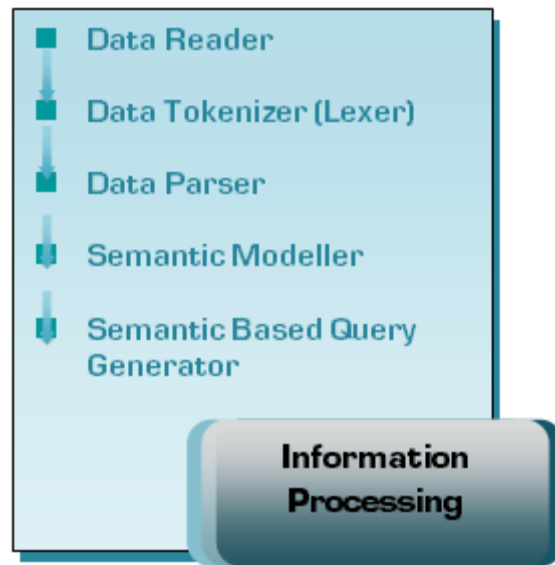


Figure 10: Information Processing

In the first step Data Reader, retrieved data from Intelligent User Interface is just read and organized without performing any analytical intelligent operation except initial prioritization of instructions in to a list. Then in the second phase Data Tokenizer (Lexer), instructions are tokenized in to the possible number of tokens which are then treated in the third step Data Parser for parsing and semantic evaluation with respect to the grammar of used natural or digital language. If someone has the idea of compiler construction then he can better understand the process of tokenization and parsing because this works on almost the same principles that are normally followed by any other compiler of any programming language.

Then in the third and the most important phase of Information Processing Semantic Modeler, the tokenized and semantically evaluated information is used and reorganized in Meta data based semantic model. The main purpose of the creation of Semantic Modeler is first to filter the irrelevant semantic less helping grammar based information from the actual information based context and then create a semantic model which our own system can easily analyze and process. Then in the last phase Semantic Based Query Generator based on the output of Semantic Modeler generates a new query used further for the extraction of desired result in next component Data Management.

### 5.3 Data Management

As the concept of Data Management is concerned, data management is based on two main principles. i.e., storing data and managing data. In I-SOAS Data Management is playing the heaviest component's role due to the size, because the size of this component will increase with the increase of data with the passage of time. This component is responsible for two main functions. i.e.; Semantic based Query Processing and Data Management.

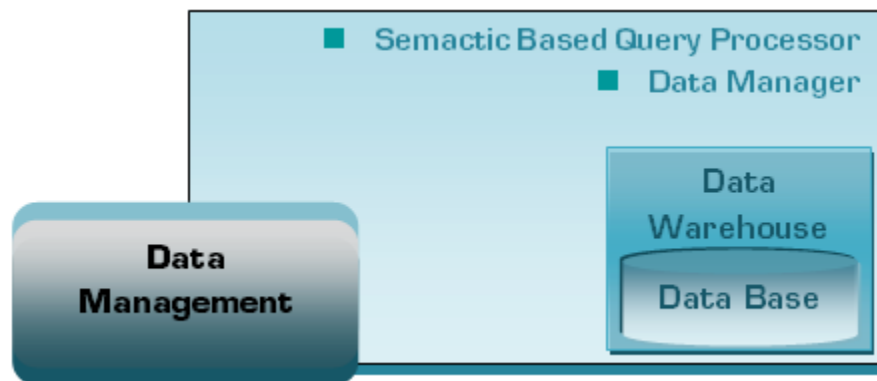


Figure 11: Data Management

Semantic based query build in IP is treated by Semantic based Query Processor to generate SQL query to run in to data warehouse to extract the required relevant information. The job of Data Manager is to manage the process of SQL query building, data extraction and creation of new indexes and storage based on newly retrieved information.

## 5.4 Data Representation

Data Representation is carrying one of the important jobs of responding back to the user with finalized results. This component consists of six sub components. i.e., Information Retriever, Information Reader, Information Tokenizer, Information Parser, Information Reconstructor and Presenter. The job of this component somehow is relevant to the job of Information Processing, but major difference is of handling data and information. IP treats data to process but Data Representation treats information.

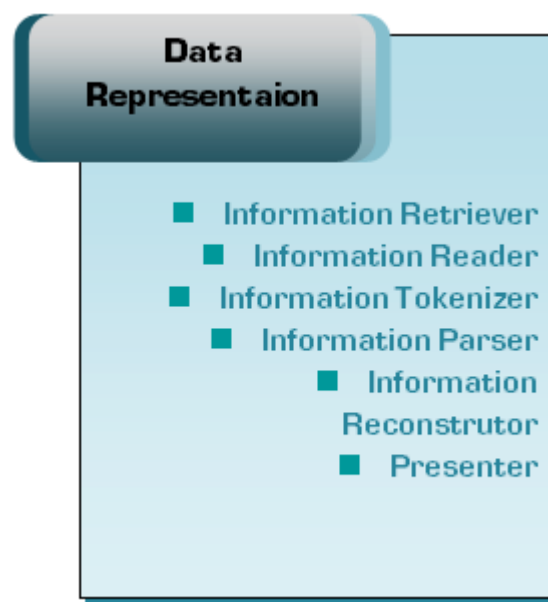


Figure 12: Data Representation

Required extracted and managed information from Data Manager is passed to Data Representation using Information Retriever, which simply read and organized by Information Reader without performing any analytical intelligent action except prioritization of informative statements. Then using Information Tokenizer and Information Parser statements are tokenized and parsed and using Information Reconstructor. Finalized formatted information is build in user's used natural language. Finally Presenter presents the resultant information to Intelligent User Interface to respond back to the user.

## 6 Implementation

This chapter provides the detailed information about implementation design of I-SOAS including the information about the theme, design requirements, system sequence and internal workflow.

### 6.1 Implementation Design

To implement I-SOAS in the form of a real time software application, a new 4-tier implementation architecture is designed.

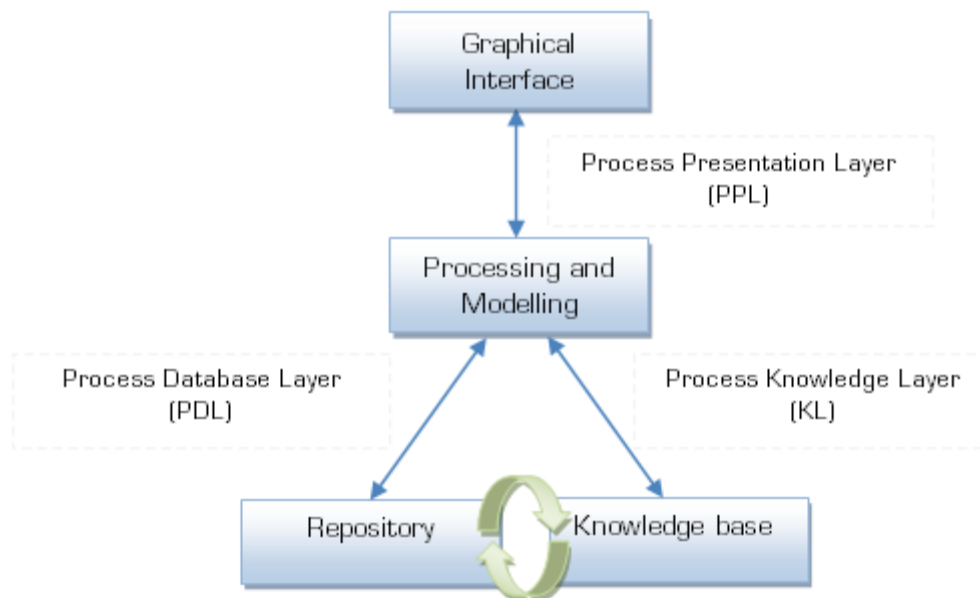


Figure 13 Implementation Architecture

The designed architecture as shown in figure 16 is consists of four modules Graphical Interface, Repository, Knowledge Base and Processing and Modeling, and three communication layers Process Presentation Layer (PPL), Process Database Layer (PDL) and Process Knowledge Layer (PKL). These are four modules and three layers are independent components but will work in integrated form to perform the user required functions.

#### 6.1.1 System Sequence Architecture

The I-SOAS System Sequence Architecture is consists of four main components .i.e., Interface, Processing & Modelling, Data Base and Knowledge Base doing the same already mentioned jobs but by communication with each other following a certain sequence.



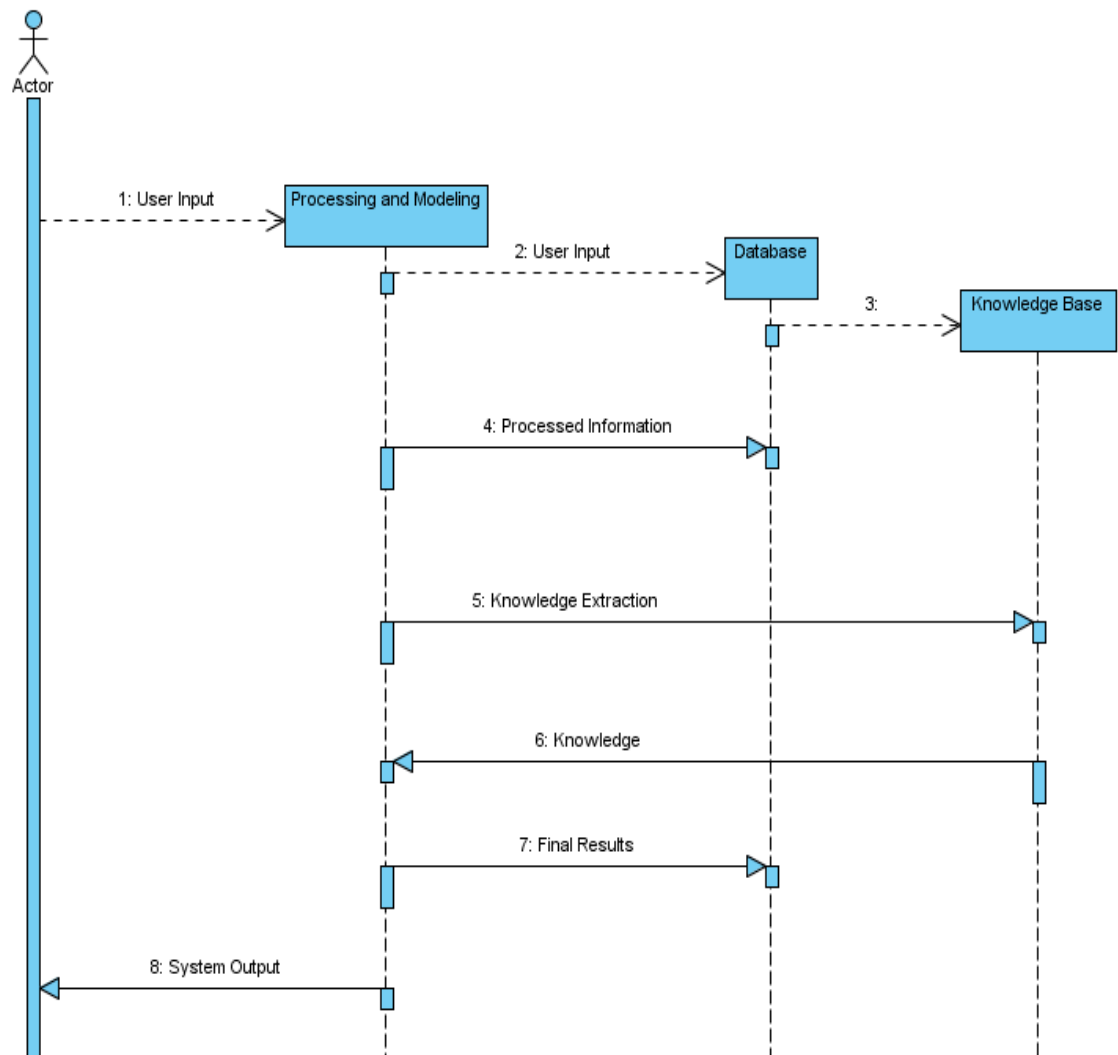


Figure 14: System Sequence Architecture

Interface will take the input from user and forward that input to the Processing and Modelling Component. Processing and Modelling Components first will save the user input in Database Component then process the input, extract Knowledge from Knowledge Component and then return final results to the Interface. Interface will give that system output to the user.

### 6.1.2 Technological Involvement

The overall architecture depending upon the selection and usage of tools and technologies for the development of I-SOAS is presented below in Figure.

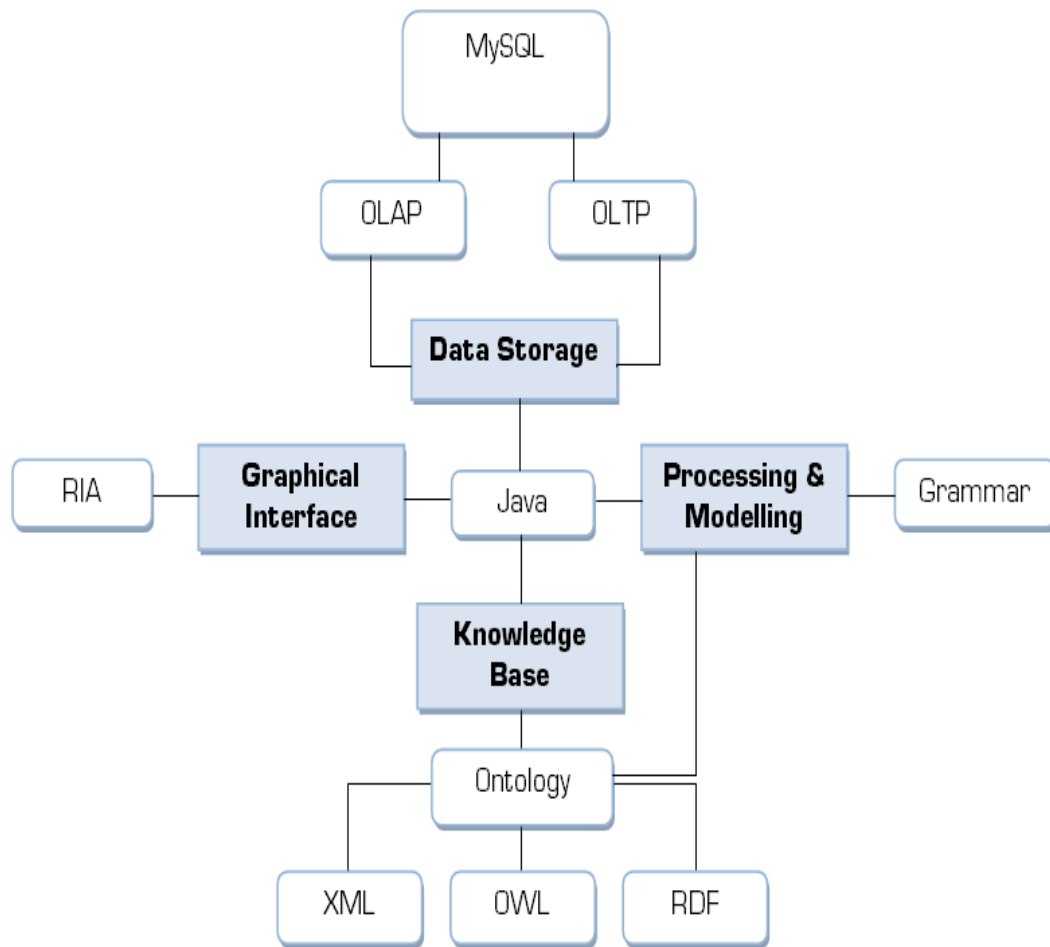


Figure 15 Involved Technologies

The overall development of each component will be performed in Java language, but moreover for the development of Graphical Interface I am considering Rich Internet Applications (RIA), for the development of Data Storage I am considering Oracle Platform, for the development of Information Processing and Knowledge Based I am considering Ontology based technologies XML [17], RDF [18] and OWL [19].

## 6.2 Graphical Interface

Designing and implementing an intelligent and user friendly human machine interface for any kind of software or hardware oriented application is always be a challenging task for the designers and developers because it is very difficult to understand the psychology of the user, nature of the work and best suit of the environment. This section of dissertation is basically about to propose the draft design of an intelligent, flexible and user friendly machine interface for Product Life Cycle Management products or PDM Systems since studies show that usability and human computer interaction issues are a major cause of acceptance problems

introducing or using such systems. Going into details of the proposition, I present prototype implementations about theme based on design requirements, designed designs and technologies involved for the development of human machine interface.

Graphical Interface module is actually the graphical user interface of I-SOAS for the user system communication. This component is responsible for two jobs

1. Taking input from user and forwarding that input to Processing and Modelling Module.
2. Getting output from Processing and Modelling Module and replying back to the user.

The proposed graphical design is based on Human Machine Interface (HMI) Design concepts .i.e., Ideologies, Principles, Patterns and Guidelines, already discussed in Chapter 3 of this dissertation.

The main theme or idea behind the design of I-SOAS Graphical Interface is to develop an intelligent graphical user interface for Product Data Management based Application having several different (may be hundreds or more) options performing several different actions for several different kinds of operational and managerial level users. Moreover I-SOAS Graphical Interface is proposed to be as much intelligent and flexible so then the user can easily be incorporate itself in using standard interface and can also redesign interface according to his own choice using provided privileged options.

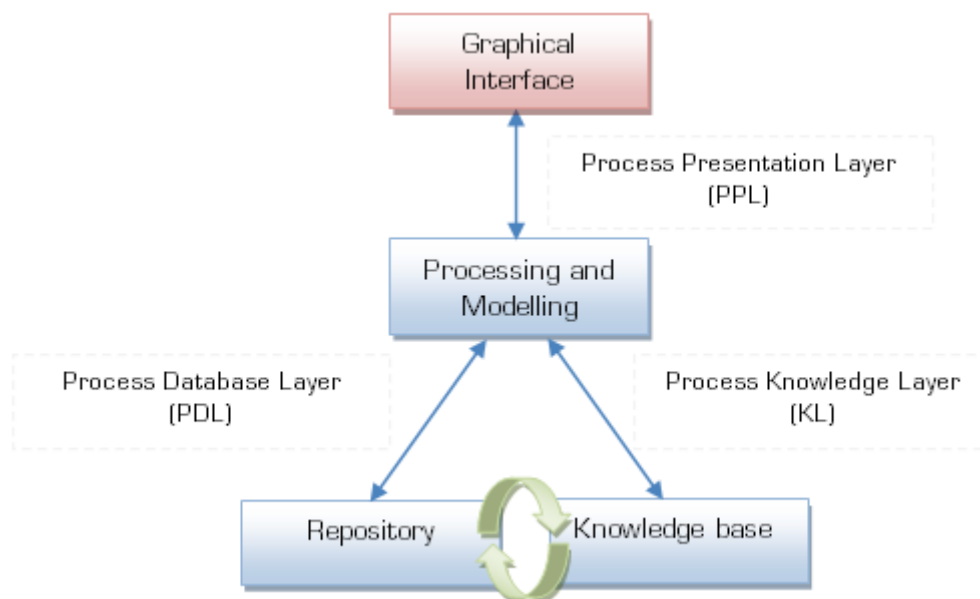


Figure 16: Implementation Architecture - Graphical Interface

Moreover the I-SOAS Graphical Interface is conceptually based on the concepts of the first component of I-SOAS's conceptual iterative architecture Intelligent User Interface (IUI) as shown in figure 11, 12 and 16.

### 6.2.1 Design Requirements

The I-SOAS Intelligent User Interface is designed to meet four major requirements. i.e., flexibility, agent based design, context awareness and natural language communication mode.

#### *Flexibility*

The designed graphical user interface must be flexible in order to enable users to redesign and reconfigure the interface itself to accommodate specific needs. Examples:

1. When software loads first time, then it should identify the level of user, whether it's a beginner or experienced. If the user is beginner than an automatic GUI trainer should be provided, which will explain the provided available options in the application, their usage and GUI design operations in best possible short time, and If the user is an experienced person, then will move user to the application directly.
2. User will be provided with two major options .i.e., Manual Option Adjustment and Standard GUI. Manual Option Adjustment will allow user to create a new GUI by providing several operations .i.e., Mouse Click and Drag Drop, Key-board Key Press, Screen Touch and Voice Recognition etc.
3. User will be provided with the option to adjust the available already defined GUI, save redesigned GUI as template, load or reload already designed template, delete running template and lock or unlock GUI.
4. User will be provided some options in the form of Standard GUI Controls like Buttons, Labels, List Box, Combo Box, Text Box etc., Color Schemes, Font, Graphics, Animations and Standard GUIs like forms to take data inputs and view the results etc.
5. User will be provided a desktop and web based platform independent graphical user interface.

#### *Agent based design*

The proposed graphical interface must contain a Personal Agent (PA), which should be capable to the following tasks .i.e.,

1. PA should be able to move in the graphical interface, and this movement should be of two types .i.e., Explicit Movement (PA can be moved by the user) and Implicit Movement (PA can move itself according to the environmental adjustments).

2. PA can be controlled and instructed by user using mouse click, drag drop, keyboard key press, screen touch and voice recognition options.
3. PA should be able to produce emotions according to the nature of job and current situation like Typing on keyboard when user is typing or asking PA to type something for him, Listening when user is speaking and asking PA for some in-formation or job, wear glasses when user is asking PA to look for something etc.
4. PA should be able to save his state and maintain his history.

### ***Context Awareness***

The proposed graphical interface must perform the following jobs .i.e.,

1. Learn from experience
2. Help user in decision making
3. Provide Intelligent GUI trainer to intelligently train the new user
4. Should be able to handle GUI itself
5. GUI must be able to present output of inputted instruction in best possible way with respect to the user, like if user is asking in voice output should be delivered in voice etc.

### ***Digital Communication Mode***

The graphical interface must contain a fixed digital communication mode, where user needs to follow only provided instructions to proceed and perform some actions like by pressing button or choosing some provided options etc.

### ***Natural Communication Mode***

The graphical interface must contain a natural communication mode, where user can instruct the software with natural language based instructions like user can write instruction in natural speaking language and software can read, understand the context and semantic, and then performed required action.

## **6.2.2 Draft - Graphical Interface Design**

The proposed graphical design is designed by following already presented designed requirements in section 5.1 and it is based on Human Machine Interface (HMI) Design concepts (already described in Chapter 3) .i.e., Ideologies, Principles, Patterns and Guidelines. This manual physical sketch is basically based on several different visible and non visible options but mainly there are two different categories of options .i.e., Control Interface and

Personal Assistant. The designed manual physical sketch of I-SOAS Graphical Inter-face is mainly consists of a Control Container. Control Container is the main front page of the graphical user interface which contains all the controls options including list boxes, mouse hover/click, drag drop, drop down list boxes and list boxes. These options are provided to the user to perform certain tasks. Moreover user can redesign or make changes in already designed graphical interface using these options and controls. As shown manual physical sketch is seemed to be a very simple and non-attractive graphical interface proposition and design because it is not the final design, the user of the application is supposed to design the required graphical user interface itself.

The proposed physical sketch of Personal Assistant (PA) is designed keeping Agent based design requirements in mind. PA is proposed to be an agent based software application having autonomous behavior for working in a particular domain by communicating, moving and collaborating with users and other software applications in an automated environment to obtain desired results. The main job of the PA in I-SOAS Graphical Interface is to communicate with user; take user's struted and instructed requests, read them and forward them for father processing back to the user with obtained results.

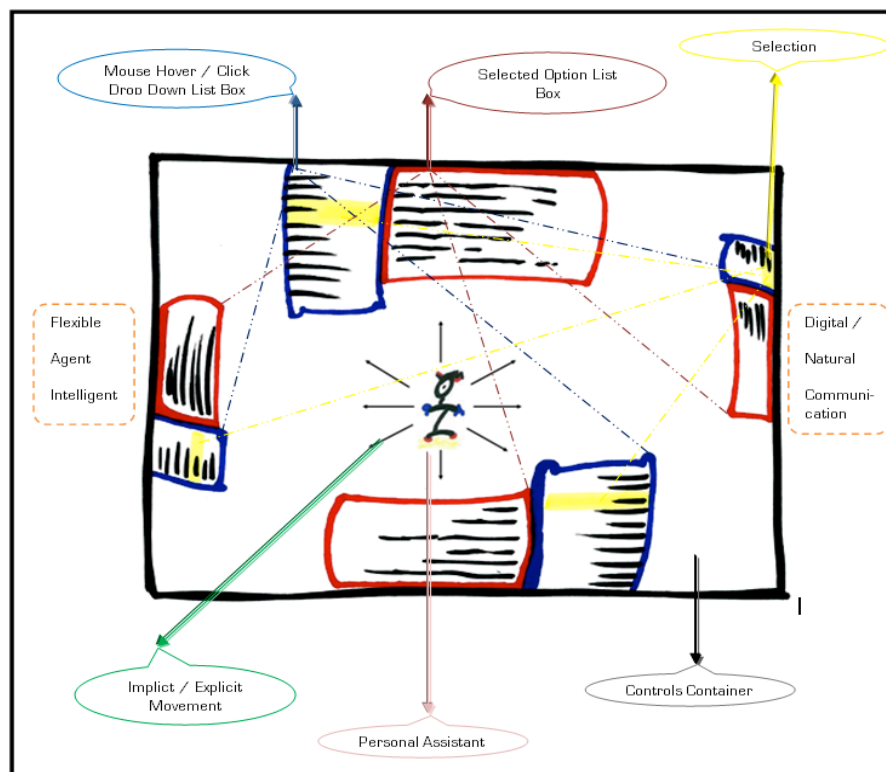


Figure 17: Manual Physical Sketch Description

### 6.2.3 Internal Work Flow

Figure 18 shows the flow of the software. The interaction starts with the user input. At first the user inputted request is analyzed and then it's classified into two major categories .i.e., Build GUI and Process Input.

If the user requested input is about to change the user out-look of graphical interface then the control will go to the Build GUI category and if the user requested input is about to extract some information by processing and modeling data then the control will go to the process input step.

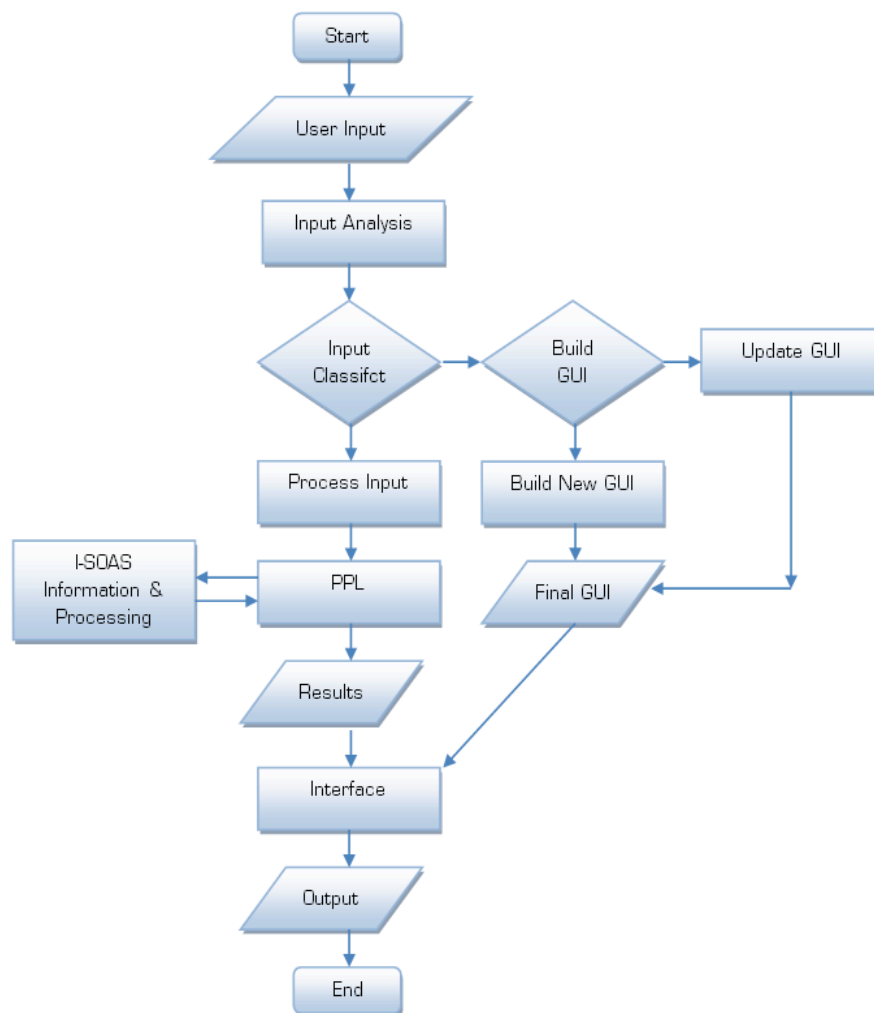


Figure 18: Graphical Interface Internal Flow

During Build GUI operations. At first the requested user input is again analyzed and categorized into two further categories .i.e., Update GUI and Build New GUI. If the user's request is about to build a completely new graphical outlook then it will move to the step Build New GUI but if the user's request is about to make some minor or major changes in

already running outlook then it will move to the Update GUI. Final interface will then become the final output from I-SOAS Graphical Interface.

During Process Input operations user's requested input forwarded to the Processing and Modelling Component and final resultant output is obtained via Presentation Process Layer. This final output will then become the final resultant information for the user.

### 6.2.4 System Sequence Design

Figure 19 shows the Sequence design of the Graphical Interface System. There are four main components. i.e., Interface, Input Classification, Build or Modify GUI and, Process Presentation Layer. The job of interface is to take input request from user and forward it to the input classification component that will verify, validate and classify the inputted user request. If user requested to make or update existing graphical interface then the input classification component will forward the user re-quest to Build of Modify GUI component which will modify or build GUI and will update the final changes on interface.

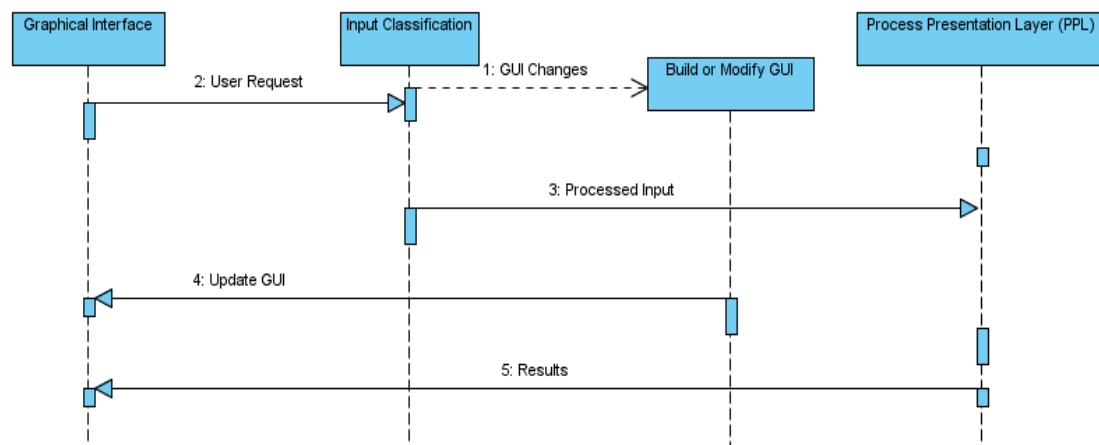


Figure 19: Graphical Interface System Sequence Design

Whereas if user requested to process and extract some information then input classification component will forward the request to Process Presentation Layer which will then forward the request to Information Processing and Modeling component for further processing and take back the resultant information to present on the GUI.

## 6.3 Processing and Modeling

Designing an intelligent application capable of reading and analyzing user's structured and unstructured text based requests and then extracting desired concrete and optimized results from knowledge base is still a challenging task for the designers because still it is very difficult to completely extract Meta data out of data. This section of dissertation is mainly about to propose a new intelligent semantic based information processor and discuss its design



in detail including the information about the theme, design requirements, designed designs and technologies involved in the development.

I-SOAS Processing and Modeling (PM) module is a semantic based system proposed and designed to read, organized, tokenize, parse, semantically evaluate, model and process information to extract desired results from I-SOAS Data Repository and I-SOAS Knowledge base.

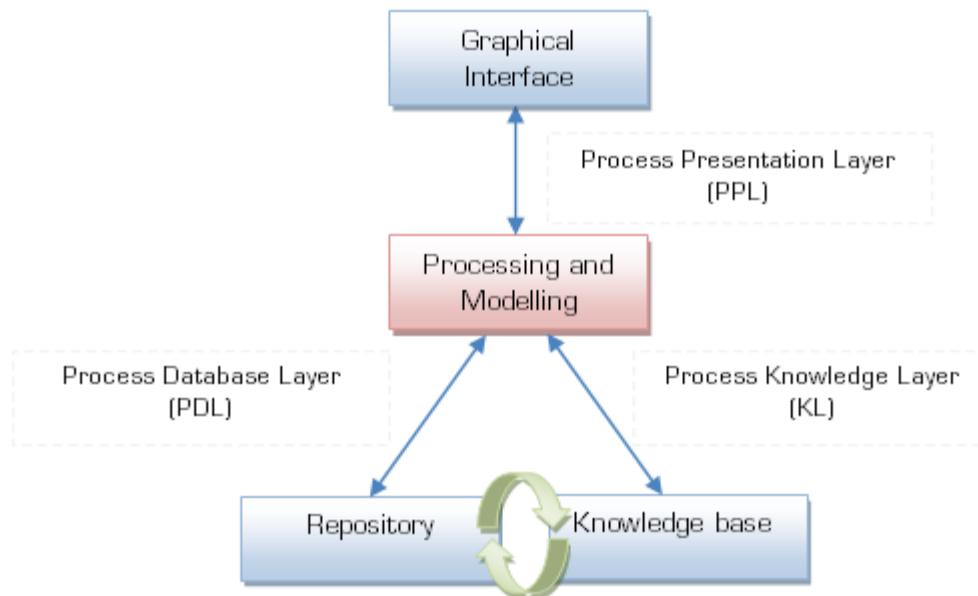


Figure 20: Implementation Architecture – Processing and Modeling

The main theme or idea behind I-SOAS Processing and Modelling is to develop a system which can process the input of the user, store and manipulate data in repository, extract knowledge out of data and return obtained results back to the user. Moreover I-SOAS Graphical Interface is conceptually based on the concepts of the second component of I-SOAS's conceptual iterative architecture Information Processing (IP) as shown in figures.

### 6.3.1 Design Requirements

The I-SOAS Processing and Modelling is designed keeping five major requirements. i.e., Input Analyze, Information Processing, Data Store and Management, Knowledge Extraction and Building Final Output in mind.

#### *Input Analysis*

The designed I-SOAS Processing and Modelling component must be capable of handling user inputted request by analyzing the inputted instructions.

#### *Information Processing*

The designed I-SOAS Processing and Modelling component must be capable of understanding the context and semantic of user's requested input by processing and modeling instructions.

#### ***Data Storage and Management***

The designed I-SOAS Processing and Modelling component must be capable of storing and managing the user's requested input and final output in database.

#### ***Knowledge Extraction***

The designed I-SOAS Processing and Modelling component must be capable of extracting user's requested knowledge from knowledge base.

#### ***Building Final Output***

The designed I-SOAS Processing and Modelling component must be capable of building final system output in user understandable format.

### **6.3.2 Internal Work Flow**

I-SOAS Processing and Modelling is supposed to start with the input text which at the first step will be stored into database then will be forwarded to I-SOAS Lexer.

I-SOAS Lexer is supposed to first tokenize the whole input text into possible number of token, and then evaluated the context with respect to the used grammar of the input language. Then these token will be forwarded to I-SOAS Parser.

I-SOAS Parser is supposed to consider all the statements or instructions as the combination of all tokens produced by I-SOAS Lexer for semantic evaluation. Resultant semantically evaluated information from I-SOAS Parser will be forwarded to I-SOAS Modeler.

I-SOAS Modeler is supposed to produce semantic based models using syntactically and semantically evaluated information. Then these models are supposed to be stored in database and forwarded to I-SOAS Resolver.

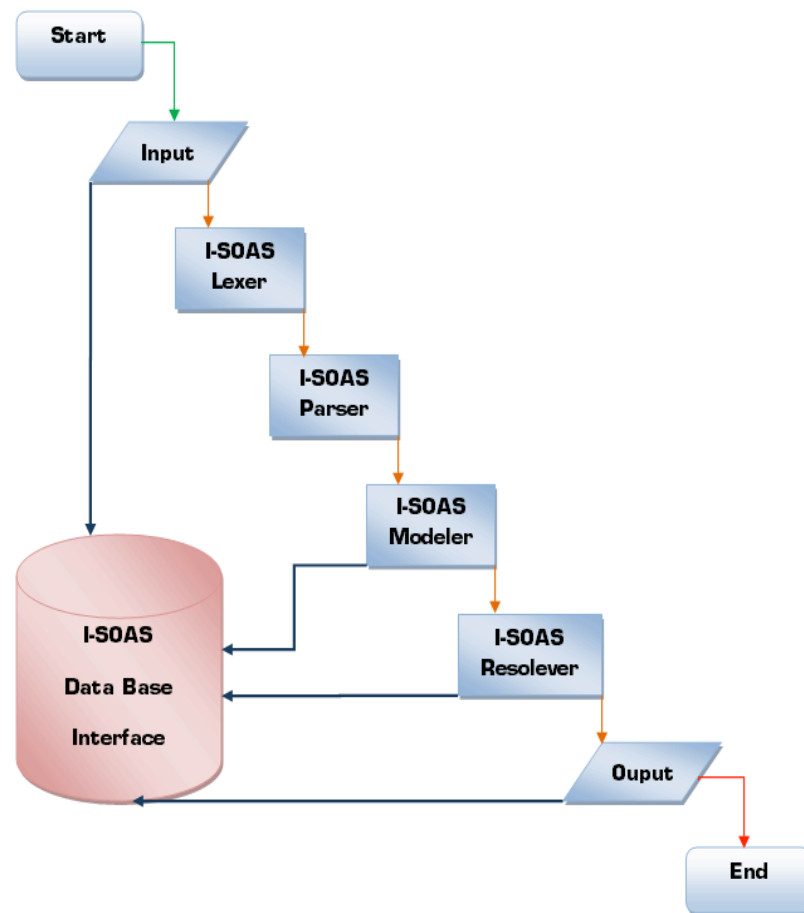


Figure 21: Process & Modelling Internal Work Flow

I-SOAS Resolver is supposed to take semantic models as input and then resolve those models by producing relationships between modeled statements. Then final resolved relation based information is stored in to database and can be treated as the final output of I-SOAS Information Processing.

### 6.3.2.1 Grammatical View

I have written and designed the grammatical view for the I-SOAS Lexer and I-SOAS Parser as shown Figure. This grammar is based on only English (Natural Language).

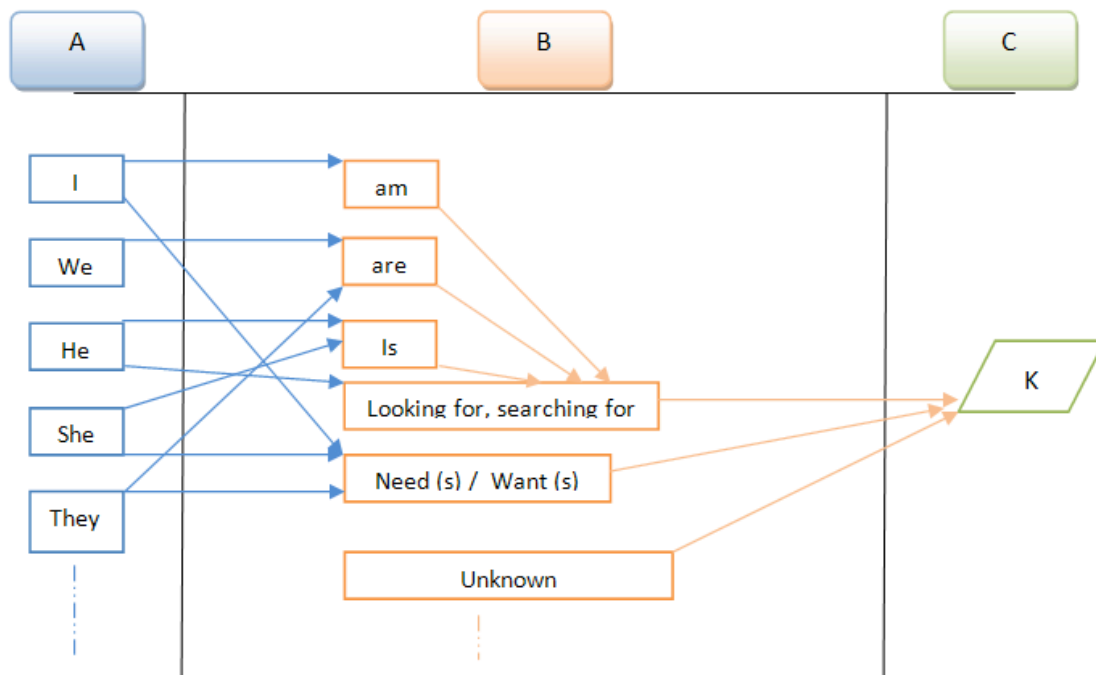


Figure 22: I-SOAS Lexer and Parser Grammar

As shown in the figure, at the moment, according to the scope of our research, the proposed grammar has been divided into three main categories. i.e., A, B and C. A category is representing the all-English words belonging to User (requesting for some results), B category is representing the English grammar structure and C is representing the main Object user is looking for (K).

### 6.3.2.2 Ontological View

To produce ontology based search I have mapped the used grammatical view in Lexer and Parser in to an ontological view as shown in figure. There three main entities. i.e., A, B and C. There are five main properties of A .i.e., *I*, *We*, *He*, *She* and *They*, six properties of B .i.e., *am*, *are*, *is*, *looking/searching for*, *need(s)/want(s)* and *unknown* and C has only one property *K*.

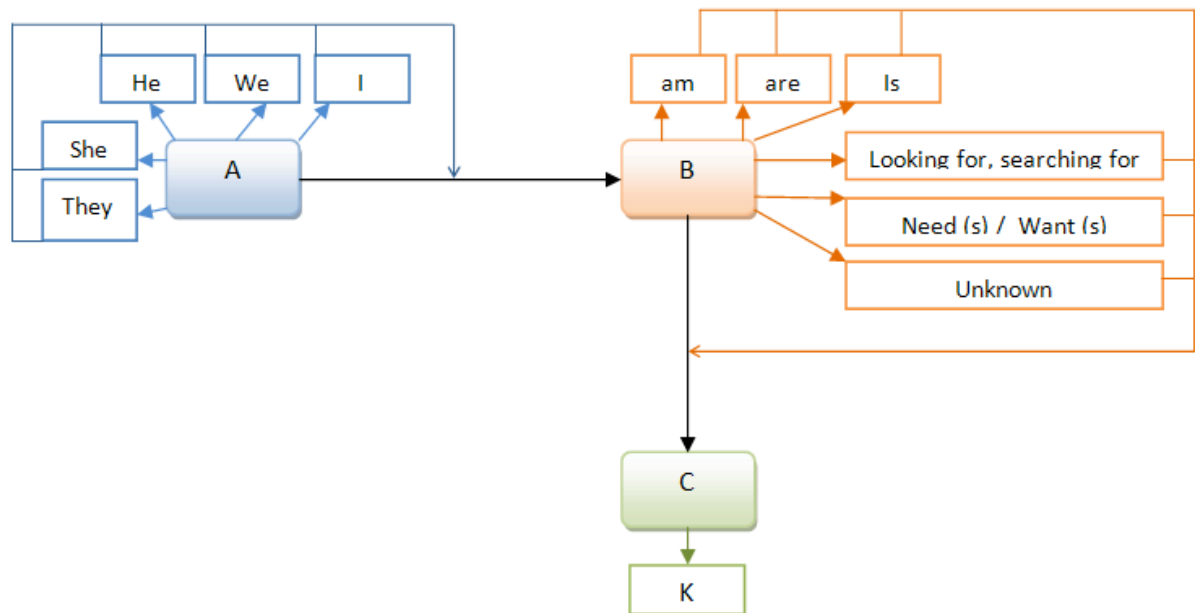


Figure 23: I-SOAS Ontology

### 6.3.3 System Sequence Design

I-SOAS Information and Processing System Sequence Design consists of the five main components. i.e., Process Presentation Layer, User Input, Process and Model Information, Process Database Layer and Process Knowledge Layer.

These five components are supposed to perform certain jobs. The job of Process Presentation Layer is to bring user requested input from I-SOAS Graphical Interface to the User Input component of I-SOAS Processing and Modeling. Then User Input component will forward the user requested inputted instructions to Process and Model Information component.

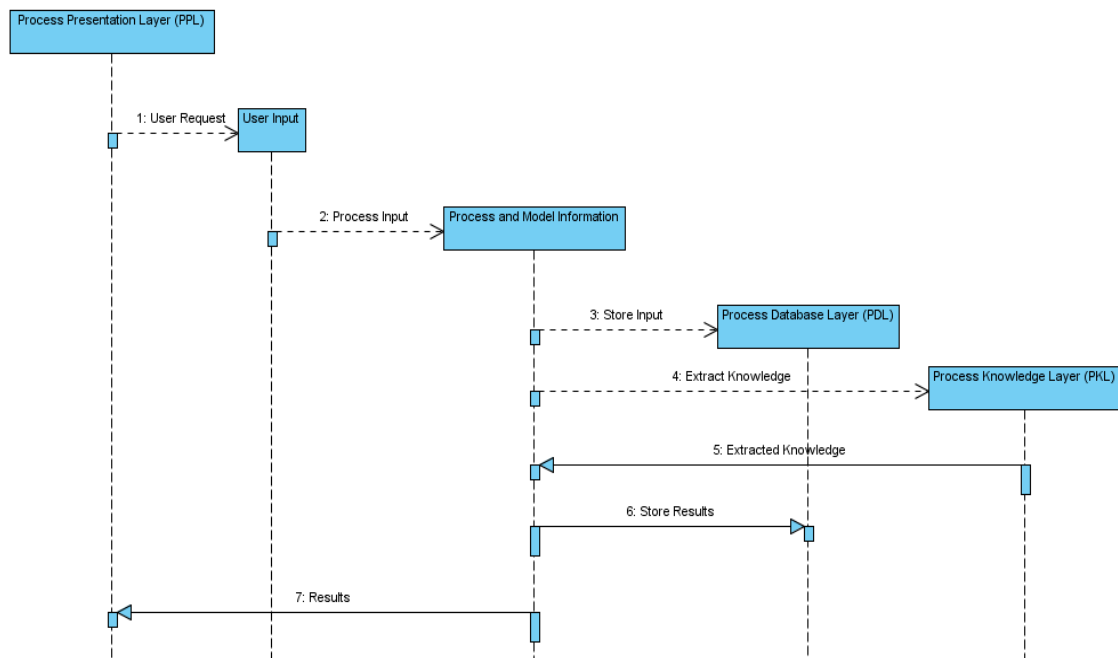


Figure 24: Process and Modelling System Sequence Design

Process and Model Information component first will store the information in to I-SOAS Database via Process Database Layer, then will process the information by lexing, parsing and semantically modeling, and then extract knowledge from I-SOAS Knowledge Base via Process Knowledge Layer. In the end Process and Model Information component will first save the information in I-SOAS Database via Process Database Layer, and then return the final system output to I-SOAS Graphical Interface via Process Presentation Layer.

## 6.4 Repository

Designing an integrated semantic based product line database application capable of managing Organizational Data well as System Data at the same time is always be a challenging task for the designers. This section of dissertation is about to present data base management system design for the Product Data Management based companies including theme, design requirements, designed designs and technologies involved in the development. The main theme or idea behind the I-SOAS Repository is to develop a repository to store and manage Product Data Management based Application's heavy volume data. Moreover The designed Repository is conceptually based and meeting the requirements of third component of I-SOAS's conceptual iterative architecture Data Management.

Repository module is actually the repository to store, extract, transforms, load and manage organizational technical and managerial data. I-SOAS Repository will work like Data Warehouses by providing several options to produce common data model for all data of interest easier to report and analyze information, prior loading data, security and retrieval of data without slowing down operational systems [65].

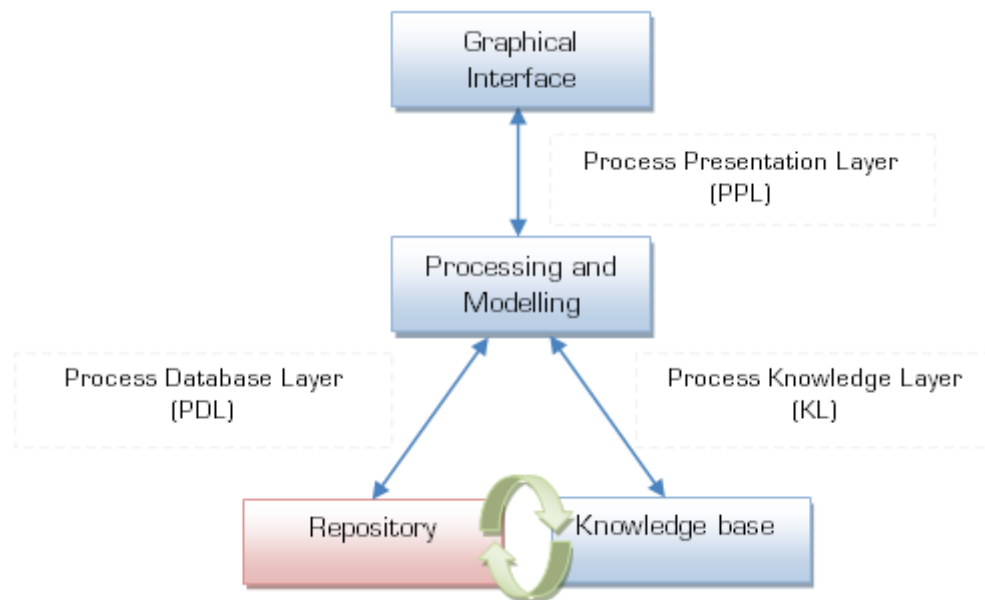


Figure 25: Implementation Architecture – Repository

The technical architecture of Repository consists of five main steps .i.e., Extraction, Transformation, Loading, Security and Job control [66].

1. Data will be extracted from different sources during Extraction
2. Data will be transformed by management, integration, de-normalization, cleansing, conversion, aggregation, and auditing [66] in transformation.
3. Data will be loaded in cyclic process in Loading.
4. Data will be secured in security.
5. Job scheduling, monitoring, logging, exception handling, error handling, and notification [66] will be performed in Job handling.

#### 6.4.1 Design Requirements

The I-SOAS Database must be designed keeping two major record keeping requirements. i.e., *Organizational Data and System Data* in mind.

##### ***Organizational Data***

The designed database must be capable of storing and managing organizational technical and non technical data divide into three sub major categories *User Data*, *Project Data* and *Product Data*.

#### ***User Data***

The designed database must be capable of storing and managing user data containing user's personal information, role(s), responsibilities, type (s), group(s) and right(s) etc.

#### ***Project Data***

The designed database must be capable of storing and managing Project data containing detailed information about project, start end dates, actions, ownerships, category, state, tasks, meeting, status and deadlines etc.

#### ***Product Data***

The designed database must be capable of storing and managing Product data containing detailed information about the product produced in the organization, release dates and important notes etc.

#### ***System Data***

The designed database must be capable of storing and managing System Data containing the information about user inputs, system output, actions, reactions, input processing, modeling and event log etc.

### **6.4.2 Relational Design**

Following the designed requirements two entity relationship diagram as designed for I-SOAS Database. i.e., I-SOAS Database Organizational Data and I-SOAS Database System Data.

#### ***I-SOAS Database Organizational Data Design***

The designed entity relationship diagram of I-SOAS Database Organizational Data is consists of 5 main relations. i.e., Organisation, Person, Staff, Project, Product and 17 supportive relations .i.e., *Name*, *Contacts*, *Contacts\_Web*, *Contact\_Telephone*, *Contact\_City*, *City\_Country*, *City\_State*, *Start\_End\_Date*, *Project\_Team*, *Meeting*, *Activity*, *Staff\_Meeting*, *Document*, *Type*, *Staff\_Document*, *Organisation\_Document*, *Project\_Document* designed and connected to store and manage organizational data, employee's (user) personal data, project data and product data.

*Organisation* relation is designed to store and main the information about respective Organisation.



*Person* relation is designed to store and main the information about every associated person with organization.

*Staff* relation is designed to store and maintain the information of each and every employee of the organization.

*Project* relation is designed to store and maintain the information of each and every running is the organization.

*Product* relation is designed to store and maintain the information of each and every product developed in the organization.

*Document* relation is designed to store and maintain the information of each organizational personal, staff and project document.

*Staff\_Document* relation is an extended relation of relation Document and designed to store and maintain the joint information about staff and document.

*Organisation\_Document* relation is an extended relation of relation Document and designed to store and maintain the joint information about organization and document.

*Project\_Document* relation is an extended relation of relation Document and designed to store and maintain the joint information about project and document.

*Name* relation is designed to store and maintain the information of the name of Organization, Person, Project and Product

*Contacts* relation is designed to store and maintain the information of the contacts of Person and Organisation.

*Contacts\_Web* relation is an extended relation of relation Contact and designed to store and maintain the information about the web contacts consisting of email address and url.

*Contact\_Telephone* relation is an extended relation of relation Contact and designed to store and maintain the information about the telephone contacts consisting of Mobile, Fax and Telephone numbers.

*Contact\_City* relation is an extended relation of relation Contact and designed to store and maintain the information about the city contacts consisting of City Name and Code.

*City\_State* relation is an extended relation of relation City and designed to store and maintain the information about the belonging State contacts of the city.

*City\_Country* relation is an extended relation of relation City and designed to store and maintain the information about the belonging Country contacts of the city.

*Project\_Team* relation is designed to store and maintain the information of each team in the organization working on any organizational project.

*Activity* relation is designed to store and maintain the information of each activity performed in the organization.

*Meeting* relation is designed to store and maintain the information of each meeting between organizational staff.

*Staff\_Meeting* relation is an extended relation of relation Meeting and designed to store and maintain the joint information about staff (attended meeting) and meeting itself.

*Type* relation is designed to store and maintain the information of the types of organization, document, project, product, system input and system output.

*Start\_End\_Date* relation is designed to store and maintain the information of the start and end date and timing of organizational projects and meetings.

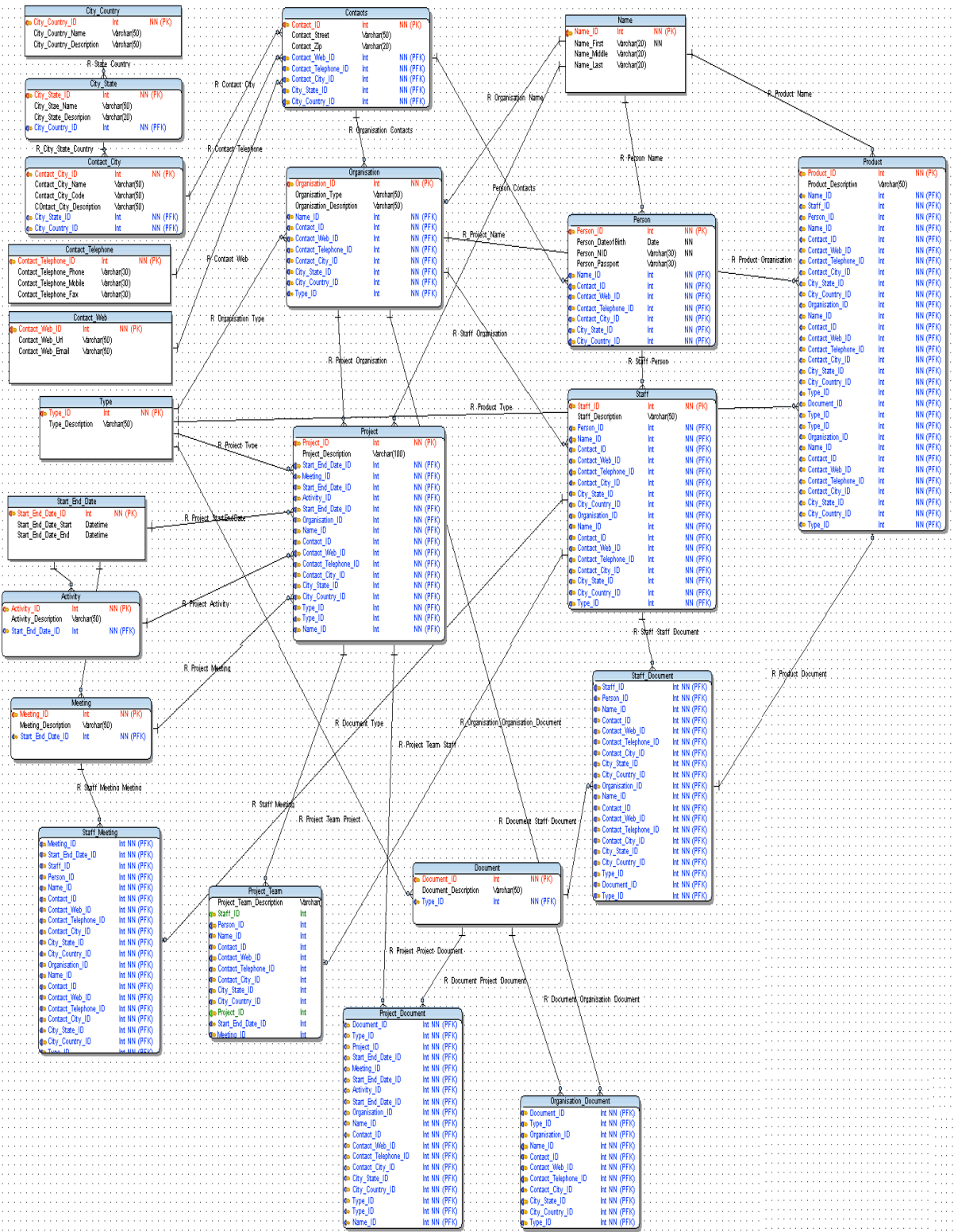


Figure 26: I-SOAS Database Organizational Data (Part 1)

### *I-SOAS Database System Data Design*

The designed entity relationship diagram of I-SOAS Database Organizational Data is consists of three relations. i.e., Login, SystemInput and SystemOutput.

*Login* relations are designed to store and maintain the user system login information (User name and Password).

*SystemInput* relation is designed to maintain user input instruction by storing information about user inputted instruction details including time, date and user (who inputted).

*SystemOutput* relation is designed to maintain system Output by storing information about system outputted instruction details including time, date, and respective input.

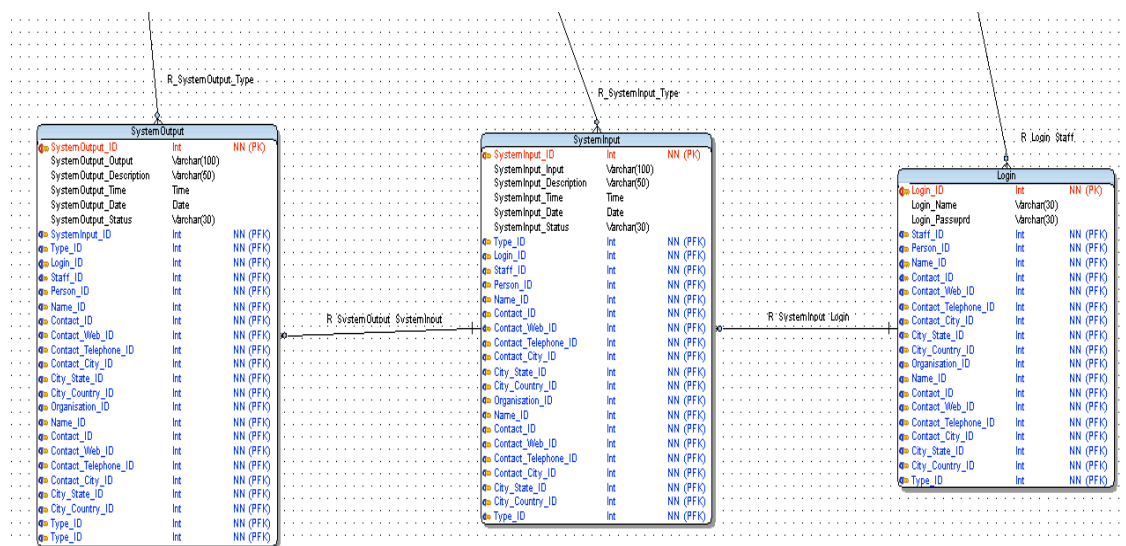


Figure 27: I-SOAS Database System Data

### 6.4.3 I-SOAS Repository: System Sequence Design

I-SOAS Repository System Sequence Design consists of three main components. i.e., Process Database Layer, Data Input and Database.

These three components are supposed to perform certain jobs. The job of Process Database Layer is to bring system data from I-SOAS Processing and Modelling and forward to Data Input. Then system data will be stored and managed in database by Data Input component. Then the final acknowledgement will be send to I-SOAS Processing and Modelling via Process Database Layer.

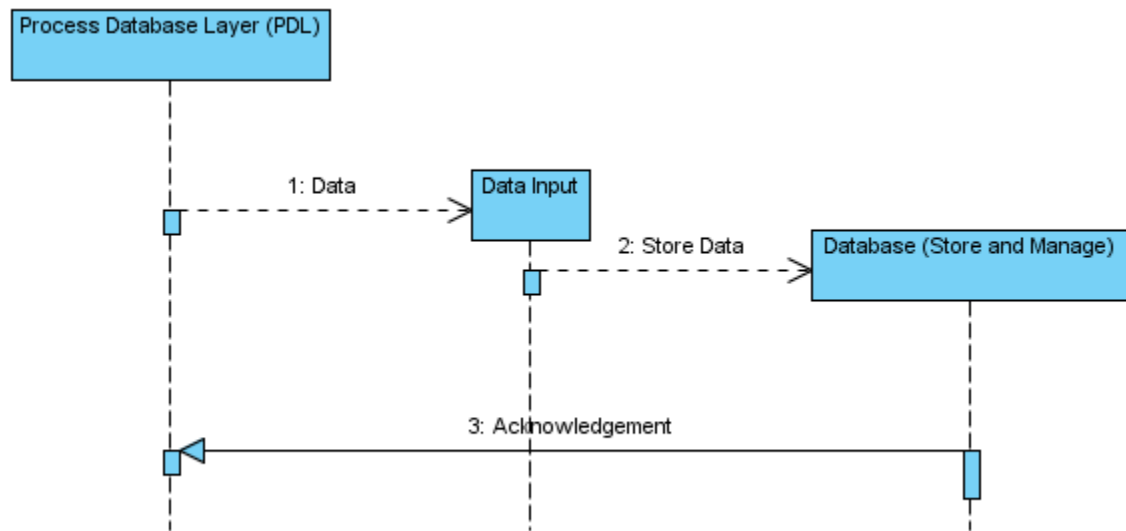


Figure 28: I-SOAS Database System Data

## 6.5 Knowledge Base

Designing an intelligent knowledge base application capable of knowledge extraction, creation, storing, management and sharing is always be a challenging task. This section of dissertation is about to design a knowledge base system for the Product Data Management based products. Moreover this section also present targeted knowledge base system's details including the information about the theme, design requirements, designed designs, life cycle and technologies involved in the development

The main theme or idea behind the Knowledge Base is to develop a Meta data based system to extract and use knowledge out of data. Moreover the designed Knowledge Base is conceptually based and meeting the requirements of third component of I-SOAS's conceptual iterative architecture Data Management.

Knowledge Base is designed to capture, manage, improve and deliver knowledge by providing information collection, retrieval, analysis, management and organization. I-SOAS KB is supposed to contain qualitative information depending upon efficient information retrieval system and content classification structure. Moreover I-SAOS is designed in way so then it can intelligently provide solutions to unsolved problem and behave like an expert system<sup>7</sup>. The structure implementation of I-SAOS can also be based on Ontologies.

<sup>7</sup> System capable of learning from experience and environment [67]

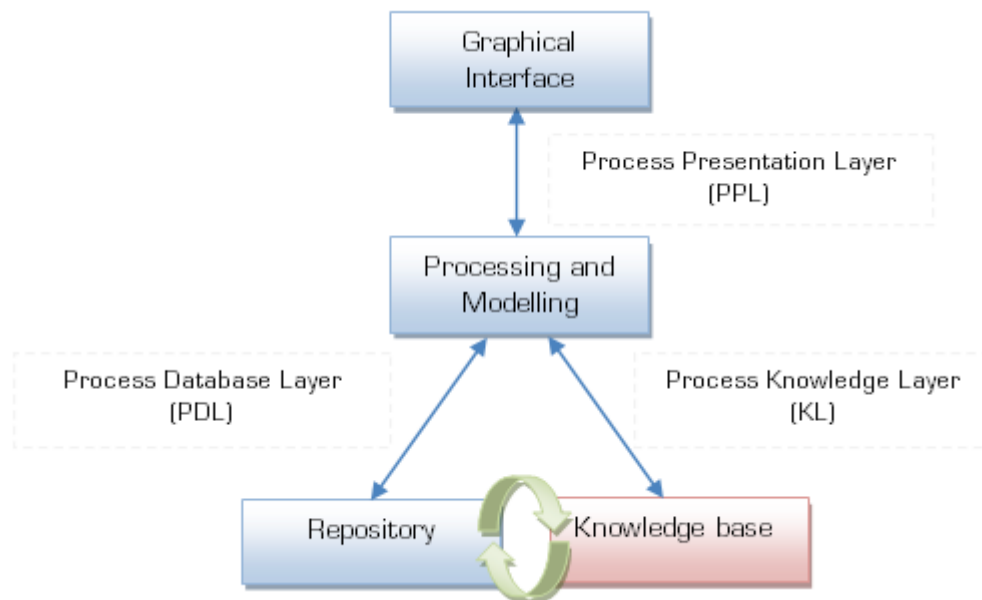


Figure 29: Implementation Architecture – Repository

### 6.5.1 Design Requirements

The Knowledge Base must be designed keeping three major requirements .i.e., Knowledge Extraction and Creation, Knowledge Store and Management and Knowledge Sharing and Usage in mind.

#### *Knowledge Extraction and Creation*

The designed Knowledge Base component must be capable of extracting and creating knowledge based on Meta data from normal raw data.

#### *Knowledge Store and Management*

The designed Knowledge Base component must be capable of storing and managing the extracted knowledge based Meta data.

#### *Knowledge Sharing and Usage*

The designed Knowledge Base component must be capable of providing knowledge based Meta data in a way then it can be easily shared and used.

### 6.5.2 I-SOAS Knowledge Base Categories

Following the standard categorization of Knowledge base [67], Knowledge Base is also divided into two categories. i.e., Knowledge Base Machine readable knowledge bases and Knowledge Base Human-readable knowledge bases.

Knowledge Base Machine-readable knowledge bases are designed to store knowledge in machine read and understandable format, consisting of logical set of instructions forming rules to describe knowledge. Whereas Knowledge Base Human readable knowledge bases are designed to extract knowledge from relevant information providing sources like white pages, manuals, organizational documents etc.

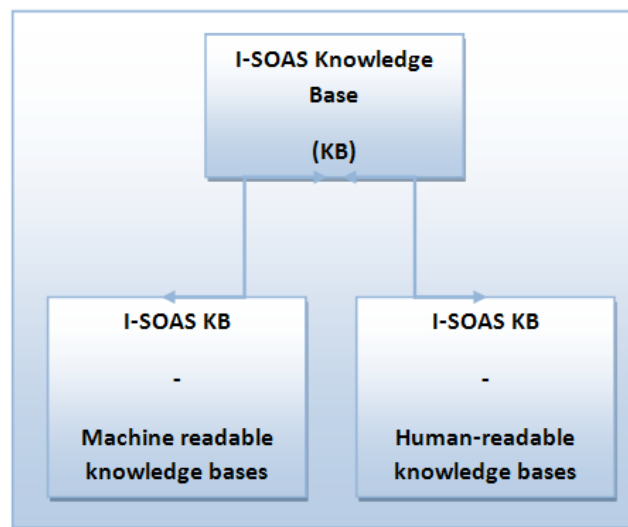


Figure 30: Knowledge Base Categories

### 6.5.3 Knowledge Base Cycle

The I-SOAS Knowledge Base works is basically divide in three main steps .i.e., *Knowledge Engineering*, *Knowledge Management* and *Knowledge Transfer* working in a cyclic clock wise order. In the first step knowledge engineering is performed to gather required knowledge then gathered knowledge is stored and maintained by knowledge management and finally managed knowledge will further be transferred to use.

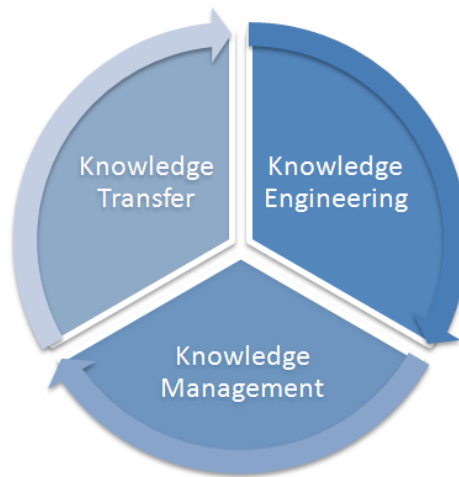


Figure 31: Knowledge Base Cycle

#### 6.5.4 System Sequence Design

I-SOAS Knowledge Base System Sequence Design consists of four main components. i.e., Process Knowledge Layer, Data Input and Capture or Create Knowledge and Manage Knowledge. These four components are supposed to perform certain jobs. The job of Process Knowledge Layer is to bring system data from I-SOAS Processing and Modelling and forward to Data Input. Then system data will be forwarded to Capture or Create Knowledge by Data Input, which will then create knowledge and forward created knowledge to Knowledge Management. Then Knowledge Management will first manage the created knowledge and then will send the final resultant knowledge to I-SOAS Processing and Modelling via Process Knowledge Layer.



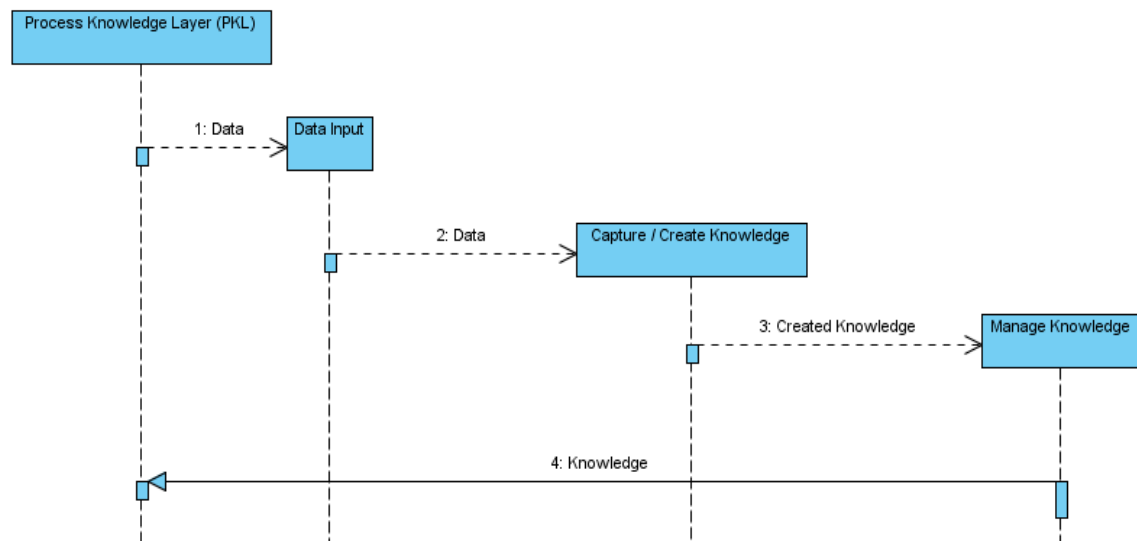


Figure 32: Knowledge Base System Sequence Design

### 6.5.5 Internal Work Flow

Knowledge Base internal workflow is basically depends upon the already above discussed Knowledge Based three steps .i.e., Knowledge Engineering, Knowledge Management and Knowledge Transfer cycle. Knowledge Base internal workflow is supposed to start with the knowledge gathering by either capturing the knowledge or by creating the knowledge (Knowledge Engineering). Resultant captured or created knowledge is first stored, organized and then categorized (Knowledge Management). Then resultant knowledge is further transferred to use by sharing and distributing it (Knowledge Transfer).

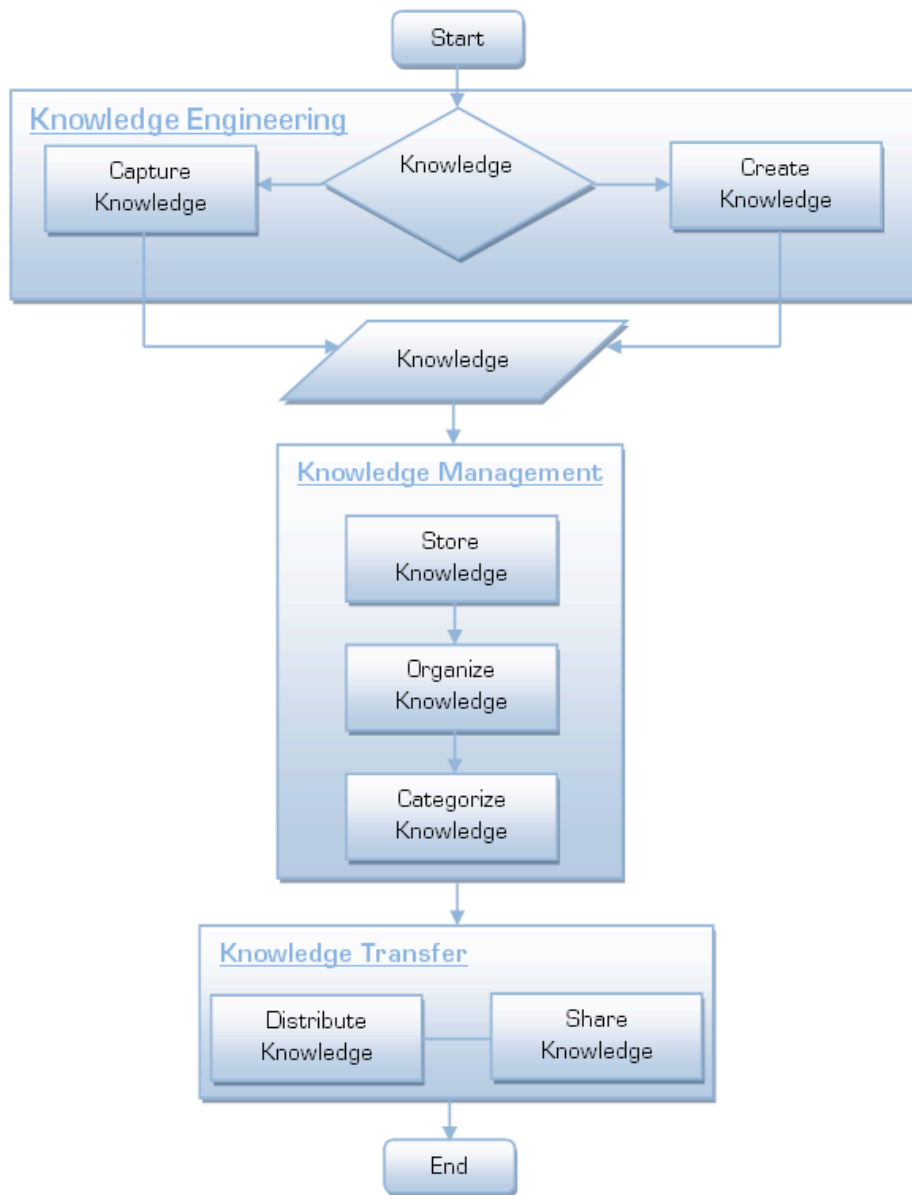


Figure 33: Knowledge Base Internal Work Flow

## 6.6 Process Presentation Layer

Process Presentation Layer is a communication layer between Graphical Interface and Processing & Modelling Modules of the I-SOAS implementation architecture. The expected job of this layer is to transfer user inputted set of instructions from Graphical Interface to Processing and Modelling & Processing module and transfer system outputted data from Processing and Modelling to Graphical Interface.

### 6.6.1 Design Requirements

Process Presentation Layer must be designed in a way that is can take data as user input from Graphical Interface and forward that to Processing and Modeling and then it can take data as system output from Processing and Modeling and forward to Graphical Interface.

### 6.6.2 System Sequence Design

Process Presentation Layer System Sequence Design is consists of three main components. i.e., Graphics Interface, Format Data and Processing and Modelling. These three components are supposed to perform certain jobs. The job of Graphics Interface is to send user-inputted data to Format Data, which will then change the format of input to Processing and Modelling input format and forward that inputted data to Processing and Modelling. Then in the same sequence but in reverse order, Processing and Modelling will send the system output to Format data, which will then change the format of outputted system data by Processing and Modelling to the acceptable data format of Graphics Interface and forward that outputted data to of Graphical Interface.

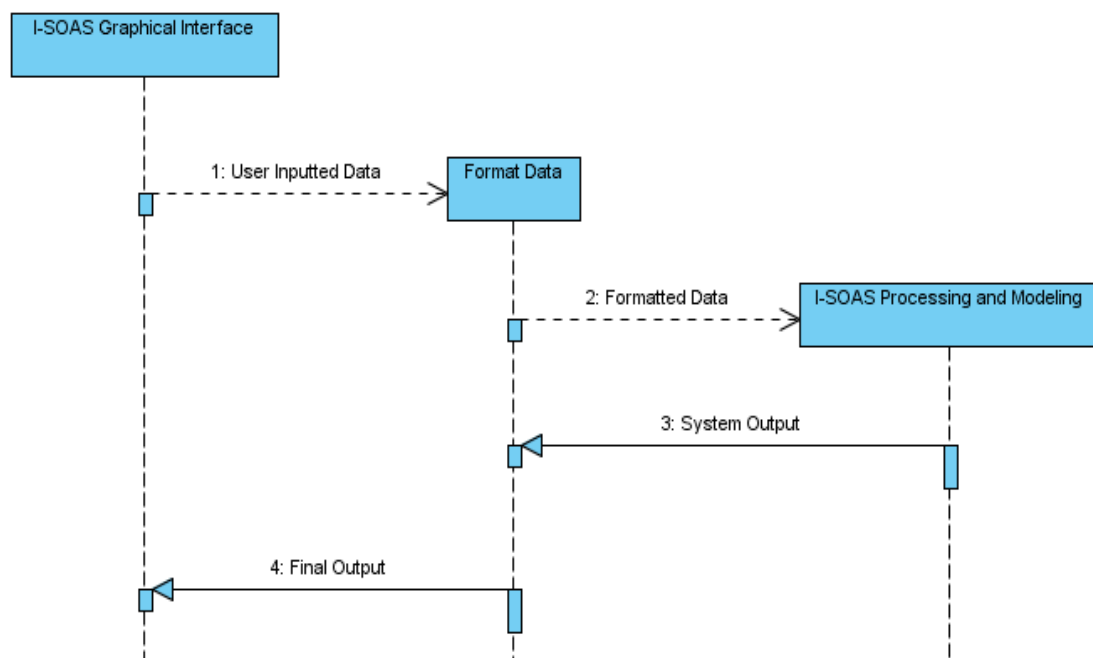


Figure 34: Process Presentation Layer System Sequence Design

### 6.6.3 Internal Work Flow

Process Presentation Layer internal work flow starts with the input data (from Graphical Interface or Processing and Modeling component). Input data first read and then analyzed, and then if inputted data is corrupt or wrong then its reported and control sent back to start for new input, but if the input data is fine then the sender is identified. If data is send by Graphical

Interface then it is formatted with respected to Processing and Modeling data format and if data is send by Processing and Modeling then it is formatted with respected to Graphical Interface data format.

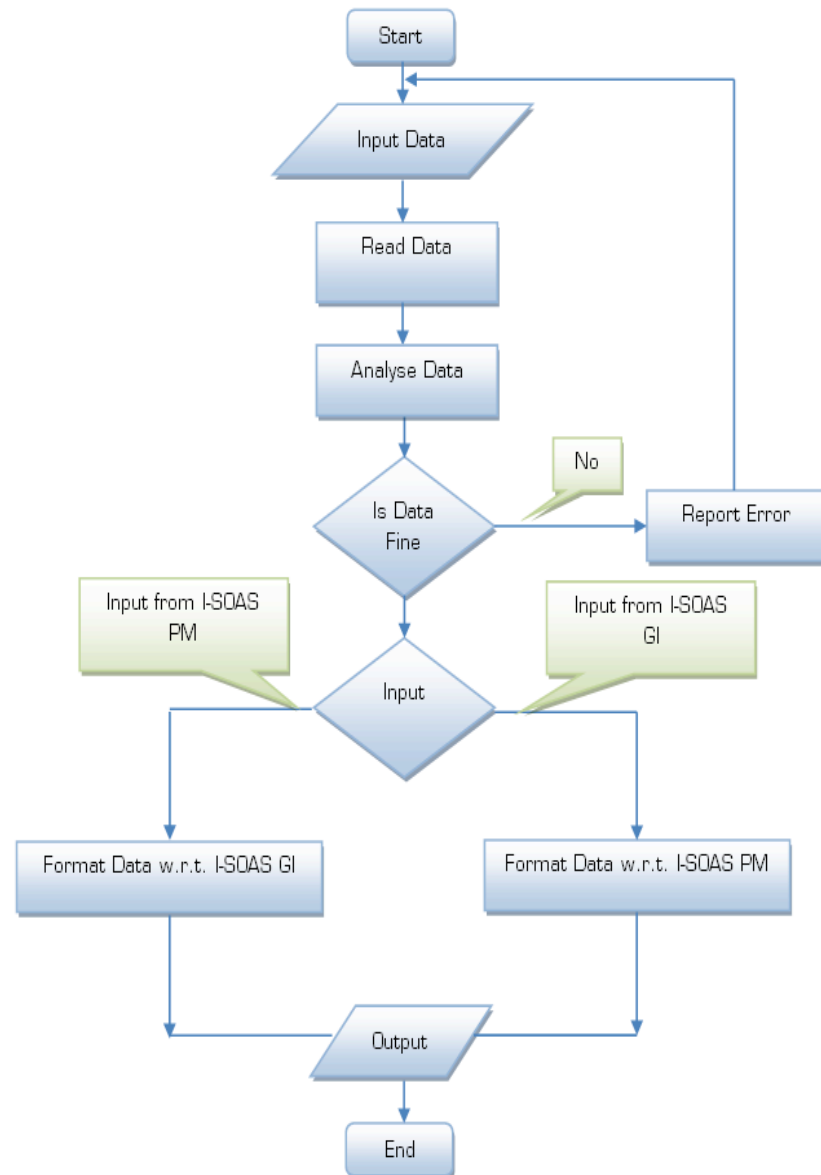


Figure 35: Process Presentation Layer Internal Work Flow

## 6.7 I-SOAS Process Database Layer

Process Database Layer is a communication layer between Repository and Processing & Modelling Modules of the I-SOAS implementation architecture. The expected job of this designed layer is simply to forward Modelling & Processing module's constructed query to Repository and return retrieved results from Repository to Modelling & Processing module.

### 6.7.1 Design Requirements

Process Database Layer must be designed in a way that is can take data from Processing and Modeling and store in to Repository.

### 6.7.2 System Sequence Design

PDL System Sequence Design is consists of three main components. i.e., Processing and Modelling, Format Data and Repository. These three components are supposed to perform certain jobs. The job of Processing and Modelling is to send data to Format Data, which will then change the format of data to Repository data format and forward that inputted data to Repository. Then in the same sequence but in reverse order, Repository will send the acknowledgement to Format data, which will then change the format of outputted acknowledgement by Repository to the acceptable data format of Processing and Modelling and forward that outputted data to of Processing and Modelling.

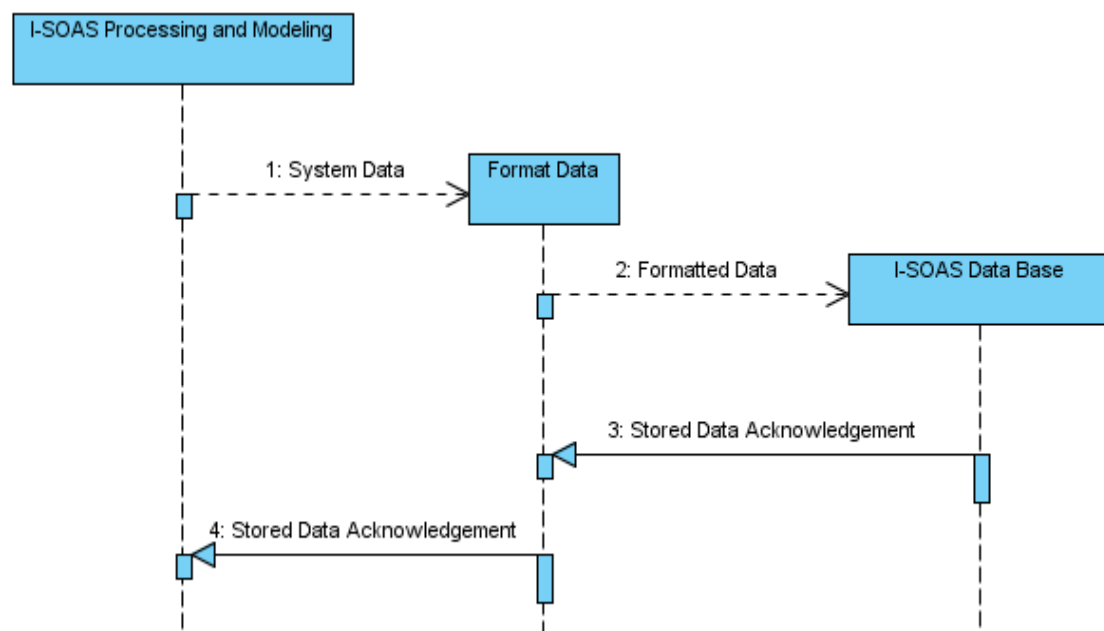


Figure 36: Process Database Layer System Sequence Design

### 6.7.3 Internal Work Flow

Process Database Layer internal workflow starts with the input data (from I-SAOS Processing and Modeling). Input data first read and then analyzed, and then if inputted data is corrupt or wrong then its reported and control sent back to start for new input, but if the input data is fine according to the input data database script (SQL Script) is generated and run to store and manipulate data.

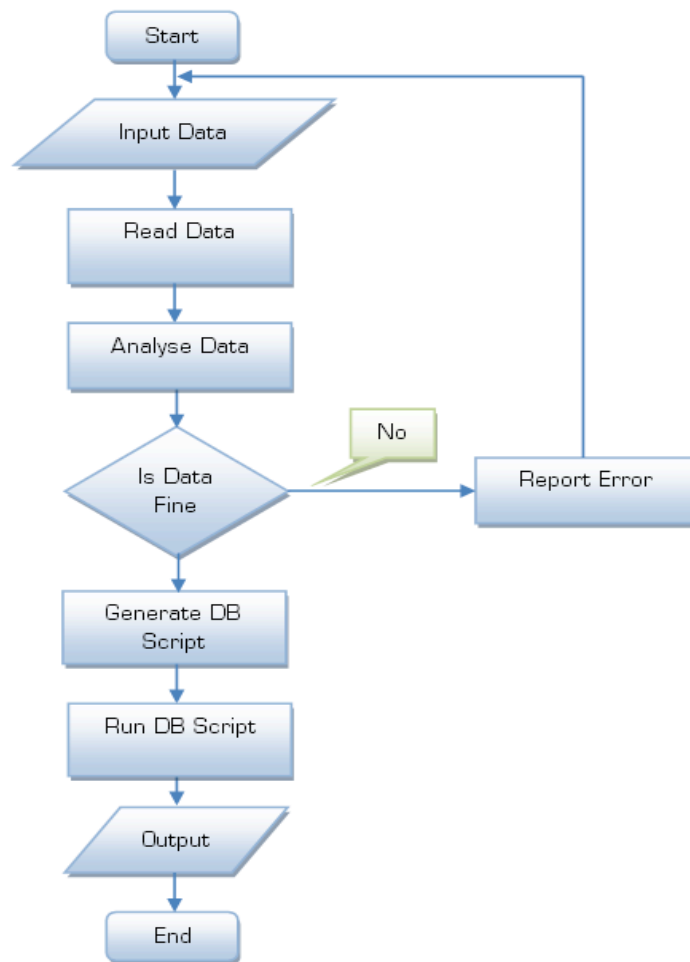


Figure 37: Process Database Layer Internal Work Flow

## 6.8 Process Knowledge Layer

Process Knowledge Layer is a communication layer between Knowledge base and Processing & Modelling Modules of the I-SOAS implementation architecture. The expected job this designed layer is simply to forward Modelling & Processing module's constructed semantic based query to Knowledge base and return extracted results from Knowledge base to Modelling & Processing module.

### 6.8.1 Design Requirements

Process Knowledge Layer must be designed in a way that is can take data Processing and Modeling and forward to Knowledge Base then it can take knowledge as output from Knowledge Base and return to Processing and Modeling.

## 6.8.2 System Sequence Design

Process Knowledge Layer System Sequence Design is consists of three main components. i.e., Processing and Modelling, Format Data and Knowledge base. These three components are supposed to perform certain jobs. The job of Processing and Modelling is to send data to Format Data, which will then change the format of data to Knowledge Base data format and forward that inputted data to Knowledge Base. Then in the same sequence but in reverse order, Knowledge Base will send the knowledge to Format data, which will then change the format of outputted knowledge by Knowledge Base to the acceptable data format of Processing and Modelling and forward that outputted data to of Processing and Modelling.

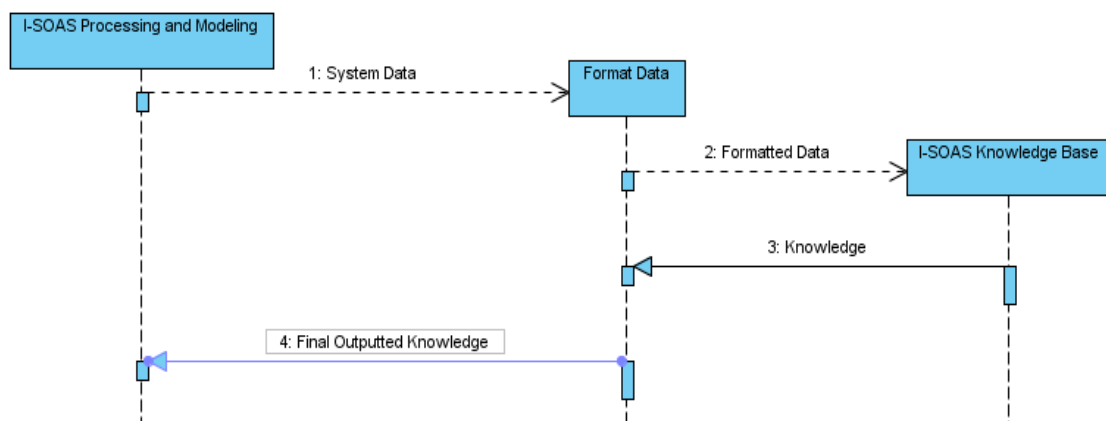


Figure 38: Process Knowledge Layer System Sequence Design

## 6.8.3 Internal Work Flow

Process Knowledge Layer internal work flow starts with the input data (from Knowledge Base or Processing and Modeling component). Input data first read and then analyzed, and then if inputted data is corrupt or wrong then its reported and control sent back to start for new input, but if the input data is fine then the sender is identified. If data is send by Knowledge Base then it is formatted with respected to I-SAOS Processing and Modeling data format and if data is send by I-SAOS Processing and Modeling then it is formatted with respected to Knowledge Base data format.

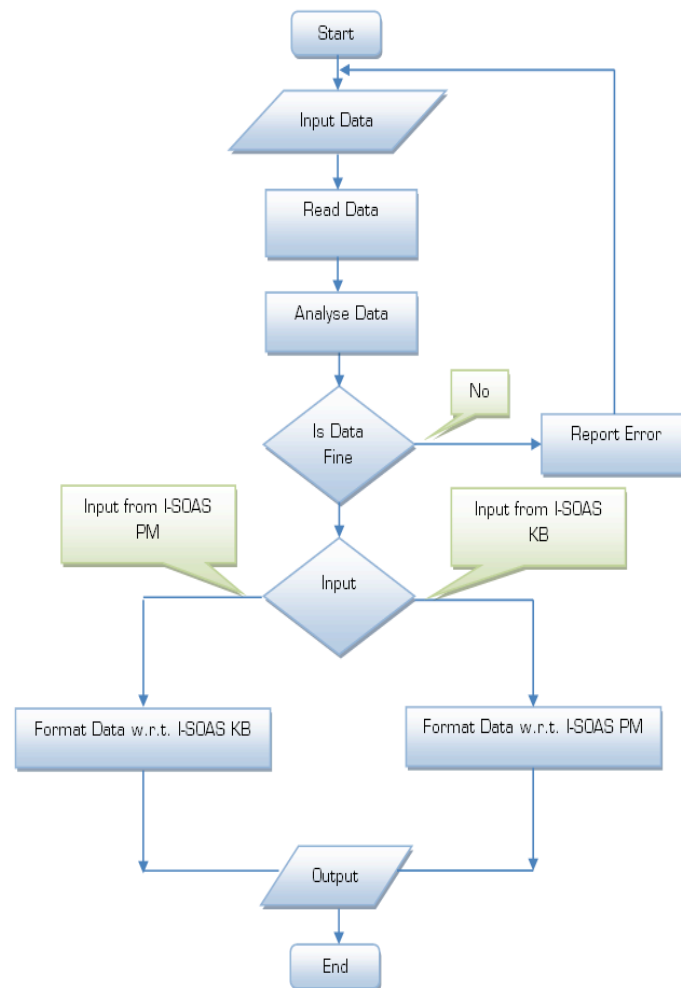


Figure 39: Process Knowledge Layer Internal Work Flow



## 7 I-SOAS Prototype

### 7.1 I-SOAS Desktop Application – Information Processor

Following the designed implementation designs and meeting design requirements, I have started the implementation of I-SOAS. By looking at all the implementation designs its quite obvious to say that the whole development process of I-SOAS Desktop Application will take around 48 months of time and number of recourses to develop.

At the moment I-SOAS Desktop Application is available in its prototyped shape after hard core one man development. I-SOAS Desktop Application's currently available version is capable of

- Running as a stable application
- Taking input from user in the form of text file
- Providing little bit flexibility in the organization and presentation of available options in the graphical machine interface
- Creating dynamic database to store user input based information
- Connect and disconnect different databases
- Opening database to view the stored information
- Configuring files based on their types (extensions)
- Applies Lexer to input (file(s))
- Applies Parser to input (file(s))
- Stored resultant information in database
- Building queries to extract stored results
- Produce different kinds of visualizations of stored user based and system processed information in database

To become capable of performing above mentioned jobs I-SOAS is currently divide into ten different front end screens .i.e., Information Processor, Create Database, Open Database, Connect Database, File Configuration, Relation Builder, Query Builder, and Result Visualizer.

### 7.1.1 Information Processor

Shown in figure is the main front end of the I-SOAS information processor.

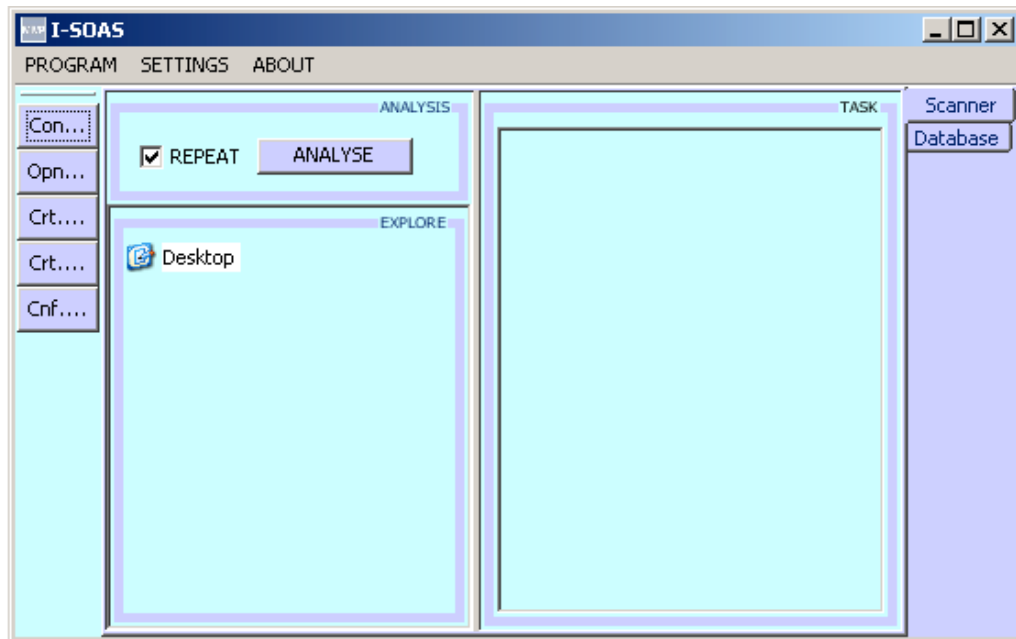


Figure 40: Information Processor

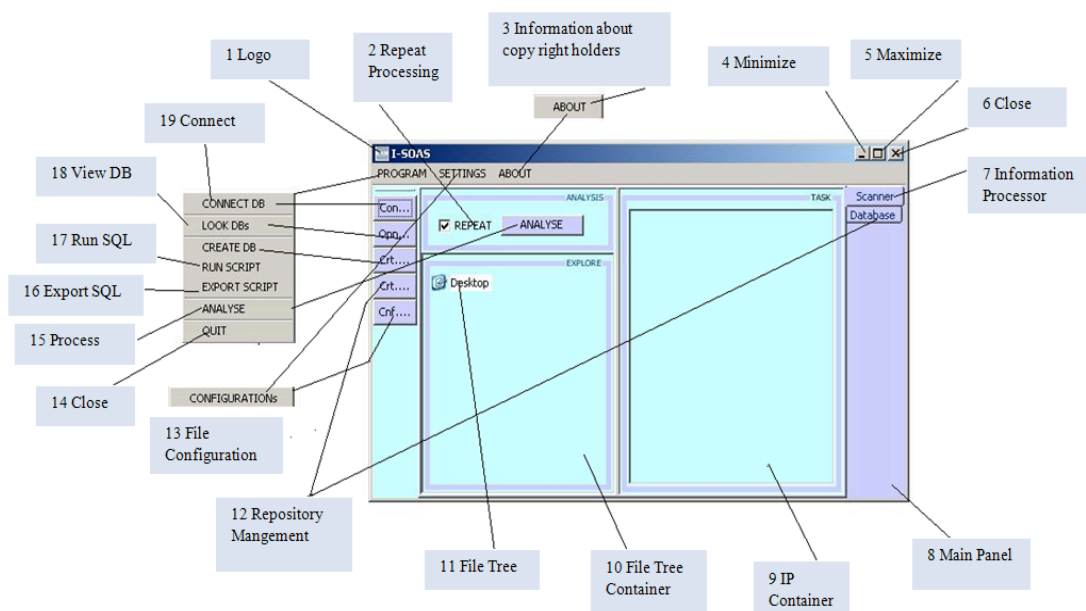


Figure 41: Information Processor Description

The whole graphical machine interface is consists of 19 main options / functionalities explained below in table 1.

No	Option	Description
1	Logo	Represents owner organization.
2	Repeat Processing	Repeats the process of search for the relevant file then process that.
3	About	About Menu opens About interface
4	Minimize	Minimizes interface
5	Maximize	Maximizes interface
6	Close	Closes the application
7	Information Processor	Starts information processing
8	Main Panel	Contains all the controls
9	IP Container	Presents the status of information processing

10	File Tree Container	Contains the file tree
11	File Tree	Maps the directory information into tree
12	Repository	Opens database interface of I-SOAS
13	File Configuration	Inputs processable file types
14	Close	Closes the application
15	Process	Starts information processing
16	Export SQL	Exports and saves the SQL based information
17	Run SQL	Runs SQL to manipulate I-SOAS Database
18	View DB	Provides the view of database
19	Connect	Provide interface to establish and test database connection

Table 1: Information Processing

### 7.1.2 Create Database

Shown in figure is the main front end of the Create Database.

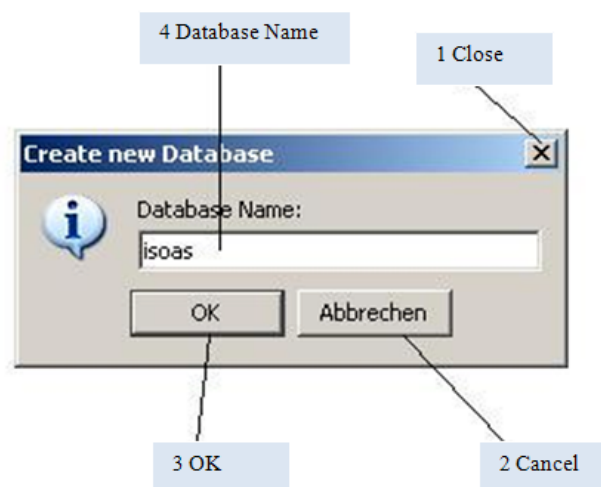


Figure 42: Create new database

The whole graphical machine interface is consists of 4 main functionalities explained below in table 2.

No	Option	Description
1	Close	Closes the interface
2	Cancel	Cancels the operation and close interface
3	OK	Creates new database
4	Database Name	Allows to enter the new database name

Table 2: Create Database

### 7.1.3 Open Database

Shown in figure is the main front end of the Open Database.

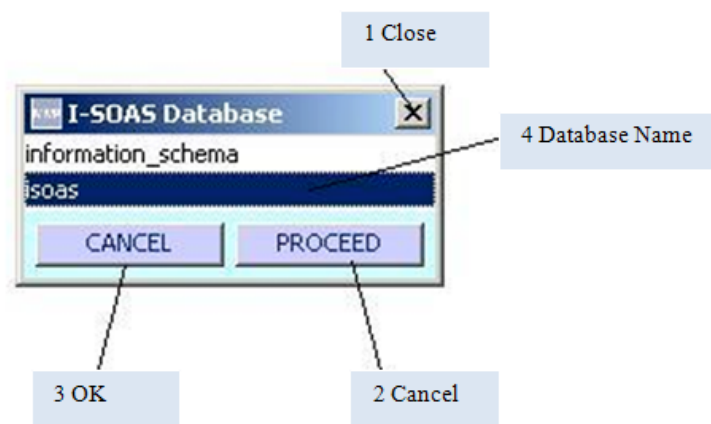


Figure 43: Open database

The whole graphical machine interface is consists of 4 main functionalities explained below in table 3.

No	Option	Description
1	Close	Closes the interface

2	Cancel	Cancels the operation and close interface
3	OK	Allows to selects database
4	Database Name	Presents information about available databases

Table 3: Open Database

### 7.1.4 Connect Database

Shown in figure is the main front end of the Connect Database.

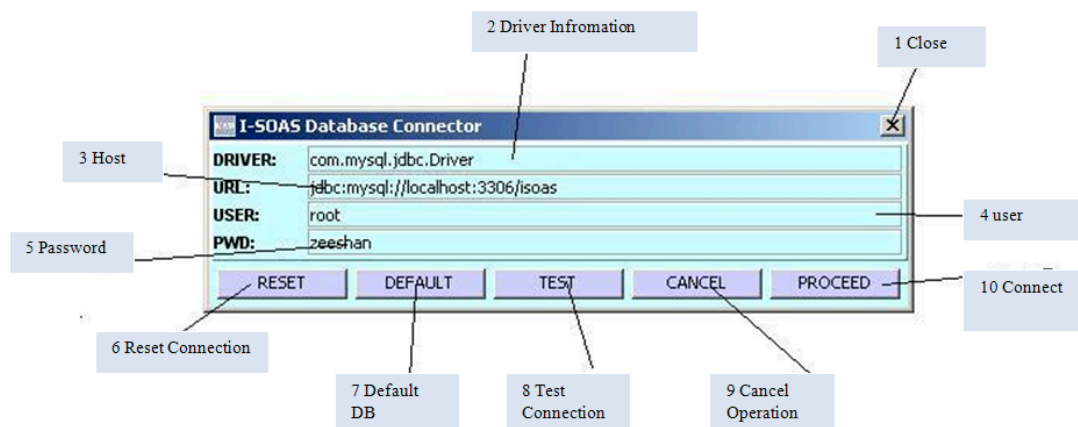


Figure 44: Connect database

The whole graphical machine interface is consists of 9 main functionalities explained below in table 4.

No	Option	Description
1	Close	Closes the interface
2	Driver Information	Provides information about used drive to connect database, user can alter as well.
3	Host	Provides information about host of database, user can alter as well.
4	User	Provides information about user, user can alter as well.

5	Password	Provides information about password, user can alter as well.
6	Reset Connection	Connects to default database
7	Default DB	Sets current database information as Default Database
8	Test Connection	Tests database connection
9	Cancel Operation	Cancels the operation and close interface

Table 4: Connect Database

### 7.1.5 File Configuration

Shown in figure is the main front end of the File Configuration.

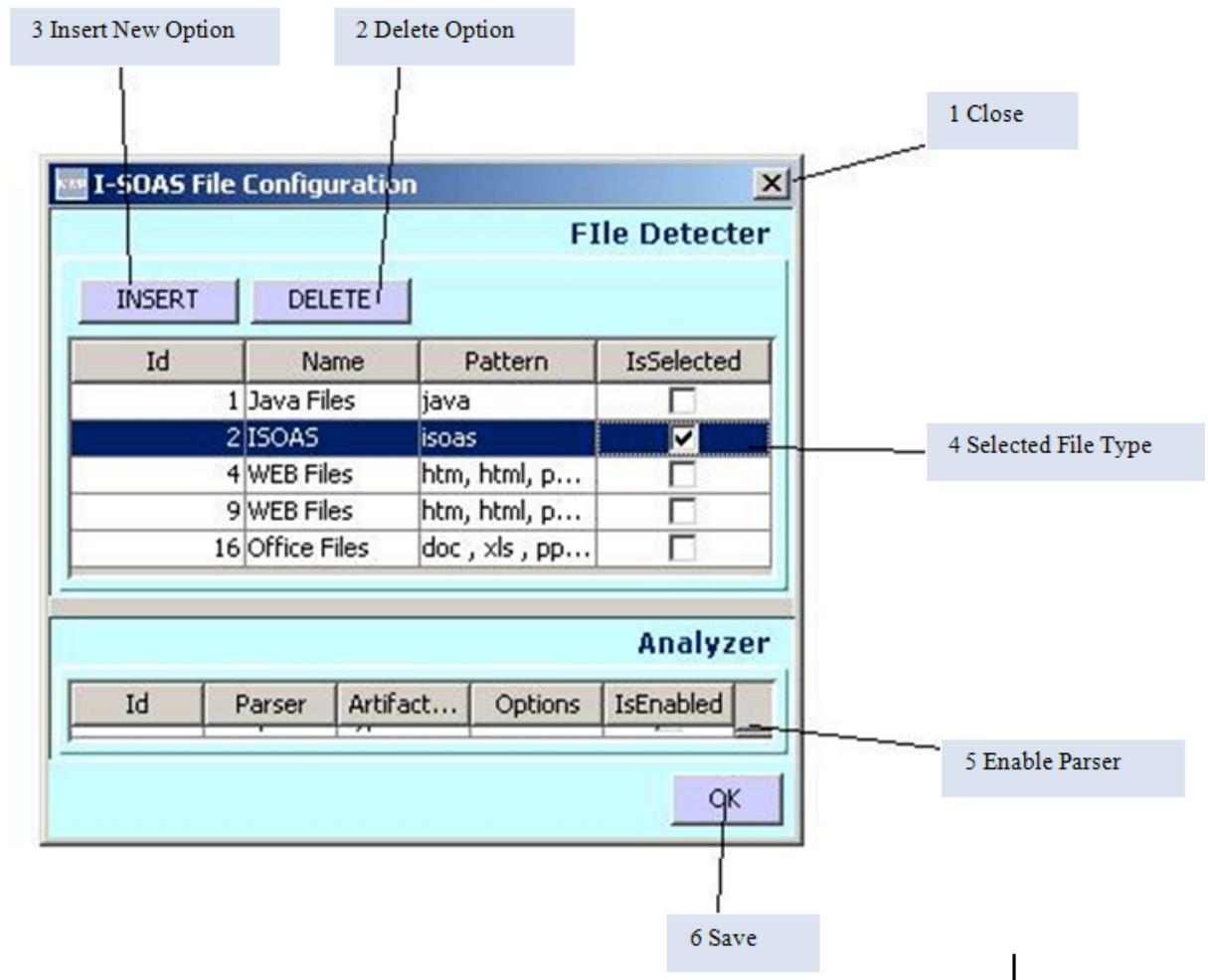


Figure 45: File Configuration

The whole graphical machine interface consists of 6 main functionalities explained below in table 5.

No	Option	Description
1	Close	Closes the interface
2	Delete	Deletes selected file type
3	Insert New	Allows user to insert a new file type
4	Selected File	Allows user to select file type to select and process by information processor



	Type	
5	Enable Parser	Enable Language Parser
6	Save	Saves file configuration information

Table 5: File Configuration

### 7.1.6 Relation Builder

Shown in figure is the main front end of the Relation Builder.

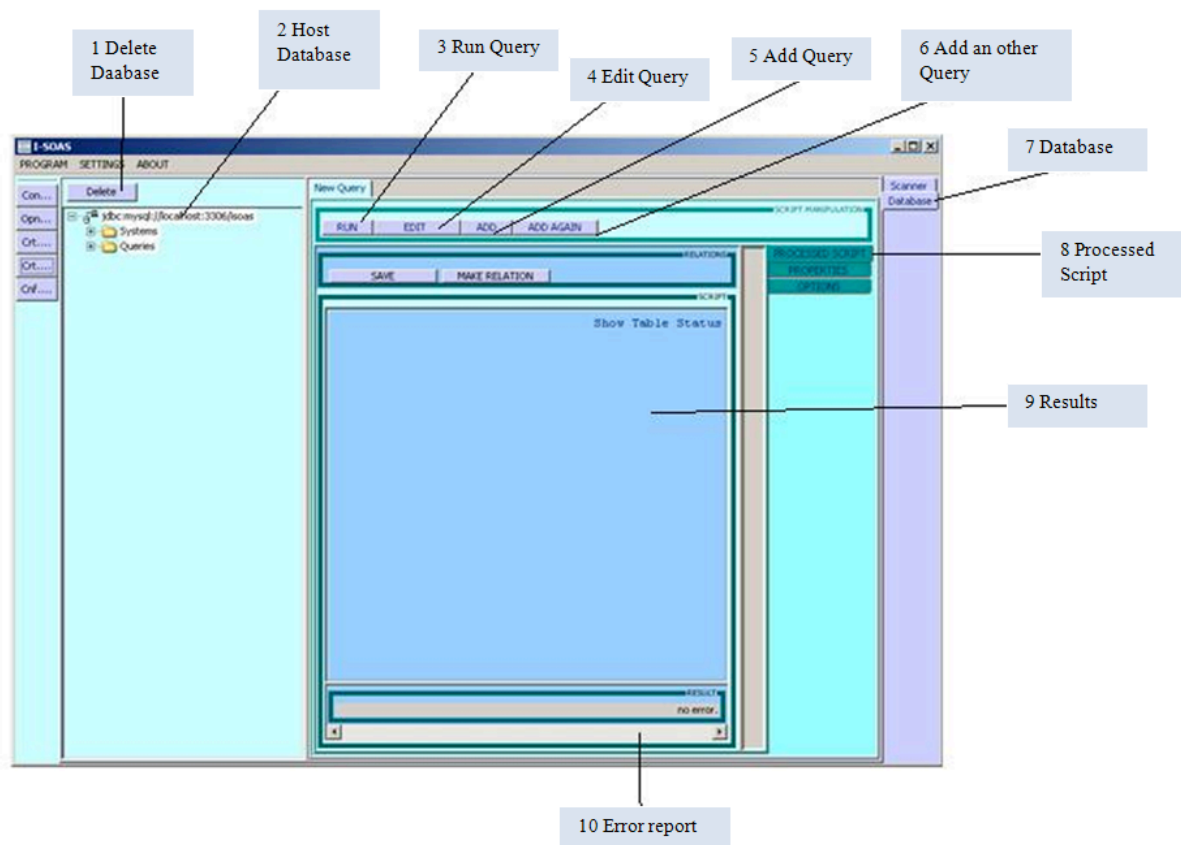


Figure 46: Relation Builder

The whole graphical machine interface is consists of 10 main functionalities explained below in table 6.

No	Option	Description
----	--------	-------------

1	Delete Database	Deletes selected database
2	Host Database	Provides information about host database
3	Run Query	Runs the query
4	Edit Query	Allows user to edit query
5	Add Query	Allows user to add query
6	Add Another Query	Allows user to add another query
7	Database	Enables database interface of I-SOAS
8	Processed Script	Interface to add, edit, run and save query script
9	Results	Presents extracted results from database
10	Error Report	Report effort (if any)

Table 6: Relation Builder

### 7.1.7 Query Builder

Shown in figure is the main front end of the Query Builder.

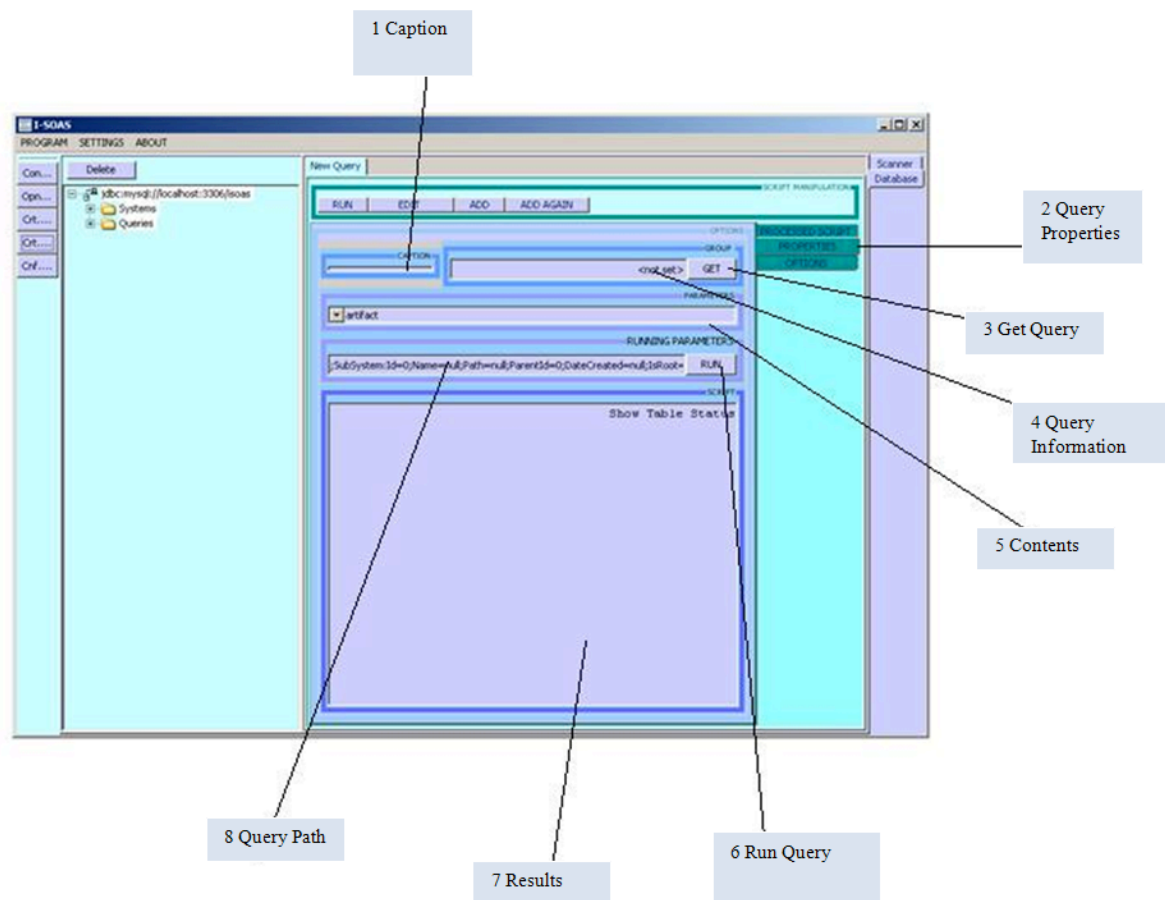


Figure 47: Query Builder

The whole graphical machine interface consists of 8 main functionalities explained below in table 7.

No	Option	Description
1	Caption	Allows user to set the caption of the query
2	Query Properties	Interface to set and get query properties
3	Get Query	Opens interface to select already stored query to use
4	Query Information	Presents the information about query

5	Contents	These are the parameters to the query
6	Run Query	Runs the query
7	Results	Presents the resultant extracted information from database
8	Query Path	Presents the query path with parameters

Table 7: Query Builder

Query Builder contains a short interface to build queries table, shown below in figure.

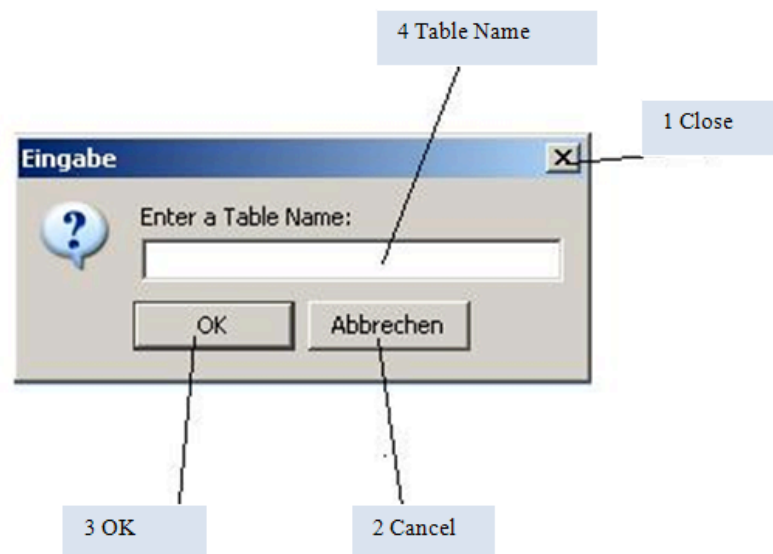


Figure 48: Query Table

The whole graphical machine interface is consists of 4 main functionalities explained below in table 8.

No	Option	Description
1	Close	Closes the interface
2	Cancel	Cancels the operation and close interface
3	OK	Creates new table in database

4	Table Name	Allows to enter new table name
---	------------	--------------------------------

Table 8: Query Table

Moreover Query Builder also contains a short interface to save build queries, shown below in figure Query Saver

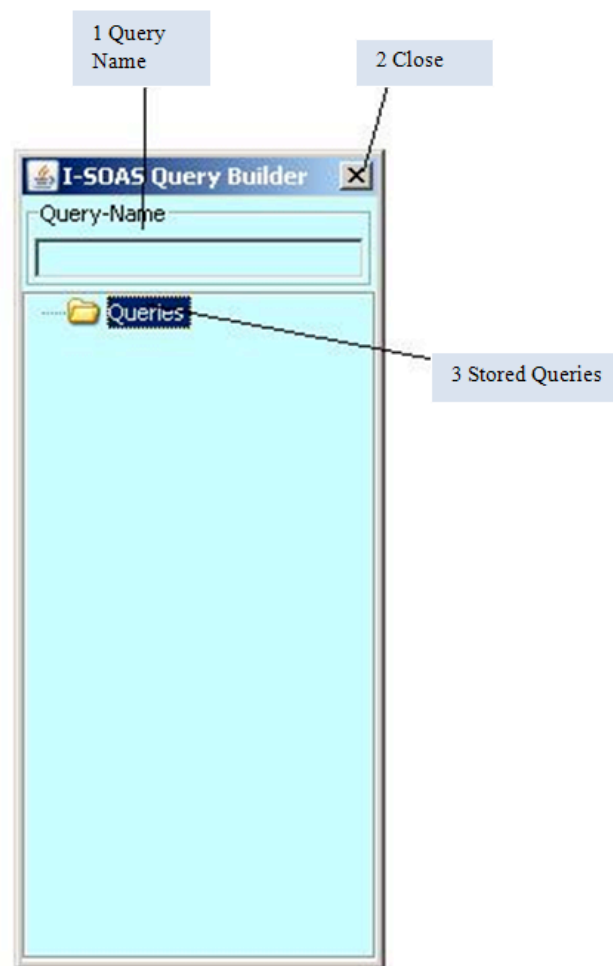


Figure 49: Query Saver

The whole graphical machine interface is consists of 4 main functionalities explained below in table 9.

No	Option	Description
----	--------	-------------

1	Query Name	Allows to enter query name to select
2	Close	Closes the interface
3	Stored Query	Shows already stored queries

Table 9: Query Saver

### 7.1.8 Result Visualizer

Shown in figure is the main front end of the Result Visualizer.

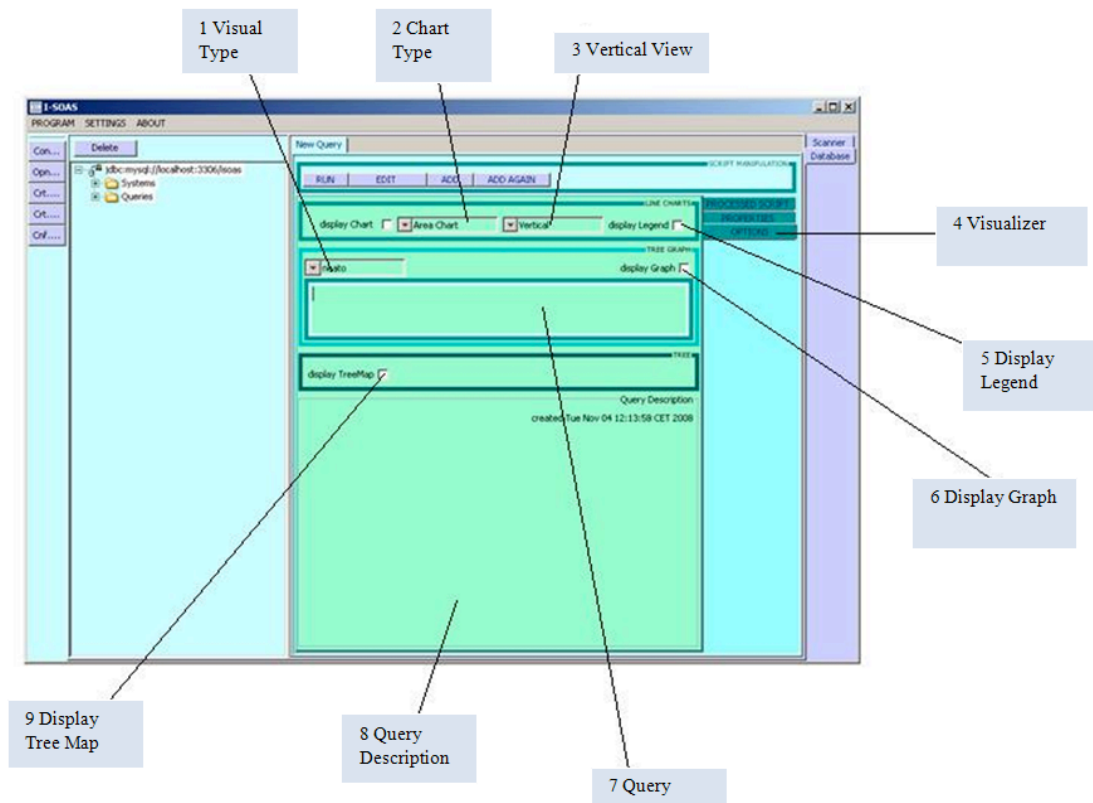


Figure 50: Result Visualizer

The whole graphical machine interface is consists of 9 main functionalities explained below in table 10.

No	Option	Description
----	--------	-------------

1	Visual Type	Allows user to select visual type
2	Chart Type	Allows user to select chart type
3	Vertical View	Allows user to select vertical or horizontal chart
4	Visualizer	Interface to visualize stored dummy and analyzed information in database
5	Display Legend	Displays legendry chart (big one)
6	Display Graph	Allows user to select to display graph
7	Query	Presents running query, also allows user to edit or type new query
8	Query Description	Presents the description of used query
9	Display Tree Map	Allows user to select to display tree map

Table 10: Result Visualizer

## 7.2 I-SOAS Web Application – Graphical Interface & Search

Following the designed implementation designs and meeting design requirements, I have started the implementation of I-SOAS Web Application. By looking at all the implementation designs its quite obvious to say that the whole development process of I-SOAS will take around 48 months of time and number of recourses to develop.

At the moment I-SOAS Web Application is available in its prototyped shape after hard core three man development. I-SOAS Web Application's currently available version is capable of

- Running as a stable application
- Providing information about research project
- Providing security system for user authentication and verification

- Providing independent SQL based static search
- Proving natural language based search using I-SOAS Desktop Application
- Providing flexible graphical user interface with several different options

### 7.2.1 Main Page

This is the main page of I-SOAS Web Application consisting of five main page links .i.e., Homepage, Search, About us, Teams and Components as shown in Figure.

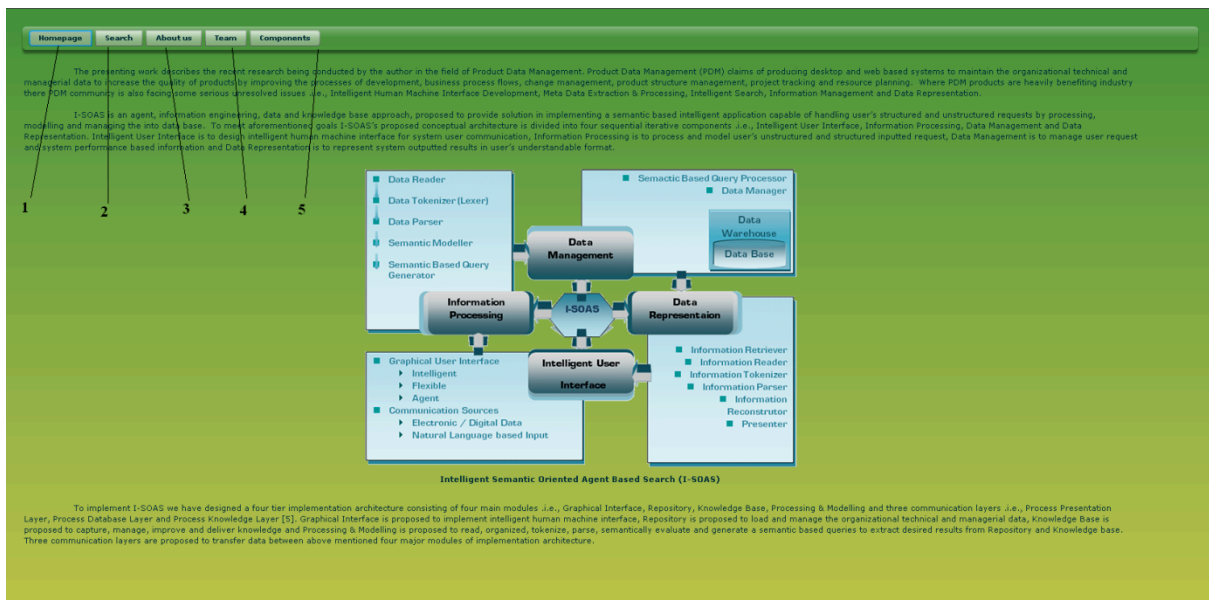


Figure 51: Main Page

No	Option	Description
1	Homepage	Link to the main page of application, providing links to other pages and information about the project I-SOAS
2	Search	Link to the Search page of application. Moreover search page also contains the secure login system to register new user, authenticate and verify the old user.
3	About Us	Link to the About US page of application, providing the information about Contacts.
4	Team	Link to the Team page of application, providing the information about team involved during the project.



5	Components	Link to the Component page of application
---	------------	---

Table 11: Main Page

## 7.2.2 Graphical User Interface

This is the graphical user interface of the I-SOAS Web Application. This page can be accessed using link Components. Graphical user interface page is divided into sections as shown in Figures

1. Default Graphical Interface
2. User Graphical Interface

The default graphical interface is basically the graphical interface with some limited and basic options which can be accessed by the user and the guest. But the User graphical interface can only be accessed by the user after logging into the application with authenticated user name and password.

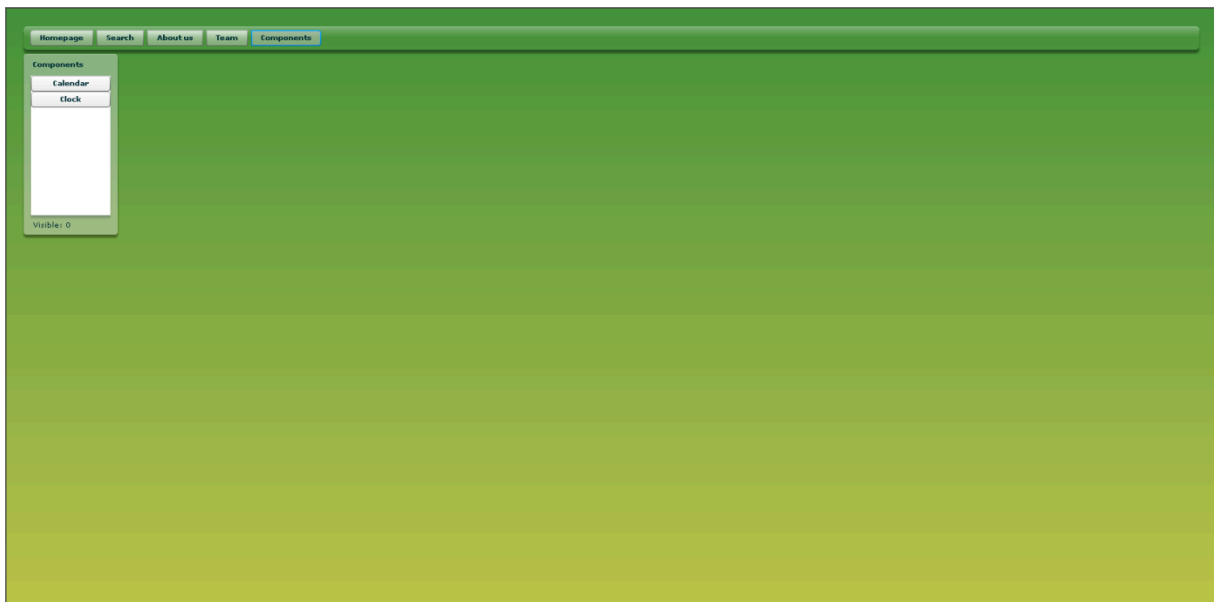


Figure 52: Default Graphical Interface – Without User login

User graphical interface is the actual interface presenting the prototype definition of proposed idea about Intelligent Graphical Interface of I-SOAS conceptual design and resultant implementation of designed Draft Graphical Interface designs of I-SOAS implementation design.

Meeting the design requirements of Draft Graphical Interface, this implemented prototype version of I-SOAS Graphical Interface is flexible and capable of

- Providing standard graphical interface, designed by system
- Providing flexible graphical user interface, so the user can redesign and reconfigure the interface itself to accommodate specific needs by Mouse Click and Drag Drop options
- Providing several different options to the user for GUI designing like user can change the look and feel by
  - changing background,
  - changing mixing background colors,
  - changing font colors,
  - changing background text colors,
  - adding some image at the background,
  - adding components from main component bar,
  - deleting added components to the interface
  - altering the placement positions
- Providing option to every user to save his own deigned GUI, so then next time if user comes online then he will be provided his own designed GUI, not the default one, but he will still have the option to redesign or alter or restore the default GUI.

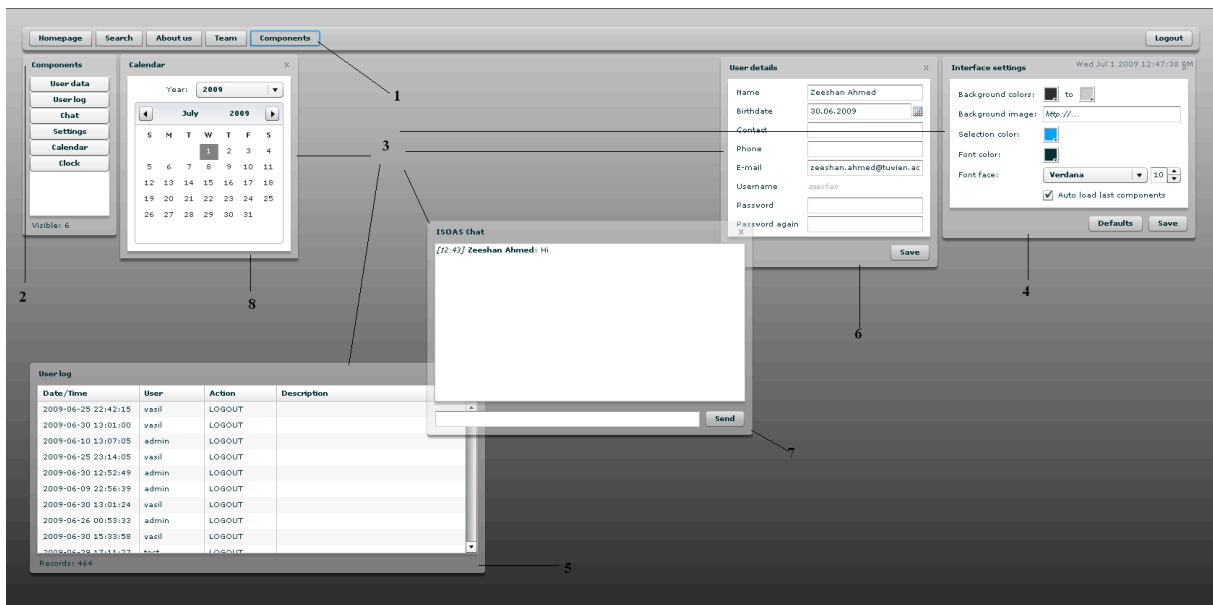


Figure 53: Graphical Interface –After User login

No	Option	Description
1	Main Component Link	To enable the Graphical Interface
2	Component Try	Providing all GUI options to the user to click and use
3	Components	All currently available components
4	Interface Setting	This components if used to change the outlook of the graphical interface by changing the color schemes and adding or removing image to the main interface
5	User Log	This components provides the detail of all the operations performed during the use of graphical interface, but this components if only visible to the user with administrative rights
6	User Details	This components provides the options to enter and alter user details

7	Chat	This component is providing option for in house chat to the login users to improve in house communications
8	Calendar	This is the simple calendar to enable user with date.

Table 12: Graphical Interface Page – User Login

### 7.2.3 Search

This is the search module I-SOAS Web Application. This page can be accessed using Search Link of the main links. Search module is based on both the design requirements of Information Processing and Intelligent User Interface of the I-SOAS conceptual architecture.

As I have earlier mentioned in the requirements of Intelligent User Interface, that the graphical user interface should provide two options to search, first to take system based instruction and then to take natural language based instruction as input and perform search, and as the result of both the searches present the findings.

Moreover, according to the earlier mentioned design requirements of Information Processing module, search module should perform Input Analysis, Information Processing and Building Final Output.

This implemented version of I-SOAS Search is capable of

- Allowing user to login to make search, because only authenticated user can search
- Providing option to user to enter the instruction in Natural Language as well as SQL Query Format
- Providing Natural Language Mode of search, to enter natural language based query like “I am looking for PDM” etc and search
- Providing SQL Query Mode of search, to enter SQL Query like “Select \* from Document” etc and search
- Providing option to make search after entering the instruction
- Providing results back to the user
- Providing option to log of from the system

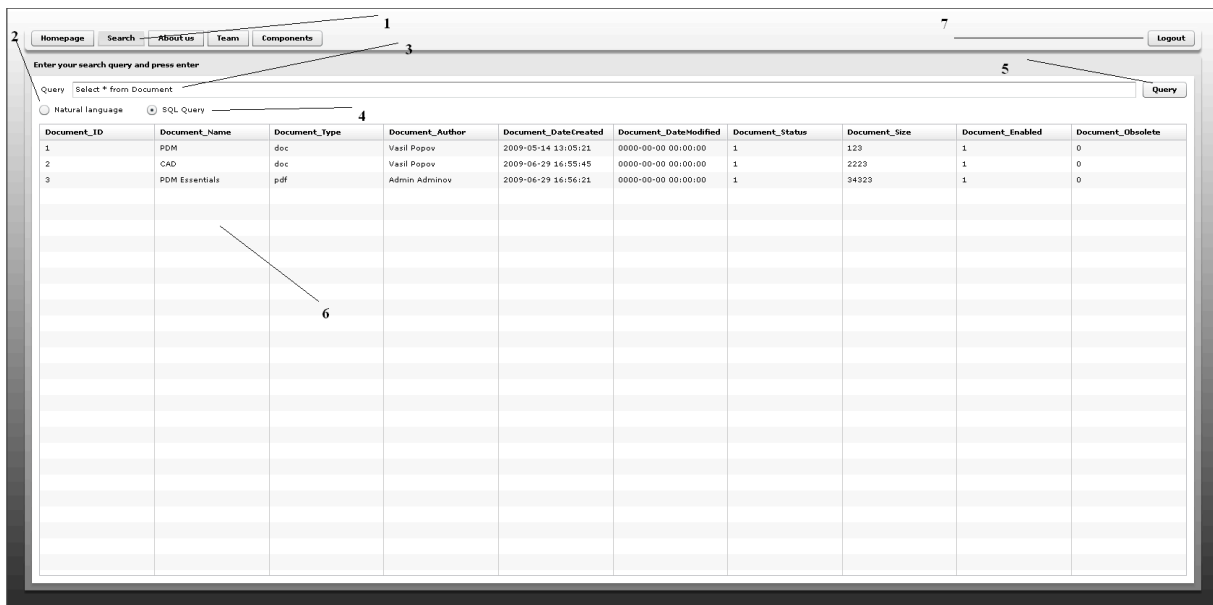


Figure 54: SQL based Search

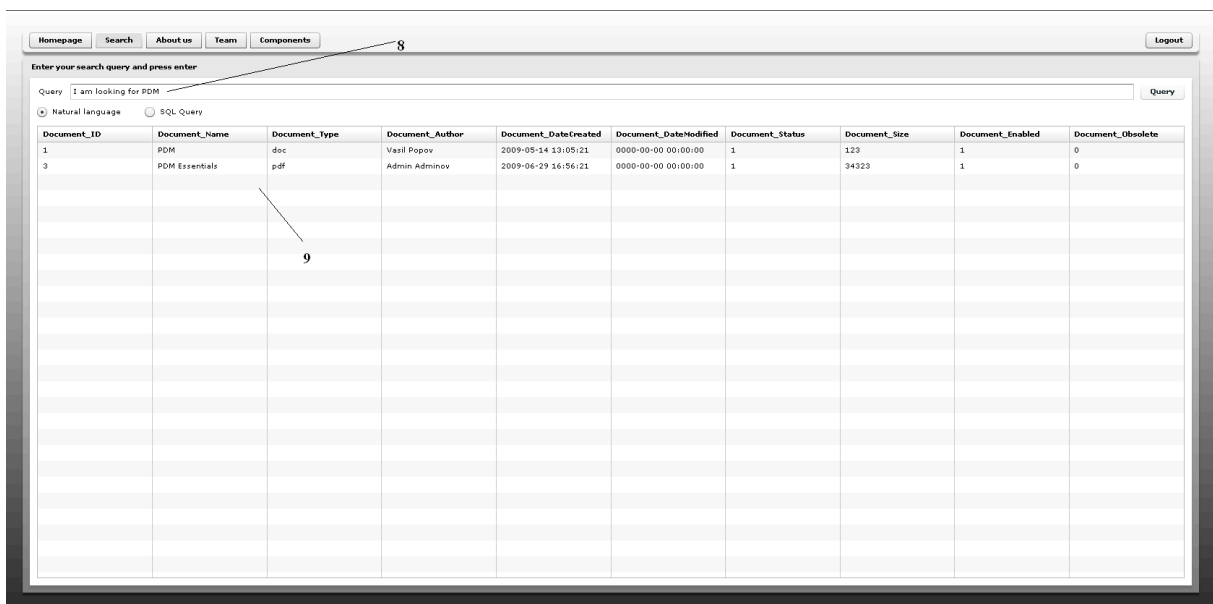


Figure 55: Natural Language based Search

No	Option	Description
1	Search Link	To enable the search

2	Natural Language	Option to enable natural language based search more write natural language based instruction to search
3	SQL Query Text	Edit box provided to use to write SQL Query to search
4	SQL Query	Option to enable SQL based search more write natural language based instruction to search
5	Query Button	Button to make search
6	Results	Results, resulted as the result of SQL Search
7	Logout	To logout user
8	NL Query	Edit box provided to use to write natural language based instruction Query to search
9	Results	Results, resulted as the result of NLP Search

Table 13: Search Page

## 8 Summary

Product Data Management (PDM) is a digital electronic way of maintaining organizational data to maintain and improve the quality of products and followed processes. PDM based products mainly maintain the information about the organization including personal involved in managerial and technical operations, running projects and manufacturing products.

Where PDM based products are heavily benefiting industry there PDM community is also facing some serious unresolved issues .i.e., successful secure platform independent PDM system implementation, PDM system deployment and reinstallation, static and unfriendly machine interface, unintelligent search and scalable standardized framework.

Many approaches and solutions systems including Meta-phase (SDRC), SherpaWorks (Inso), Enovia (IBM), CMS (WTC), Windchill (PTC), and Smarteam (Smart Solutions) [2] are proposed targeting these above mentioned problem oriented issues all together and on individual basis but still there is no as such one promising approach or product exists which claims of providing all the solutions.

Targeting some of above-mentioned PDM based problem oriented issues. i.e., Static and Unfriendly Graphical User Interface, Static and Unintelligent Search and Platform Independent System, I have also proposed an approach called Intelligent Semantic Oriented Search (I-SOAS).

I-SOAS is an agent, information engineering & modeling, database and knowledge base approach, proposed to provide solution in implementing a semantic based intelligent application capable of handling user's structured and unstructured requests by processing, modeling and managing the into data base. To meet aforementioned goals I-SOAS's proposed conceptual architecture is divided into four sequential iterative components. i.e., Intelligent User Interface (IUI), Information Processing (IP), Data Management (DM) and Data Representation (DR). IUI is proposed to design intelligent human machine interface for system user communication, IP is proposed to process and model user's unstructured and structured inputted request by reading, lexing, parsing, and semantic modeling, DM is proposed to manage user request and system performance based information and DR is proposed to represent system outputted results in user's understandable format.

To implement I-SOAS I have designed an implementation architecture consisting of four main modules. i.e., I-SOAS Graphical Interface, I-SOAS Repository, I-SOAS Know-ledge Base, I-SOAS Processing Modelling and three communication layers .i.e., Process Presentation Layer (PPL), Process Database Layer (PDL), and Process Knowledge Layer (PKL).

I-SOAS Graphical Interface is proposed to implement intelligent human machine interface, I-SOAS Repository is proposed to load and manage the organizational technical and managerial data, I-SOAS Knowledge Base is proposed to capture, manage, improve and deliver knowledge and ISOAS Processing and Modelling is proposed to read, organized, tokenize, parse, semantically evaluate and generate a semantic based queries to extract desired results

from I-SOAS Repository and I-SOAS Knowledge base. Three communication layers are proposed to transfer data between above-mentioned four major modules of I-SOAS Implementation Architecture.

As this an ongoing in process research project, so, right now the development of main four modules i.e., I-SOAS Graphical Interface, I-SOAS Repository, I-SOAS Know-ledge Base, I-SOAS Processing Modelling and three communication layers. i.e., Process Presentation Layer (PPL), Process Database Layer (PDL), and Process Knowledge Layer (PKL) is in process, but I have developed the prototype version of I-SOAS.

At the moment, developed prototype version (desktop and web) of I-SOAS is available, which is capable of running standalone and providing intelligent search and flexible graphical user interface.

In future if this research continues then I am hoping for the full version of I-SOAS web application consisting of complete proposed graphical interface options and the development of knowledge base system.

Thanks...



## 9 References

- [1] Sean B. Palmer, “The Semantic Web: An Introduction”, viewed February, 28. 2007, <<http://infomesh.net/2001/swintro>>
- [2] World Wide Web, March 2007, <[http://en.wikipedia.org/wiki/World\\_Wide\\_Web](http://en.wikipedia.org/wiki/World_Wide_Web)>
- [3] Wernher Behrendt, “Ambient Intelligence Semantic Web or Web 2.0”, In Proceedings of Semantic Content Engineering 2005, Salzburg 2005
- [4] Klaus Tochtermann and Herman Maurer, “Semantic technologies – An Introduction”, Semantic Technologies Showcase the Austrian Situation, pp. 15-20, 2006
- [5] Tim Berners-Lee, James Hendler and Ora Lassila, “The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”, May 2001, Viewed February 2007, <<http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>>
- [6] Danbri, “Metadata and Resource Description”, 5 April 2001, viewed February, 28. 2007, <<http://www.w3.org/Metadata>>
- [7] Tim Berners-Lee, 05 December 2006, viewed February, 28. 2007, <<http://www.w3.org/People/Berners-Lee>>
- [8] Reunion, viewed April 2007, <<http://www.reunion.com>>
- [9] Sebastian Ryszard Kruck, Mariusz Cygan, Piotr Piotrowski, Krystian Samp, Adam Westerski and Stefan Decker, “Building a Heterogeneous Network of Digital Libraries on Semantic Web”, In Proceedings of Semantic Systems From Visions to Applications 2006, Vienna Austria 2006
- [10] Google, viewed February 2007, <<http://www.google.com>>
- [11] Magnus Niemann, Malgorzata Mochol and Robert Tolksdorf, “Improving Online Hotel Search – What Do We Need Semantic For”, In Proceedings of Semantic Systems from Visions to Applications 2006, Vienna Austria 2006
- [12] Ora Lassila and Ralph R. Swick, “Resource Description Framework (RDF) Model and Syntax Specification”, viewed February 2007, <<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>>
- [13] Edd Dumbill, “The Semantic Web: A Primer”, November 2000, Viewed February 2007, <<http://www.xml.com/pub/a/2000/11/01/semanticweb/index.html>>

- [14] B. Omelayenko., “Learning of ontologies for the Web: the analysis of existent approaches”, In Proceedings of the International Workshop on Web Dynamics, held in conj. with the 8th International Conference on Database Theory (ICDT’01), London, UK, 3 January 2001
- [15] D. Fensel. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2000
- [16] The W3C Extensible Markup Language (XML), viewed February 2007, <<http://www.w3.org/XML>>
- [17] OWL Web Ontology Language, viewed February 2007, <<http://www.w3.org/TR/owl-features>>
- [18] DAML, viewed February 2007, <<http://www.daml.org>>
- [19] Welcome to OIL, viewed February 2007, <<http://www.ontoknowledge.org/oil>>
- [20] Holger Knublauch, Mark A. Musen, Natasha F. Noy, "Tutorial: Creating Semantic Web (OWL) Ontologies with Protégé", 2nd International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, USA, October 20-23th, 2003
- [21] What Is An RDF Triple? , viewed February, 28. 2007, <[http://www.robertprice.co.uk/robblog/archive/2004/10/What\\_Is\\_An\\_RDF\\_Triple\\_.shtml](http://www.robertprice.co.uk/robblog/archive/2004/10/What_Is_An_RDF_Triple_.shtml)>
- [22] Mark Siebert, Pierre Smits, Leo Sauermann and Andreas Dengel, “Increasing Search Quality with the Semantic Desktop in Proposal Development”, In the proceedings of Practical Aspects of Knowledge Management 6th International Conference PAKM 2006, Vienna Austria 2006
- [23] Livelink, viewed February, 28. 2007, <<http://www.greggriffiths.org/livelink>>
- [24] Jena – A Semantic Web Framework for Java, viewed February, 28. 2007, <<http://jena.sourceforge.net>>
- [25] Guizhen Yang and Michael Kifer, “Reasoning about Anonymous Resources and Meta Statements on the Semantic Web”, In Journal on Data Semantics (JoDS), Volume I, pp. 69-97, 2003
- [26] Markus Linder, Martin Schliefnig, Dieter Fensel, Schahram Dustdar, Heinrich Otruba and Tassilo Pellegrini, “The realization of Semantic Web based E-Commerce and its impact on Business, Consumers and the Economy”, In Proceedings of Semantic Systems From Visions to Applications 2006, Vienna Austria 2006
- [27] Sam Joseph, “P2P MetaData Search Layers”, In the proceedings of Agents and Peer-to-Peer Computing, Second International Workshop, AP2PC 2003, Melbourne Australia 2003

- [28] David Huynh, Stefano Mazzocchi, and David Karger, “Piggy Bank: Experience the Semantic Web Inside Your Web Browser”, Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 413 – 430, 2005.
- [29] Gerhard Weikum, Jens Graupmann, Ralf Schenkel, and Martin Theobald, Towards a Statistically Semantic Web, P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 3–17, 2004. Springer-Verlag Berlin Heidelberg 2004
- [30] Grit Denker, Son Nguyen, and Andrew Ton, “OWL-S Semantics of Security Web Services: a Case Study”, J. Davies et al. (Eds.): ESWS 2004, LNCS 3053, pp. 240–253, 2004. Springer-Verlag Berlin Heidelberg 2004
- [31] Witold Abramowicz, Tomasz Kaczmarek, and Krzysztof Węcel, “How Much Intelligence in the SemanticWeb?”, P.S. Szczepaniak et al. (Eds.): AWIC 2005, LNAI 3528, pp. 1–6, 2005. Springer-Verlag Berlin Heidelberg 2005
- [32] Heiner Stuckenschmidt, “Approximate Information Filtering on the Semantic Web”, M. Jarke et al. (Eds.): KI 2002, LNAI 2479, pp. 114–128, 2002. Springer-Verlag Berlin Heidelberg 2002
- [33] D. Fensel. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2000
- [34] Okkyung Choi, SeokHyun Yoon, Myeongeun Oh, and Sangyong Han, “Semantic Web Search Model for Information Retrieval of the Semantic Data”, C.-W. Chung et al. (Eds.): HSI 2003, LNCS 2713, pp. 588-593, 2003., Springer-Verlag Berlin Heidelberg 2003
- [35] Jos´e Saias and Paulo Quaresma, "A Methodology to Create Legal Ontologies in a Logic Programming Information Retrieval System", V.R. Benjamins et al. (Eds.): Law and the Semantic Web, 3369, pp. 185–200, 2005. Springer-Verlag Berlin Heidelberg 2005
- [36] Amalia Todirascu, Laurent Romary, and Dalila Bekhouche, "Vulcain – An Ontology-Based Information Extraction System" B. Andersson et al. (Eds.): NLDB 2002, LNCS 2553, pp. 64–75, 2002. Springer-Verlag Berlin Heidelberg 2002
- [37] Marc Ehrig and York Sure, "Ontology Mapping - An Integrated Approach", J. Davies et al. (Eds.): ESWS 2004, LNCS 3053, pp. 76–91, 2004. Springer-Verlag Berlin Heidelberg 2004
- [38] V.R. Benjamins, J. Contreras, M. Blázquez, J.M. Dodero, A. Garcia, E. Navas, F. Hernandez and C. Wert, “Cultural Heritage and the Semantic Web”, J. Davies et al. (Eds.): ESWS 2004, LNCS 3053, pp. 433-444, 2004, Springer-Verlag Berlin Heidelberg 2004
- [39] Judy Brown, Stuart Marshall: "Sharing Human-Computer Interaction and Software Engineering Design Artifacts", In the Proceedings of Computer Human Interaction Conference, 1998, ISBN: 0-8186-9206-5, SA, Australia

[40] Sidney L. Smith and Jane N. Mosier "GUIDELINES FOR DESIGNING USER INTERFACE SOFTWARE", ESD -TR -86-278, The MITRE Corporation Bedford ,August 1986 , Massachusetts USA

[41]Klemmer, S. R and Lee, B: "Notebooks that Share and Walls that Remember: Electronic Capture of Design Education Artifacts". In Conference Supplement to UIST, 2005: ACM Symposium on User Interface Software and Technology. October 23-26, 2005, Seattle, WA

[42] Niels Olof Bouvin, "Human Computer Interaction", last modified: 19-Jan-2004 16:01:49 CET, reviewed 02 April 2008, < [www.daimi.au.dk](http://www.daimi.au.dk) >

[43] Jenifer Tidwell, "COMMON GROUND: A Pattern Language for Human-Computer Interface Design", Reviewed 02 April 2008, <[http://www.mit.edu/~jtidwell/common\\_ground\\_onefile.html](http://www.mit.edu/~jtidwell/common_ground_onefile.html)>

[44] Mark Grand, Preview of Patterns in Java Volume 2, reviewed at 04 April 2008, <[http://www.mindspring.com/~mgrand/pattern\\_synopses2.htm](http://www.mindspring.com/~mgrand/pattern_synopses2.htm)>

[45] Kent Beck, Reviewed at 02 April 2008, <<http://c2.com/ppr/about/author/kent.html>>

[46] Tom Erickson, "The Interaction Design Patterns Page", reviewed 02 April 2008, <<http://www.visi.com/~snowfall/InteractionPatterns.html>>

[47] Thomas T. Hewett, "ACM SIGCHI curricula for human-computer interaction", Technical Report, Pages: 162,1992, ISBN:0-89791-474-0, ACM New York, USA

[48] Rich Internet Application, reviewed 19 May 2008, <[http://en.wikipedia.org/wiki/Rich\\_Internet\\_application](http://en.wikipedia.org/wiki/Rich_Internet_application)>

[49] FLEX, reviewed 19 May 2008, <<http://www.adobe.com/products/flex/>>

[50] Adobe Flex, reviewed 19 May 2008, <[http://en.wikipedia.org/wiki/Adobe\\_Flex](http://en.wikipedia.org/wiki/Adobe_Flex)>

[51] Ajax, reviewed 19 May 2008, <<http://www.asp.net/ajax>>

[52] Ajax, reviewed 19 May 2008, <<http://www.w3schools.com/Ajax/Default.Asp>>

[53] Ajax, reviewed 19 May 2008, <<http://en.wikipedia.org/wiki/AJAX>>

[54] Ajax, reviewed 19 May 2008, <[http://en.wikipedia.org/wiki/Rich\\_Internet\\_application#JavaScript\\_2F\\_Ajax](http://en.wikipedia.org/wiki/Rich_Internet_application#JavaScript_2F_Ajax)>

[55] OpenLaszlo, reviewed 19 May 2008, <http://en.wikipedia.org/wiki/OpenLaszlo>

[56] Rich\_Internet\_application, reviewed 19 May 2008, [http://en.wikipedia.org/wiki/Rich\\_Internet\\_application#JavaScript\\_2F\\_Ajax](http://en.wikipedia.org/wiki/Rich_Internet_application#JavaScript_2F_Ajax)

- [57] What is OpenLaszlo, and what's it good for? reviewed 19 May 2008, <http://www.donhopkins.com/home/124^>
- [58] Silverlight, reviewed 19 May 2008, <http://en.wikipedia.org/wiki/Silverlight>
- [59] Product Data Management (PDM), reviewed May 2007, < <http://www.johnstark.com> >
- [60] Product Data Management, Reviewed April 2007, <[http://en.wikipedia.org/wiki/Product\\_Data\\_Management](http://en.wikipedia.org/wiki/Product_Data_Management)>
- [61] C. S. Sung, Sam Joon Park, "A component-based product data management system", In Springer-Verlag London Limited 2006, Received: 18 April 2005 / Accepted: 25 October 2005 / Published online: 2006
- [62] Yi Zhuang Yueting Zhauang, Qing Li and Lei Chen, "Towards Interactive Indexing for Large Chinese Calligraphic Character Database", In the proceedings of ACM Fifteenth Conference on Information and Knowledge Management (CIKM 06), November 5-11 2006, Virginia USA
- [63] C. S. Sung, Sam Joon Park, "A component-based product data management system", In Springer-Verlag London Limited 2006, Received: 18 April 2005 / Accepted: 25 October 2005 / Published online: 2006
- [64] M.Y. Huang, Y.J. Lin, Hu Xu, "A framework for web-based product data management using J2EE", In Springer-Verlag London Limited, Received: 8 October 2002 / Accepted: 13 March 2003 / Published online: 27 October 2004
- [65] Data Warehouse, Reviewed 06 August 2008, <[http://en.wikipedia.org/wiki/Data\\_warehouse](http://en.wikipedia.org/wiki/Data_warehouse)>
- [66] Laura Hadley, "Developing a Data Warehouse Architecture", Reviewed 06 August 2008, <<http://www.users.qwest.net/~lauramh/resume/thorn.htm>>
- [67] Knowledge base, Reviewed 07 August 2008, <[http://en.wikipedia.org/wiki/Knowledge\\_base](http://en.wikipedia.org/wiki/Knowledge_base)>
- [68] Java, Reviewed 01 September 2008, <<http://www.java.com/en/>>
- [69] Language technology, Reviewed 03 September 2008, <[http://en.wikipedia.org/wiki/Human\\_language\\_technology](http://en.wikipedia.org/wiki/Human_language_technology)>
- [70] Hans Uszkoreit, DFKI-LT - What is Language Technology?, Reviewed 03 September 2008, <<http://www.dfki.de/lt/lt-general.php>>
- [71] TWENTYONE: Disclosure and dissemination of documents on sustainable development, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?ltc/twentyone-e>>

- [72] MULINEX: Multilingual Indexing, Navigation and Editing Extensions for the WWW, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/mulinex-e>>
- [73] MIETTA: Multilingual Tourist Information on the World Wide Web, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/mietta-e>>
- [74] OLIVE: Play it again, Sam! Retrieval of video material based on speech-recognition, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/olive-e>>
- [75] PARADIME: Intelligent Extraction of Information from On-line Documents, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/paradime-e>>
- [76] Whiteboard: Multilevel Annotation for Dynamic Free Text Processing, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/whiteboard-e>>
- [77] GETESS: German Text Exploitation and Search System, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/getess-e>>
- [78] TG/2: Practical generation of natural language text, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/tg2-e>>
- [79] TEMSIS: Transnational Environmental Management Support & Information System, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/temsis-e>>
- [80] COSMA: Automated Appointment Scheduling by E-Mail, Reviewed 03 September 2008, <<http://www.dfki.de/pas/f2w.cgi?lhc/cosma-e>>
- [81] K2 Mobility Sustainable Vehicle Technologies (K2 Mobility SVT), Reviewed 08 September 2008, <<http://www.vif.tugraz.at/index-spot.php?lang=2&id=62&beam=1>>
- [82] VIVACE, Reviewed 05 September 2008 <<http://www.vivaceproject.com/content/introduction/intro.php>>
- [83] VIVACE Overview, Reviewed 05 September 2008, <<http://www.vivaceproject.com/generalpresentation.pdf>>
- [84] Promise-PLM, PROMISE - PRODUCT LIFECYCLE MANAGEMENT AND INFORMATION TRACKING USING SMART EMBEDDED SYSTEMS, Reviewed 04 September 2008, <[http://www.promise-plm.com/tmp\\_promise\\_site\\_11.asp?lang=3&sua=1&s=10](http://www.promise-plm.com/tmp_promise_site_11.asp?lang=3&sua=1&s=10)>
- [85] Windchill, Reviewed 06 November 2008 <[http://www.ptc.com/WCMS/files/56909/en/2757\\_Windchill\\_bro\\_ViewONLY.pdf](http://www.ptc.com/WCMS/files/56909/en/2757_Windchill_bro_ViewONLY.pdf)>
- [86] CIM DATABASE 2.9.5, User Manual by CONTACT Software GmbH

[87] Nihit Aurora, "Careers in Computer Integrated Manufacturing (CIM)", Reviewed 12 January 2009, <<http://pdf.employmentcrossing.com/manufacturing/340023.pdf>>

[88] Product lifecycle management, Reviewed at 12 January 2009, <[http://en.wikipedia.org/wiki/Product\\_Lifecycle\\_Management](http://en.wikipedia.org/wiki/Product_Lifecycle_Management)>

[89] François Guimbretière, Maureen Stone, Terry Winograd, "Fluid Interaction with High-resolution Wall-size Displays", Proc. ACM Symposium on User Interface Software and Technology, 2001

## 10 Related Publications

- **Zeeshan Ahmed**: "*Proposing Semantic Oriented Agent and Knowledge base Product Data Management*", At Emerald: Information Management and Computer Security Journal, Manuscript ID IMCS-02-2009-0004, by Emerald – Scholar One Manuscript Central.
- **Zeeshan Ahmed**: "PDM Problem and Solutions", At IPROMS, the 5th Virtual Conference, of the EU-funded FP6 I\*PROMS Network of Excellence on Innovative Production Machines and Systems, 2009
- **Zeeshan Ahmed**: "I-SOAS Desktop Tool Demo", At IEEE The Second International Conference on the Applications of Digital Information and Web Technologies, IEEE-ICADWIT, London Metropolitan University, 2009
- **Zeeshan Ahmed**: "Semantic Oriented Intelligent and Flexible Graphical Machine Interface", At IEEE The Second International Conference on the Applications of Digital Information and Web Technologies, IEEE- ICADWIT, 2009
- **Zeeshan Ahmed**: "Product Data Management and I-SOAS Implementation Designs", At Third International Conference on Information & Communication Technologies, IEEE - ICICT2009, 2009
- **Zeeshan Ahmed**, Detlef Gerhard, "Design Implementation of I-SOAS IPM for Advanced Product Data Management", In the proceedings of The Second IEEE International Conference On Computer, Control & Communication (IEEE-IC4 2009), IEEE Catalog Number: CFP0963F; ISBN: 978-1-4244-3314-8; Library of Congress: 2008909474, 2009
- **Zeeshan Ahmed**: "PDM based I-SOAS Data Warehouse Design", In the proceedings of FIFTH International Conference on Statistical Sciences: Mathematics, Paper ID 125, ISBN 978-969-8858-04-9, Vol. 17, 2009
- **Zeeshan Ahmed**, Detlef Gerhard: "*Design Implementation of Semantic Oriented Agent and Knowledge based approach for Intelligent Human Machine Data Manipulation*", In the proceedings of 4th Virtual International Conference on Innovative Production Machines and Systems, IPROMS, 2008
- **Zeeshan Ahmed**, Detlef Gerhard: "*Semantic Oriented Intelligent Electronic Learning*", In the proceedings of IEEE - International conference on Information & Communication Technologies, IEEE - ICICT, 2008
- **Zeeshan Ahmed**, Detlef Gerhard: "*I-SOAS towards Product Data Management (PDM) based Application's Problems*", In the proceedings of IEEE Doctoral Symposium on Research in Computer Science. IEEE - DSCS, 2008
- **Zeeshan Ahmed**, Detlef Gerhard: "*Intelligent Graphical User Interface, Information Processing and Data Management based Solution towards Intelligent Data*



- Manipulation and Representation*", In online proceedings of 4th Virtual International Conference on Innovative Production Machines and Systems, IPROMS , 2008
- **Zeeshan Ahmed**, Detlef Gerhard: "*Information Engineering and Knowledge Management based approach towards Intelligent E-Learning Systems Development*", In online proceedings of 4th Virtual International Conference on Innovative Production Machines and Systems, IPROMS , 2008
  - **Zeeshan Ahmed**, Detlef Gerhard: "Semantic Oriented Agent based Approach towards Engineering Data Management, Web Information Retrieval and User System Communication Problems", In the proceedings of 3rd International Conference for Internet Technology and Secured Transactions, ICITST 08, IEEE - CST, 2008
  - **Zeeshan Ahmed**, Sudhir Kumar Ganti, Hans Kyhlbäck: "Design Artifact's, Design Principles, Problems, Goals and Importance ", In Proceedings 4th International Statistical Conference May 9-11, Paper ID 42, Vol. 15, 57-68, ISBN 978-969-8858-04-9, 2008
  - **Zeeshan Ahmed**, Detlef Gerhard: "Contributions of PDM Systems in Organizational Technical Data Management", In the proceedings of The First IEEE International Conference On Computer, Control & Communication, IEEE-IC4, 2007
  - **Zeeshan Ahmed**, Detlef Gerhard: "An Agent based Approach towards Metadata Extraction, Modelling and Information Retrieval over the Web", In the proceedings of First International Workshop on Cultural Heritage on the Semantic Web in conjunction with the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference 2007, ISWC + ASWC, P 117, 2007
  - **Zeeshan Ahmed**, Detlef Gerhard: "Role of Ontology in Semantic Web Development", In the proceedings of First International Workshop on Cultural Heritage on the Semantic Web in conjunction with the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference 2007, ISWC + ASWC , P 119, 2007
  - **Zeeshan Ahmed**, Detlef Gerhard: "Designing a Dynamic Components and Agent based Approach for Semantic Information Retrieval", In the proceedings of 6<sup>th</sup> CIIT Workshop on Research in Computing , P 8, 2007
  - **Zeeshan Ahmed**, Detlef Gerhard: "How does Ontology Contribute in Semantic Web Development?", In the proceedings of 6<sup>th</sup> CIIT Workshop on Research in Computing (CWRC Fall 2007), P 3, 2007
  - **Zeeshan Ahmed**, Detlef Gerhard: "Web to Semantic Web & Role of Ontology", In the proceedings of National Conference on Information and Communication Technologies, NCICT 2007, P 100, 2007
  - **Zeeshan Ahmed**, Detlef Gerhard: "Personal Assistant towards Semantic Information Retrieval", In the proceedings of Fifth International Workshop on Ontologies and Semantic Web for E-Learning SWEL'07, 13th International Conference on Artificial Intelligence in Education AIED 2007, P115, 2007

## 10.1 Related BS & MS Thesis

### 1. Topic: Semantic Web Integration Approaches for Product Data Management (PDM)

Student: Mr. Vasil Popov

Program: MSc Computer Science

University: Technical University-Sofia, Bulgaria

### 2. Topic: Ontological, Warehoused and Knowledge Management Based Product Data Management

Student: Mr. Faraz Omer

Program: MSc Computer Science

University: Stockholm University – Stockholm, Sweden

### 3. Topic: High IT Failure Rate: A Management Prospect

Student: Mazhar Tajammal Hashmi

Program: MSc Business Administration

University: Blekinge University – Ronneby, Sweden

### 4. Topic: Information Processing and Semantic Modelling towards Product Data Management

Student: Ms. Ina Tacheva

Program: BSc Computer Science

University: Technical University-Sofia, Bulgaria

# 11 Appendix

## 11.1 Grammar Used in I-SOAS

```

grammar grammarISOAS;

options {
    language=Java;
}

tokens {
    PLUS                = '+';
    MINUS               = '-';
    MULT                = '*';
    DIV                 = '/';
}

/*
 *      I-SAOS Parser Rules
 */

astmt    : A;
bstmt    : B;
cstmt    : C;
stmt1    : A B C;
stmt2    : B C;

/*
 *      I-SAOS Lexer Rules
 */

fragment DIGIT          : '0'..'9';
NUMBER                  : (DIGIT)+;

A      :          T'We'T'They'T'He'T'She';

B      :          'need''want''look for''look about''search''ask for''seek for''needs''wants''looks for''looks about''seraches''asks for''seeks for''am looking for''am searching''am asking for''am seeking''am in search of''are looking for''are searching''are asking for''are seeking''are in search of''is looking for''is searching''is asking for''is seeking''is in search of''define''quest''questing''identify''scratch around''root about''find''after';

```

*C* :  
 ('0..'9'|'a'..'z'|'A'..'Z'|'PDM'|'CAD'|'document'|'printer'|'presentation'|'application'|'co  
 ntract'|'office'|'section'|'quarter'|'airport'|'boulevard'|'street'|'country'|'sity'|'town'|'shop'|'bus  
 stop'|'hotel'|'hostel'|'theater'|'cinema'|'movies'|'picture'|'film'|'song'|'singer'|'music'|'lyrics'|'radi  
 o'|'group'|'game'|'news'|'job'|'train  
 station'|'torrents'|'subtitles'|'gifts'|'clothes'|'shoes'|'dress'|'clothing'|'banks'|'weather'|'books'|'m  
 agazines'|'newspaper'|'publications'|'articles'|'events'|'concerts');

*PLUS* : '+';

*MINUS* : '-';

*MULT* : '\*';

*DIV* : '/';

*WS*: (' |\r|\t|\u00C|\n') {\$channel=HIDDEN;} ;