A SURVEY OF ARTIFICIAL INTELLIGENCE TECHNIQUES FOR CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ÇANKAYA UNIVERSITY

BY

CEMALETTİN ÖCAL FİDANBOY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2009

Title of the Thesis : **A Survey of Artificial Intelligence Techniques for Capability Maturity Model Integration (CMMI)**

Submitted by **Cemalettin Öcal Fidanboy**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Yahya K. Baykal
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Mehmet Reşit Tolun
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Mehmet Reşit Tolun
Supervisor

Examination Date 03.09.2009

Examining Committee Members

Prof. Dr. Mehmet Reşit Tolun (Çankaya Univ.)

Prof. Dr. Ziya Aktaş (Başkent Univ.)

Asst. Prof. Dr. Abdülkadir Görür (Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Cemalettin Öcal Fidanboy

Signature :

Date : 03.09.2009

ABSTRACT

A SURVEY OF ARTIFICIAL INTELLIGENCE TECHNIQUES
FOR CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

Fidanboy, Cemalettin Öcal
M.S.c., Department of Computer Engineering
Supervisor : Prof. Dr. Mehmet Reşit Tolun

September 2009, 83 pages

Our purpose in this thesis is to investigate the current artificial intelligence applications in scope of the CMMI process areas. Firstly, research is made regarding the CMMI Model. Then, current studies about CMMI process areas by using artificial intelligence techniques were examined.

The overall aim of the thesis is to perform a survey about artificial intelligence techniques conjunction with Capability Maturity Model Integration (CMMI) process areas. As a result, future work evaluation in artificial intelligence applications with CMMI was performed.

Keywords: CMMI, Artificial Intelligence, Software Quality, Process Improvement

ÖZ

BÜTÜNLEŞİK YETENEK OLGUNLUK MODELİ (CMMI) İÇİN
YAPAY ZEKA TEKNİKLERİNİN İNCELENMESİ

Fidanboy, Cemalettin Öcal
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi : Prof. Dr. Mehmet Reşit Tolun

Eylül 2009, 83 sayfa

Bu tezdeki amacımız, CMMI süreç alanları kapsamında yapılan, güncel yapay zeka uygulamalarını araştırmaktır. Öncelikle, CMMI Modeli ile ilgili araştırma yapılmıştır. Sonra, yapay zeka yöntemleri kullanılarak yapılan CMMI süreç alanları ile ilgili güncel çalışmalar incelenmiştir.

Tezin genel amacı, CMMI süreç alanları kapsamında yapılan yapay zeka uygulamaları hakkında bir araştırma yapmaktır. Sonuç olarak, CMMI ile yapay zeka uygulamaları hakkında gelecekte yapılabilecek araştırmalar için bir değerlendirme yapılmıştır.

Anahtar Kelimeler: CMMI, Yapay Zeka, Yazılım Kalitesi, Süreç İyileştirme

ACKNOWLEDGMENTS

I first thank to my thesis supervisor Prof. Dr. Mehmet Reşit TOLUN for his guidance, advice, criticism, encouragements and insight throughout the research.

I would also like to thank to Dr. Kıvanç DİNÇER for his suggestions, comments and encouragement during the writing of this thesis.

Finally, I would like to thank to my wife and my daughter for their love and morale support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CAR	Causal Analysis and Resolution
CI	Computational Intelligence
CM	Configuration Management
CMMI	Capability Maturity Model Integration
CMMI-ACQ	CMMI for Acquisition
CMMI-DEV	CMMI for Development
CMMI-SVC	CMMI for Services
DAR	Decision Analysis and Resolution
EPWPS	Evaluating Processes and Work Products Service
GG	Generic Goal
GP	Generic Practice
IPM	Integrated Project Management
IPPD	Integrated Product and Process Development
KMMM	Knowledge Management Maturity Model
MA	Measurement and Analysis
MNS	Message Notification Service
OIDSA	Ontology-based Intelligent Decision Support Agent
OPD	Organizational Process Definition
OPF	Organizational Process Focus
OPP	Organizational Process Performance
ORA	Ontological Reasoning Agent
OT	Organizational Training
OTK	On to Knowledge
PA	Process Area
PI	Product Integration
PMC	Project Monitoring and Control
POIS	Providing Objective Insight Service

POS	Part-of-Speech
PP	Project Planning
PPQA	Product and Process Quality Assurance
QPM	Quantitative Project Management
RD	Requirement Development
RSKM	Risk Management
SAM	Supplier Agreement Management
SCAMPI	The Standard CMMI Appraisal Method for Process Improvement
SD	Semantic Distance
SE	Software Engineering
SG	Specific Goal
SP	Specific Practice
TRS	Term Relation Strength
TS	Technical Solution
TW	Term Word
VAL	Validation
VER	Verification

CHAPTER 1

INTRODUCTION

Today, information technologies are progressing rapidly throughout the world. The number of software development projects is increasing day by day. Unfortunately, the majority of software projects can not attain the desired success. The basic reason for this failure is that the necessary importance is not given to software quality models and standards. However, to achieve success in software development, one must act according to the defined processes. Software development is really a complex job. Software development is not about only the code generation. During software development, at the same time, one should apply many processes (project management, process management, support and technical processes, etc.) as parallel.

To ensure higher customer satisfaction, we have to achieve quality in software. Software quality development is one of the most fundamental ways to produce reliable products and services. "Successful software depends not only on technical experience but on how to apply our software processes and how to perform systematic process improvement activities. So, if we want to achieve better software quality, we need some guidelines for process integration and product improvement activities. There are many ways in the literature to improve software processes. CMMI (Capability Maturity Model Integration) is the popular model to improve our processes. CMMI is a model that provides guidance to organizations on the better development of software products and related services. The CMMI Product Suite contains an enormous amount of information and guidance to help an organization improve its processes [1]."

W. Edwards Deming (1982) described the "quality chain reaction": When product and service quality improves, costs decrease because there is less rework and fewer delays. Because costs have decreased, productivity is improved. Higher productivity lets a company capture a bigger piece of the market with lower price and higher quality. Greater market share leads to more business and higher employment. Thus, a focus on quality leads to increased business success. Process definition and process improvement enables this desired improvement in product and service quality [2].

The quality of software products depends on the capability and maturity of their processes. Software process improvement activities have a strong impact on lower development and maintenance costs, increased predictability and controllability of software products and processes, and shorter time to market [3].

This thesis focuses on basic principles of CMMI and explains the relationship between artificial intelligence techniques. The CMMI model has been investigated. At the same time, the artificial intelligence work within the scope of CMMI in the literature was examined.

1.1. The Purpose and Scope of the Study

This study aims to investigate AI methods for CMMI-DEV v.1.2. Some CMMI processes and artificial intelligence techniques were examined. In this thesis, As a result of literature research, we saw that some CMMI processes have been successfully applied using AI techniques.

These CMMI Process Areas are presented below:

- Product and Process Quality Assurance (PPQA)
- Project Monitoring and Control (PMC)
- Project Planning (PP)

- Configuration Management (CM)

- Measurement and Analysis (MA)

The overall aim of the thesis is to investigate how to increase the effectiveness of process improvement works by using artificial intelligence applications.

AI techniques used in this context has demonstrated some similarities. So, we can use similar techniques for other CMMI Process Areas. The study describes the structure of CMMI and artificial intelligence research in these targets. Basic aim is to assist the work to be done in the future.

1.2. Roadmap

In this study, the basic goal is to explain the concept of CMMI. First, a literature survey was done for CMMI and some artificial intelligence techniques. Then, on the basis of CMMI, the studies using artificial intelligence applications have been investigated. As a result, future work evaluation in artificial intelligence applications with CMMI was performed.

1.3. Organization of the Thesis

Chapter 2 provides detailed information about CMMI-DEV v.1.2 Model. History of CMMI, CMMI for development, CMMI representations, CMMI capability and maturity levels and CMMI process areas are described from a process improvement view point. On the other hand, detail descriptions were described for some CMMI processes (Process and Product Quality Assurance, Project Monitoring and Control, Project Planning, Configuration Management and Measurement and Analysis).

In Chapter 3, the relationship between software engineering and artificial intelligence is described. This section provides information about AI techniques which are used for some CMMI process areas. These techniques are explained from the CMMI point of view. Then, by using artificial intelligence techniques in the scope of CMMI process areas of current studies are examined. Finally, Chapter 4 presents the summary and conclusion of the study.

CHAPTER 2

CAPABILITY MATURITY MODEL INTEGRATION

2.1 Capability Maturity Model Integration (CMMI)

"CMMI (Capability Maturity Model Integration) is a process improvement maturity model for the development of products and services. It consists of best practices that address development and maintenance activities that cover the product lifecycle from conception through delivery and maintenance [4]." It is used to plan, define, implement, deployment, benchmark and improve processes in an organization. The product suite, defines the levels through which organizations evolve as they improve their processes. This product suit contains the models, appraisal methods and trainings used to support improvements. It gives an opportunity for process improvement priorities and goals [2].

The CMMI focuses on the managerial, technical and support aspects of software development. It's a management model used to gain insight into and control over the development. It is a model for improving the capability maturity of development processes. CMMI was developed by the Software Engineering Institute (SEI) of Carnegie Mellon University. It assesses and evaluated the organizational maturity at five levels [4].

Over the past years, CMMI have been broadly used for assessing organizational maturity and process capability throughout the world. Many organizations are used to CMMI appraisals. They have trust in CMMI because of its comprehensive descriptions of how the various good practices fit together [5].

CMMI supports process integration and product improvement. It enables the integration of multiple disciplines into one process-improvement framework. It provides a framework for introducing new disciplines as needs arise and therefore it reduces the cost of implementing model-based improvement. It is designed to build on legacy process improvement efforts and investments [6].

CMMI is not a process. It is a model representing an ideal state that is used as a benchmark. CMMI describes best practices but it doesn't contain any knowledge about how to implement those practices. If you want to achieve better process quality, you have to interpret the model in respect of your necessities. CMMI describes what is expected in processes and what is expected in the processes depend on organization itself [2].

CMMI is used in companies of the most diverse industry sectors all over the world. It is used to analyze the strengths and improvement opportunities of an organization. Also it is used to systematic improvements. CMMI can provide a path for an organization to achieve its performance goals.

CMMI work has started approximately 18 years ago and it became a most popular process improvement and quality management model in information technology world. It is being adopted as the preferred information technology quality standard by more and more organizations. [7]. CMMI can be tailored to the needs of the organization. It can be used by many organizations for process improvement.

CMMI is a process model which describes best practices in a systematic way. It improves the capability and maturity levels of an organization in three constellations:

- Development of Product and Services - CMMI for Development (CMMI-DEV)

- Service delivery – CMMI for Services (CMMI-SVC – in pilot phase)

- Supplier management – CMMI for Acquisition (CMMI-ACQ)

A constellation is defined as "a grouping of model components that are unique to a specific use but also contain a core set of process areas that will not change from constellation to constellation." These constellations set the stage for potential future CMMI expansion [2].

The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) is an appraisal method to appraise an organization. SCAMPI appraisals can help those organizations identify the strengths and weaknesses of their current processes [8].

2.2 History of CMMI

The CMMI Product Suite was developed by the CMMI Product Team to improve on the existing Software Capability Maturity Model (SW-CMM) released in 1991. In 2000, the team published the original CMMI model. The model was also designed to support the integration of other disciplines. In August 2006, CMMI Product Suite Version 1.2 was introduced with the release of the CMMI for Development, CMMI-DEV v1.2 model. The CMMI for Acquisition constellation was released on 2007[9].

- CMMI for Development (CMMI-DEV), v1.2 was released in August 2006.

- CMMI for Acquisition (CMMI-ACQ), v1.2 was released in November 2007.

- CMMI for Services (CMMI-SVC), v1.2 was released in February 2009.

- CMMI Product Suite (includes Development, Acquisition, and Services), v1.3 is expected to be released in 2010 [10].

2.3 CMMI for Development

"CMMI for Development is a reference model that covers the development and maintenance activities applied to both products and services. Organizations from many industries, including aerospace, banking, computer hardware, software, defense, automobile manufacturing, and telecommunications use CMMI for Development [4]."

CMMI-DEV v.1.2 is the current version of the Capability Maturity Model Integration. It contains generic approaches for software engineering, systems engineering and integrated process and product development together into a single framework. It allows greater flexibility in reaching the model's goals. The "DEV" stands for Development. The CMMI-DEV (version 1.2) is the model useful for organizations that have to develop products, processes, or both for their customers [7].

CMMI-DEV v.1.2 includes practices components for the following Information Technology disciplines:

- Systems engineering
- Software engineering
- Integrated product and process development.

CMMI-DEV v.1.2 is a model specifically designed for Information Technology sector. It can be used for non-IT sectors, but main aim is to help IT organizations develop highly reliable management, engineering, and quality-support systems [7].

2.4 CMMI Model Structure

The CMMI is structured as follows [11]:

- Maturity Levels (staged representation) or Capability Levels (continuous representation)

- Process Areas
- Goals—Generic and Specific
- Practices—Generic and Specific

CMMI contains two representations. These are "Continuous Representation" and "Staged Representation". It depends on the organization's goal which representation is chosen. If organizations want to measure the maturity of their whole processes, they will select the staged representation. If they want to determine process capability in order to improve processes, they will choose the continuous representation.

They have some differences for selection, organization and presentation of model components but contain same practices. This can be defined as being similar to different views into a data set in a database. In the continuous representation, process areas are organized in categories and in the staged representation they are organized by maturity level.

Process areas (PAs) contain some best practices. A PA is defined as "a cluster of related best practices that when implemented collectively satisfies a set of goals considered important for making significant improvement in that area" [2]. Process area structure is presented in Figure 1.

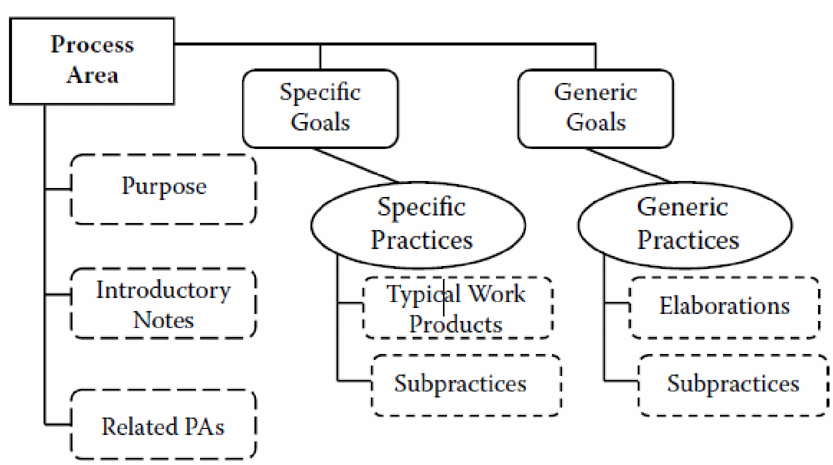

Figure 1: Process Area Structure [2]

2.4.1 Model Structure of Continuous Representation

The continuous representation has specific goals that organize specific practices and generic goals that organize generic practices. Each specific and generic practice corresponds to a capability level. Specific goals and specific practices apply to individual process areas [11].

Figure 2 shows the CMMI model components in the continuous representation.

Continuous models have process areas that contain practices. The practices of a process area in a continuous model are organized in a manner that supports individual process area growth and improvement. The generic practices are grouped into capability levels. Process areas are institutionalized by implementing the generic practices in those process areas. Total capability levels of all process areas show organizational improvement [1].

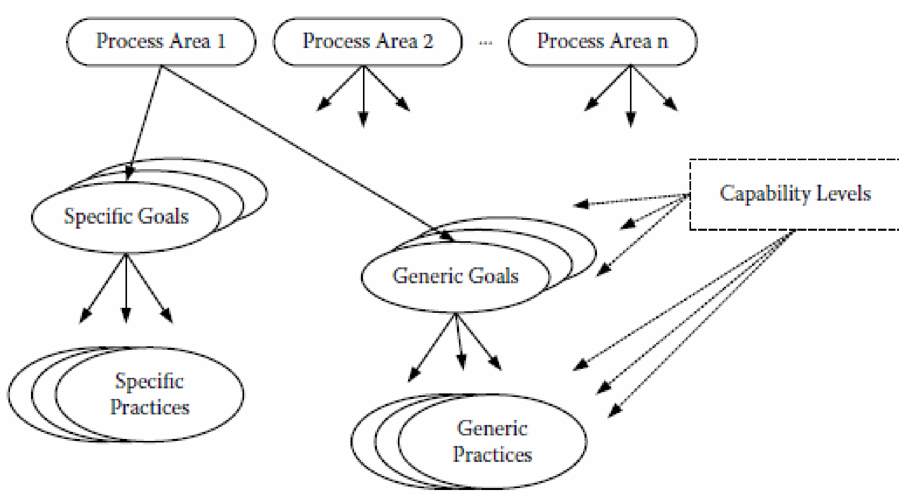

Figure 2: CMMI Model Components in Continuous Representation [11]

The continuous representation and staged representation use the same basic structure. However, continuous representation contains four process categories. (Process Management, Project Management, Engineering and Support)

Table 1 shows the continuous grouping of process areas:

Table 1: Continuous Groupings of CMMI Process Areas [1]

Continuous Grouping	**Acronyms**	**Process Areas**
Process Management	OPF	Organizational Process Focus
	OPD+IPPD	Organizational Process Definition + IPPD
	OT	Organizational Training
	OPP	Organizational Process Performance
	OID	Organizational Innovation and Deployment
Project Management	PP	Project Planning
	PMC	Project Monitoring and Control
	SAM	Supplier Agreement Management
	IPM+IPPD	Integrated Project + IPPD Management
	RSKM	Risk Management
	QPM	Quantitative Project Management
Engineering	REQM	Requirements Management
	RD	Requirements Development
	TS	Technical Solution
	PI	Product Integration
	VER	Verification
	VAL	Validation
Support	CM	Configuration Management
	PPQA	Process and Product Quality Assurance
	MA	Measurement and Analysis
	DAR	Decision Analysis and Resolution
	CAR	Causal Analysis and Resolution

Process Management

These process areas contain main process management activities (defining, planning, implementing and monitoring a process etc.) These are organizational level process areas, not the project level [11].

There are five Process Management Process Areas:

- Organizational Process Focus
- Organizational Process Definition (with Integrated Product and Process Development—IPPD)
- Organizational Training
- Organizational Process Performance
- Organizational Innovation and Deployment [11].

These process areas address an organization's efforts to define, plan, deploy, implement, monitor, control, verify, measure, and improve its processes. Organization Process Performance builds on the capabilities in Organizational Process Focus and Organizational Process Definition. Organizational Innovation and Deployment builds on the capabilities of the other Process Management process areas [1].

On the staged representation, Maturity Level 3 contains Organizational Process Focus, Organizational Process Definitions process areas and Organizational Training. The Organizational Training process area arranges organizational training activities. Organizational Process Performance Process Area is located in Maturity Level 4 and Organizational Innovation and Deployment Process Areas is located in Maturity Level 5. When using the CMMI continuous representation for process improvement, we should consider these relationships [1].

Process areas in the process management category are organizational level process areas. These process areas deal with the process improvement activities. These process areas establish a process improvement approach to an organization. These process areas provides guidelines for establishing the process architecture, a process asset library, a measurement database and address some practices for establishing an organizational training capability [2].

Project Management

These process areas contain main project management activities such as planning, tracking, and controlling projects [11].

There are six Project Management Process Areas:

- Project Planning
- Project Monitoring and Control
- Supplier Agreement Management
- Integrated Project Management (with Integrated Product and Process Development—IPPD)
- Risk Management
- Quantitative Project Management

There is a close relationship exists between four of these process areas, with Integrated Project Management and Quantitative Project Management building on the capabilities of Project Planning and Project Monitoring and Control. Project Planning and Project Monitoring and Control exist in Maturity Level 2. Integrated Project Management exists in Maturity Level 3 and Quantitative Project Management exists in Maturity Level 4. When using continuous representation, we should understand this relationship and plan our projects accordingly [1].

The basic project management processes are Project Planning (PP), Project Monitoring and Control (PMC), and Supplier Agreement Management (SAM). The advanced processes are Integrated Project Management (IPM), Risk Management (RSKM), and Quantitative Project Management (QPM). "Adding more rigor and better management techniques, such as sharing information among projects in the organization, managing stakeholders, and performing rigorous risk management to the basic management processes helps the organization achieve more effective management processes. It may be advantageous to consider the basic and advanced process areas together when planning and implementing process improvements [2]."

Engineering

These Process Areas consist of technical functions related to building and delivering a product. There are six Engineering Process Areas (11):

- Requirements Development
- Requirements Management
- Technical Solution
- Product Integration
- Verification
- Validation

In the Requirements Development Process Area, requirements are identified and documented and changes to those requirements are controlled through the Requirements Management Process Area. When requirements are sufficiently defined and stable enough, product architecture is developed in the Technical Solution Process Area. Requirements analysis and high-level design may continue iteratively until the design architecture is mature enough to proceed with detailed component design and implementation. When product components are implemented, they are integrated using processes described in the Product Integration Process Area. The Verification and Validation Process Areas are implemented throughout the life cycle [2].

Support

These Process Areas consist of related support functions related to managing change request, obtain quality assurance for projects, managing measurement and analysis activities and decision analysis and resolution activities.

There are five Support Process Areas:

- Configuration Management
- Process and Product Quality Assurance
- Measurement and Analysis
- Decision Analysis and Resolution
- Causal Analysis and Resolution

Support process areas support all processes and perform as parallel. They are performed throughout the project life cycle [11].

2.4.2 Model Structure of Staged Representation

The staged representation contains maturity levels. A maturity level defines the level of performance that can be expected from an organization.

For example, Maturity Level 1 organizations have no structured processes. Maturity Level 2 organizations have a basic project management processes. There are five maturity levels [11].

A staged model provides a predefined road map for organizational improvement works. It contains series of "stages" that are called “maturity levels." Each maturity level has a set of process areas. These process areas indicate where an organization should focus to improve its organizational process. Each process area is described in terms of the practices that contribute to satisfying its goals. The practices describe some activities related process areas. In a particular maturity level, all process areas should satisfy the specific and generic goals [1].

Figure 3 shows the CMMI model components in the staged representation.

A process area is the fundamental organizational feature of all the CMMI models. CMMI uses only the most important issues for process improvement and then groups those issues into "areas." Each constellation contains number of process areas [1].

Each maturity level contains various process areas. “A process area is a group of practices or activities performed collectively in order to achieve a specific objective”. For example, Process and Product Quality Assurance Process Area exists in the Maturity Level 2. In the same way, Organizational Process Focus Process Area exists in the Maturity Level 3 [11].

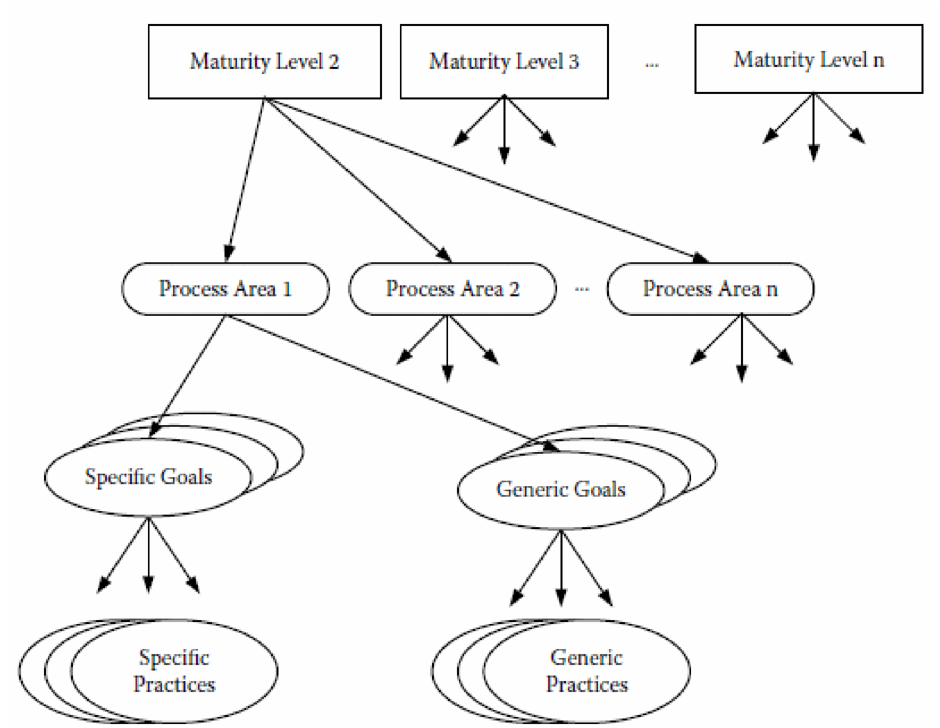

Figure 3: CMMI Model Components in Staged Representation [11]

Each Process Area has several goals that need to be satisfied in order to satisfy the objectives of the Process Area.

There are two types of goals:

Specific goals (SG): Goals that relate only to the specific Process Areas under study.

Generic goals (GG): Goals that are common to multiple Process Areas throughout the model.

Practices are activities that must be performed to satisfy the goals for each PA. Each practice relates to only one goal.

There are two types of practices:

Specific practices (SP): practices that relate to specific goals.

Generic practices (GP): practices associated with the generic goals for institutionalization.

For example, in the Project Monitoring and Control Process Area, one of the specific practices is to monitor project planning parameters. Another is to monitor project risks. These subpractices and examples help understanding the practices in more detail.

2.5 CMMI Capability and Maturity Levels

Capability levels belong to a continuous representation. These levels apply to an organization's process improvement achievement in individual process areas. These levels are a means for incrementally improving the processes corresponding to a given process area. There are six capability levels. These levels are numbered 0 through 5 [4].

Maturity levels belong to a staged representation. These levels apply to an organization's process improvement achievement across multiple process areas. There are five maturity levels, numbered 1 through 5 [4]. Comparison of Capability and Maturity Levels are given in Table 2.

Table 2: Comparison of Capability and Maturity Levels [4]

Level	**Continuous Representation (Capability Levels)**	**Staged Representation (Maturity Levels)**
Level 0	Incomplete	N / A
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	Quantitatively Managed	Quantitatively Managed
Level 5	Optimizing	Optimizing

2.5.1 Capability Levels

The continuous representation of a selected CMMI model differentiates between capability levels to describe discrete levels of process improvement.

The continuous representation has the same basic information as the staged representation. The continuous representation focuses process improvement on actions to be completed within process areas.

These levels are called capability levels.

In a CMMI continuous representation there are six capability levels [11]:

- Level 0: Incomplete
- Level 1: Performed
- Level 2: Managed
- Level 3: Defined
- Level 4: Quantitatively Managed
- Level 5: Optimizing
- Incomplete: Process not performed or partially performed.
- Performed: Process satisfies all of the specific goals of the process area.
- Managed: Process is performed process that is also monitored, controlled and reviewed.
- Defined: Process is characterized as a defined process. A defined process is a managed process that is tailored from the organization's set of standard processes, and contributes work products, measures, and other process-improvement information to the organizational process assets.

- Quantitatively Managed: A quantitatively managed process is a defined process that is controlled using statistical and other quantitative techniques.

- Optimizing: An optimizing process is a quantitatively managed process that is adapted to meet current and projected business objectives.

These capability levels are described below:

Capability Level 0: Incomplete

An incomplete process does not apply all of the Capability Level 1 specific practices in the process area that has been selected. This is equivalent to Maturity Level 1 in the staged representation [11].

Capability Level 1: Performed

A Capability Level 1 process performs all of the Capability Level 1 specific practices. There is no stable performance. It can not be meet specific objectives such as quality, cost, and schedule. On the other hand, some useful work can be done. This is a starting point for process improvement activities. At this capability level, you perform some activities but you can not prove these works definitely. [11].

Capability Level 2: Managed

"A managed process is planned, performed, monitored, and controlled for individual projects, groups, or stand-alone processes to achieve an aim [4]". In capability level 2, a project actively manages in your organization. You develop some metrics to support your project. These metrics are consistently collected. You establish a management approach. However there is some absence for institutionalization.

Capability Level 3: Defined

A defined process is a managed process that is tailored from the organization's set of standard processes [4]." You have organization's set of standard processes and you have to fit you organizational standard processes. In capability level 3, you have some tailoring guidelines. If there are some deviations beyond those allowed by the tailoring guidelines, these are documented, reviewed, and approved.

Capability Level 4: Quantitatively Managed

A quantitatively managed process is a defined process that is controlled using statistical and some quantitative techniques. At this capability level, you have some statistical and quantitative techniques for product quality, service quality, process performance, and other business objectives and all of activities performs in statistical terms and controls throughout the life cycle [11].

Capability Level 5: Optimizing

"An optimizing process is a quantitatively managed process that improved based on an understanding of the common causes of process variation inherent in the process [11]." Capability level 5 focuses on continually improving process performance. This level contains some innovative improvement activities for project's defined processes and the organization's set of standard processes. This level focus on performance results across the organization. In this level, common causes of problems can be solved easily.

2.5.2 Maturity Levels

In a CMMI staged presentation, there are five maturity levels. When we select a CMMI model that uses a staged representation, we see some differences between these maturity levels that provide a strong way to predict the future performance of organizations:

- Initial. At this level processes unpredictable, poorly controlled and reactive. Organization processes are ad hoc and chaotic.

- Managed. At this level process characterized for an individual project and is often reactive. Requirements are managed and processes are planned, measured and controlled.
- Defined. Process is characterized for organization and is proactive. A process is established and improved over time.

- Quantitatively managed. At this level process measured and controlled. Quality and process performance measurement and detailed analyzing of the given results.

- Optimizing. Focus on continuous process improvement. Processes are continually improved by using collected quantitative improvement objectives.

These maturity levels are described below:

Maturity Level 1- Initial

There is no structured process in the organization. Development work contains chaotic and ad hoc issues. Budgets and schedules are often unstable. Product quality cannot be estimated [11].

Maturity Level 2 - Managed

There are basic project management processes and these processes are followed. Institutionalization is achieved by satisfying the generic goals and generic practices. There is adherence to organizational policies. Project personal follows a documented plan and process description. Project Managers use adequate funding and resources. There is an appropriate assignment of responsibility and authority. Training is important and personal are trained in their processes. There are configuration management principles. Work products are placed under appropriate configuration management. Process performance is monitored and controlled and corrective actions are taken. Processes, work products and services are objectively reviewed. Noncompliance issues management is applied. According to management disciplines, activities, status and results are reviewed. When necessary, corrective actions are taken. Interaction with relevant stakeholders is identified.

Maturity Level 3 - Defined

The organization has achieved all of the goals of Level 2. There is and organizational way of doing business, with tailoring of this organizational method allowed under predefined conditions. The organization has an organization's set of standard processes. Level 3 continues with defining a strong, meaningful, organization wide approach to developing products. An important distinction between Level 2 and Level 3 is that at Level 3, processes are described in more detail and more rigorously than at Level 2. Processes are managed more proactively, based on a more sophisticated understanding of the interrelationships and measurements of the processes and parts of the processes. Level 3 is more sophisticated, more organized, and establishes an organizational identity—a way of doing business particular to this organization [11].

Maturity Level 4 – Quantitatively Managed

For Maturity Level 4, the organization has achieved all of the goals of Levels 2 and 3. Statistical and quantitative techniques are at this level. Organizations use these techniques to controls their processes. Lots of quality improvement activities are performed, managed and understood in statistical terms throughout the life of the processes. This level focuses on using metrics to make decisions. At Level 4, processes are quantitatively predictable. You can find special causes of process variation and you can take corrective actions faithfully [11].

Maturity Level 5 – Optimizing

An organization has achieved all of the goals of Levels 2, 3, and 4. Processes are improved based on an understanding of common causes of variation within the processes [11].

2.6 Detailed Description of Some Process Areas (PPQA, PMC, PP, CM, MA)

The Process and Product Quality Assurance (PPQA), Configuration Management (CM) and Measurement and Analysis (MA) are support process areas at Maturity Level 2. Project Planning (PP), and Project Monitoring and Control (PMC) are project management process areas at Maturity Level 2.

2.6.1 Process and Product Quality Assurance (PPQA)

The Process and Product Quality Assurance (PPQA) is a support process area at Maturity Level 2.

"The purpose of Process and Product Quality Assurance (PPQA) Process Area is to provide staff and management with objective insight into processes and associated work products [4]."

Process and Product Quality Assurance Process Area involves objectively evaluation of performed process, work products and services against the process descriptions, standards and procedures. This process area identifies and documents noncompliance issues and provides some feedback to project managers and staff on the result of quality assurance activities. Main aim of this process is to establish a visibility of the project about process and products. This process area ensures that planned processes are applied. There is a strong relationship between Verification Process Area, Validation Process Area and Product and Process Quality Assurance Process Area. These two process area deals with some work product but they use different perspectives.

Process Quality Assurance Process Area supports objectivity because objectivity is the most important and critical factor to successful quality assurance. Objective evaluation is performed by an independent group, usually external to the projects, to process and product quality assurance tasks. For this reason, quality assurance personal should select outside of the project. However, some organizations with a history of successful process improvement may use any personal in the quality assurance process. In this statement, it may be sufficient to ensure that the person evaluating the processes and work products does not evaluate his own work to ensure their objectivity [12][2].

Quality assurance activities begin in the early phase of the project. Initially, a quality assurance plan is prepared. Later, all quality assurance activities perform according to quality assurance plan. When noncompliance issues are identified during auditing activities, these issues are reported and resolved. Corrective actions are monitored to closure. If there is a conflict during these audits, noncompliance issues escalate to management level by quality assurance person. In that way, the Quality Assurance staff becomes directly participated in process improvement activities [12][2].

Specific Goal and Practice Summary for PPQA are presented below [4]:

SG 1 Objectively Evaluate Processes and Work Products

- SP 1.1 Objectively Evaluate Processes
- SP 1.2 Objectively Evaluate Work Products and Services

SG 2 Provide Objective Insight

- SP 2.1 Communicate and Ensure Resolution of Noncompliance Issues
- SP 2.2 Establish Records

Process and Product Quality Assurance Context Diagram is given in Figure 4.

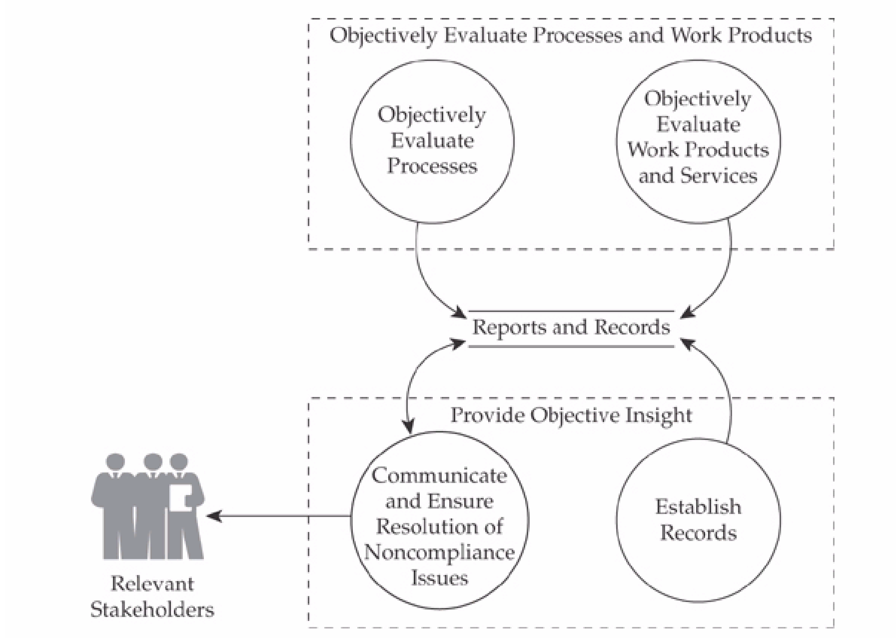

Figure 4: Process and Product Quality Assurance Context Diagram [1]

There are some expected work products from process and product quality assurance process area such as audit reports, noncompliance reports, corrective actions quality trends and process improvement suggestions etc.

2.6.2 Project Monitoring and Control (PMC)

The Project Monitoring and Control (PMC) is a project management process area at Maturity Level 2.

“The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the project’s progress so that appropriate corrective actions can be taken when the project’s performance deviates significantly from the plan [4].”

Specific Goal and Practice Summary for Project Monitoring and Control Process Area is presented below [4]:

SG 1 Monitor Project against Plan

- SP 1.1 Monitor Project Planning Parameters
- SP 1.2 Monitor Commitments
- SP 1.3 Monitor Project Risks
- SP 1.4 Monitor Data Management
- SP 1.5 Monitor Stakeholder Involvement
- SP 1.6 Conduct Progress Reviews
- SP 1.7 Conduct Milestone Reviews

SG 2 Manage Corrective Action to Closure

- SP 2.1 Analyze Issues
- SP 2.2 Take Corrective Action
- SP 2.3 Manage Corrective Action

Project Monitoring and Control Context Diagram is presented in Figure 5.

Project plans are the fundamental artifact of project monitoring and control. Project monitoring and control activities are performed according to project plans. Monitoring is done through project progress reviews and milestone reviews. Monitoring contains:

- measuring the actual values of planning parameters,
- comparing these parameters to the estimates in the plan and

- determining deviations.

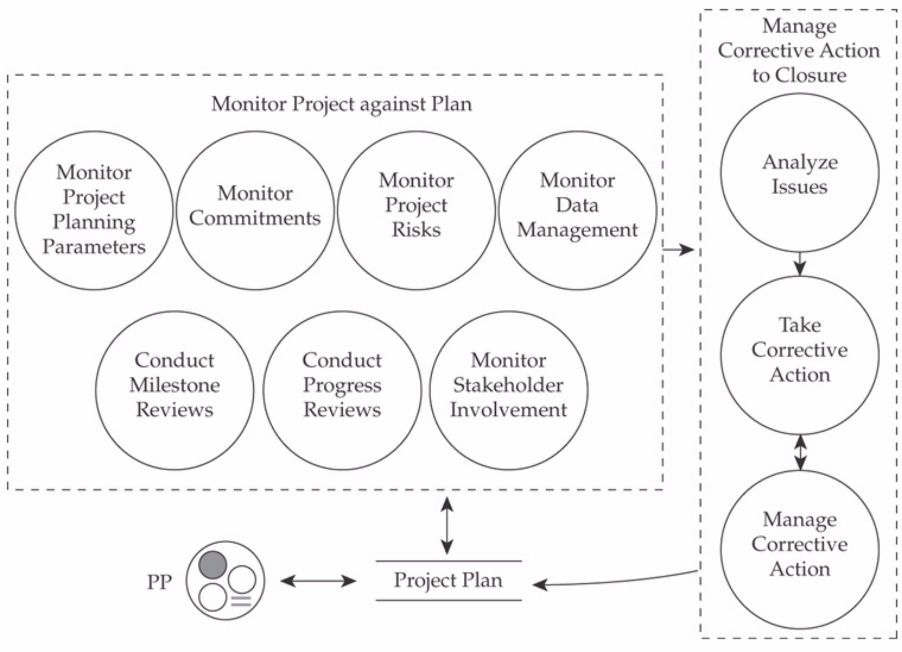

Figure 5: Project Monitoring and Control Context Diagram [1]

Main typical work products of this process are records of project performance and significant deviations. At this process area,

- Project progress is monitored against the project schedule,

- Project's cost and effort is monitored,

- Attributes of work products and tasks are monitored and

- Resources are monitored.

At this process area, external and internal commitments are reviewed and the results of these commitments are documented.

Project Plans contains some project related risks. Risks are periodically monitored according to plans. When necessary, risk documentation is revised. Risk monitoring records are collected and stored.

Project data are periodically monitored against the project plan. Data management records are collected and stored.

Stakeholder involvement status is periodically reviewed. Results of the stakeholder involvement status reviews are documented.

Project's progress, performance and issues are periodically reviewed. Project progress review meetings are performed with internal and external stakeholders. At these meetings, results of projects measures are reviewed. Significant issues and deviations are identified and documented. Problem reports and change requests are tracked to closure.

Milestone reviews meetings are conducted. Milestone review results are documented. At these meetings, project plan, status, project risks and commitments are reviewed. Significant issues are identified and action items are track to closure.

Issues are collected, analyzed. If necessary, appropriate corrective actions are determined and documented. Corrective actions are reviewed with relevant stakeholders. These actions are managed to closure.

There are some expected work products from project monitoring and control process area such as records of project performance, measurements of effort, budget, schedule, and staffing, monitoring of risk, stakeholder involvement, training records etc.[4][12].

2.6.3 Project Planning (PP)

The Project Planning (PP) is a project management process area at Maturity Level 2.

"The purpose of Project Planning (PP) is to establish and maintain plans that define project activities [4]."

Specific Goal and Practice Summary for Project Planning Process Area is presented below [4]:

SG 1 Establish Estimates

- SP 1.1 Estimate the Scope of the Project
- SP 1.2 Establish Estimates of Work Product and Task Attributes
- SP 1.3 Define Project Lifecycle
- SP 1.4 Determine Estimates of Effort and Cost

SG 2 Develop a Project Plan

- SP 2.1 Establish the Budget and Schedule
- SP 2.2 Identify Project Risks
- SP 2.3 Plan for Data Management
- SP 2.4 Plan for Project Resources
- SP 2.5 Plan for Needed Knowledge and Skills
- SP 2.6 Plan Stakeholder Involvement
- SP 2.7 Establish the Project Plan

SG 3 Obtain Commitment to the Plan

- SP 3.1 Review Plans That Affect the Project
- SP 3.2 Reconcile Work and Resource

Project Planning Context Diagram is given in Figure 6.

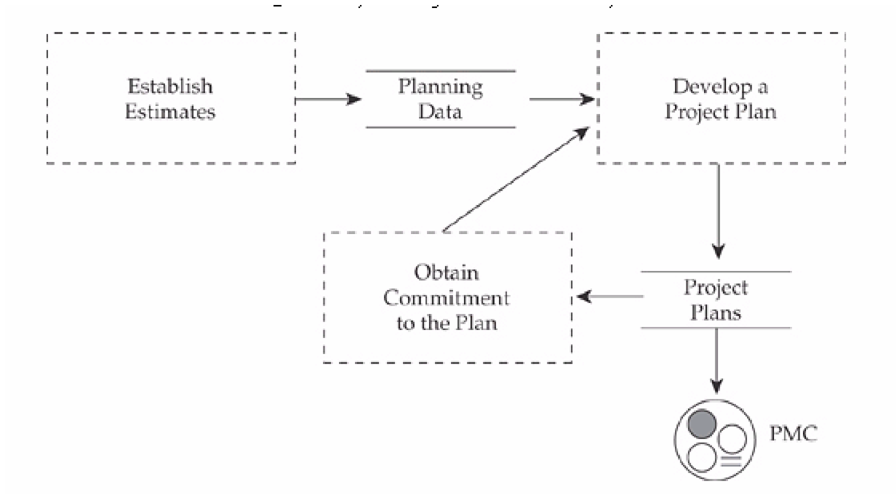

Figure 6: Project Planning Context Diagram [1]

The project planning process area contains basic project management activities. These are:

- developing estimates and project plan,
- interaction with stakeholders,
- revision of the project plan.

Project Planning Process Area involves some estimation activities for project. Estimates of project planning parameters are established. These parameters include information needed by the project to perform planning, budgeting, staffing, coordinating and reporting activities.

At this process area, based on an understanding of the requirements, scope of work is defined. Work breakdown Structure are established. Work products that will be produced are determined. Task descriptions are performed. Work package descriptions are produced.

Estimates of the attributes of the work products and tasks are established. Technical approach and estimation models for project are defined.

Project life cycle phases are defied. Project effort and cost for work products and tasks are estimated.

A project plan is established for project. Project's budged and schedule is established. Schedule dependencies are defined. Major milestones, schedule assumptions, constraints and task dependencies are identified.

Project risks is identified and analyzed. Risk impact and risk priorities are determined. Project data management is planned. Necessary resources for the project are planned. Staffing requirements, critical facilities and equipment lists are determined.

Required knowledge and skills are planned to perform the project. Availability of these knowledge and skills are assessed. For this purpose, a mechanism is established for providing needed knowledge and skills. Stakeholder involvement is planned.

There are some expected work products from project planning process area such as Project plan, stakeholder involvement plan, resource plan, data management plan, training plan, risk management plan, budget, and schedule [12].

2.6.4 Requirements Management (REQM)

The Requirement Management (REQM) is an engineering process area at Maturity Level 2.

"The purpose of Requirements Management (REQM) is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products. [4]".

Specific Goal and Practice Summary for Requirement Management Process Area are presented below [4]:

SG 1 Manage Requirements

- SP 1.1 Obtain an Understanding of Requirements
- SP 1.2 Obtain Commitment to Requirements
- SP 1.3 Manage Requirements Changes
- SP 1.4 Maintain Bidirectional Traceability of Requirements
- SP 1.5 Identify Inconsistencies Between Project Work and Requirements

Requirements Management Context Diagram is given in Figure 7.

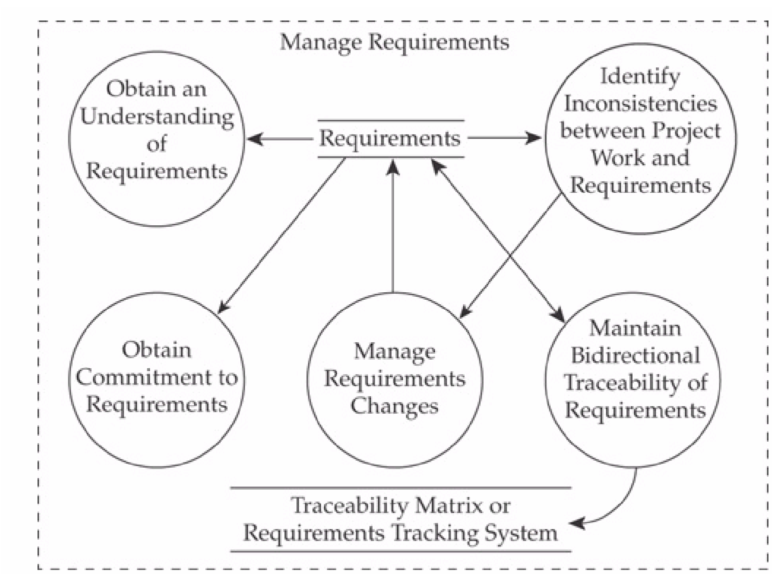

Figure 7: Requirements Management Context Diagram [1]

Requirement Management process manages and controls all documented product and product component requirements. The project maintains requirements over the life of the project. All changes to the requirements are managed. Relationships among the requirements, the project plan and work products are maintained. Inconsistencies among the requirements, project plans and the work products are identified. If necessary, corrective actions are taken.

Project staff develops an understanding with the requirements providers on the meaning of the requirements. A list of criteria is established for distinguishing appropriate requirement providers. Requirements are analyzed to ensure that established criteria are met.

Commitments are obtained from the project participant. Requirement impact analyzed is performed. These commitments are documented.

Requirements changes are managed according to the requirements management procedure. Requirements status is determined. Requirement database is established.

Traceability among the requirements and work products are established. Requirements traceability matrix is produced and a requirement tracking system is established.

Inconsistencies between project work and requirements are identified. If necessary, corrective actions are taken.

There are some expected work products for Requirement Management Process Area such as requirements changes and their status, impact of changes, requirements traceability, detected inconsistencies and their resolution [12].

2.6.5 Configuration Management (CM)

The Configuration Management (CM) is a support process area at Maturity Level 2.

“The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits [4]”.

Specific Goal and Practice Summary for Configuration Management Process Area [4]:

SG 1 Establish Baselines

- SP 1.1 Identify Configuration Items
- SP 1.2 Establish a Configuration Management System
- SP 1.3 Create or Release Baselines

SG 2 Track and Control Changes

- SP 2.1 Track Change Requests
- SP 2.2 Control Configuration Items

SG 3 Establish Integrity

- SP 3.1 Establish Configuration Management Records
- SP 3.2 Perform Configuration Audits

In this process area, a configuration management system is established. Baselines of identified work products are established. Configuration of the selected work products, components and other configuration items are identified. These configuration items are placed in configuration management system. According to a configuration management plan, baselines are created or released for internal use and for delivery of the customer.

Configuration Management Context Diagram is presented in Figure 8.

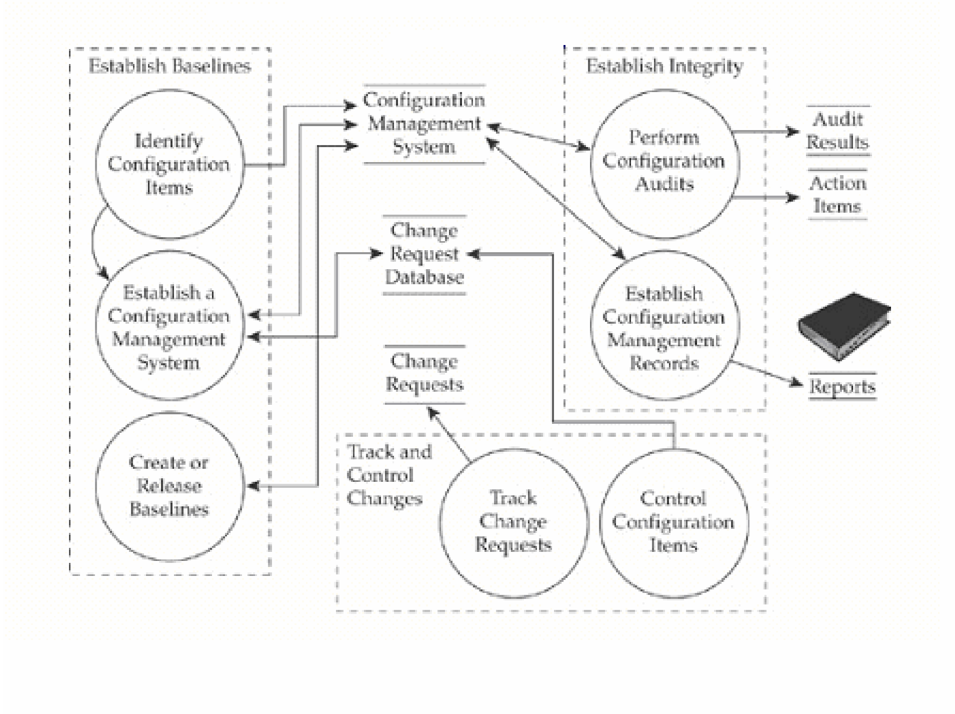

Figure 8: Configuration Management Context Diagram [1]

The Configuration Management Process Area addresses four considerable processes:

- Configuration identification,
- change control,
- configuration status accounting and
- configuration audits.

Configuration changes are tracked and configuration items are controlled in this configuration management system.

Integrity of baselines is established and maintained. Configuration management records are established and maintained.

Configuration status accounting and configuration audits (physical audits and functional audits) are performed.

2.6.6 Measurement and Analysis (MA)

The Measurement and Analysis (MA) is a support process area at Maturity Level 2.

"The purpose of Measurement and Analysis (MA) is to develop and sustain a measurement capability that is used to support management information needs. [4]".

Specific Goal and Practice Summary for MA [4]:

SG 1 Align Measurement and Analysis Activities

- SP 1.1 Establish Measurement Objectives
- SP 1.2 Specify Measures
- SP 1.3 Specify Data Collection and Storage Procedures
- SP 1.4 Specify Analysis Procedures

SG 2 Provide Measurement Results

- SP 2.1 Collect Measurement Data
- SP 2.2 Analyze Measurement Data
- SP 2.3 Store Data and Results
- SP 2.4 Communicate Results

Measurement and Analysis Context Diagram is presented in Figure 9.

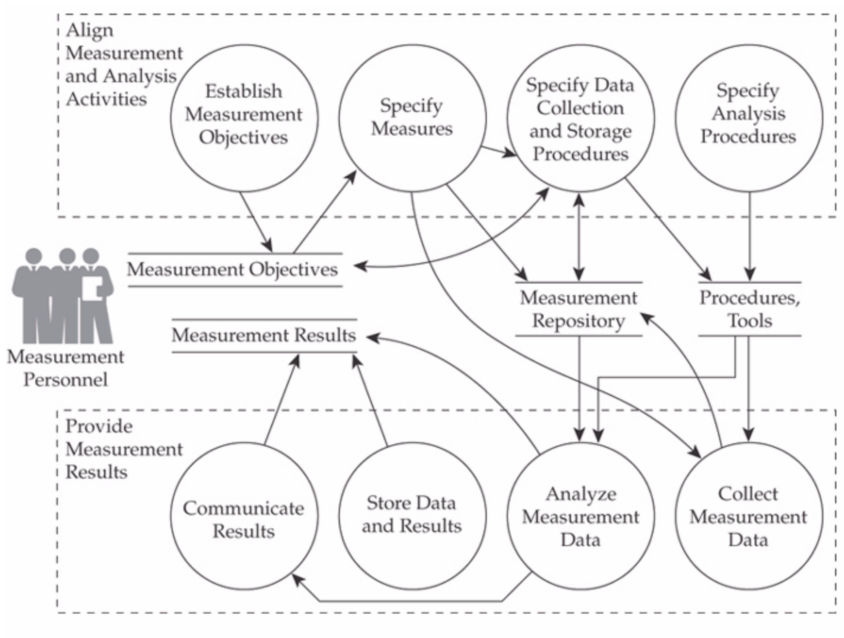

Figure 9: Measurement and Analysis Context Diagram [1]

In the Measurement and Analysis process area, measurement objectives are established and maintained. These measurement objectives are derived from identified information needs and objectives. Specifications of base and derived measures are defined. Data collection and storage procedures are establishes. If necessary, data collection tools are used. Measurement data is analyzed and reported. Measurement results are provided. Base and derived measurement data sets are prepared. The data are obtained and checked for completeness and integrity. Measurement data are analyzed. Measurement data, measurement specifications and analysis results are managed and stored. Measurement and analysis result are reported to all relevant stakeholders.

There are some expected work products for Measurement and Analysis Process Area such as business objectives, measurement objectives, measures collection and analysis tools, measurement data, and measurement reports etc. [12]

CHAPTER 3

ARTIFICIAL INTELLIGENCE TECHNIQUES AND CMMI

3.1 Artificial Intelligence in Software Engineering

"Software Engineering is a knowledge-intensive activity. It requires extensive knowledge of the application domain and of the target software itself. Many Software Engineering costs can be attributed to the ineffectiveness of current techniques for managing this knowledge, and Artificial Intelligence techniques can help alleviate this situation Artificial Intelligence techniques are necessary in order to build effective systems to support SE activities [13]".

Software engineering uses artificial intelligence tools or techniques to support human developers. There are some powerful artificial intelligence tools for knowledge representation and reasoning mechanisms. These tools can be useful for software development. A software project includes some critical tasks for organizing and utilizing knowledge and typically involves the services of both domain experts and software experts. Domain experts possess wide knowledge about a new software system and they do not understand software development extensively. Likewise, the software developers possess extensive knowledge about software development process, but they may have little experience in the application domain. For a number of years, researchers have been trying to use artificial intelligence (AI) techniques to automate the software engineering processes. So, there is a big gap and a significant risk for any software system. We can solve these problems using artificial intelligence techniques for software development process [14].

Artificial intelligence in software engineering can be used in three ways generally:

- automatic programming,
- software reuse and
- process modeling [14].

Automatic programming is the oldest method. “Automatic programming today means generating a machine-readable program from a high-level or natural language description of the problem to be solved”. Software reuse is currently a research focus and developers use process modeling for creating a knowledge base for understanding of the software system [14].

An automatic programming system would entirely remove human efforts from the business of coding large systems and can be more effective compared to human efforts. Maintenance effort can be simplified and new implementation of the system would be transformed automatically [14].

Another hot research topic in software engineering is reusability. If reusable components were widely available, tremendous savings could be realized. A reusable component can be stored in a library, and inserted into new software, so saving can be realized on a large scale. We have to think reusability at the beginning of the project. “A component library is a collection of software components that have been designed and certified for reuse”. In using a component library, we should develop a specification for a component that is to be inserted into a program. Next, we should identify a set of candidate components from the library, and we should evaluate each candidate for its suitability. Consequently, we should select one candidate as being the best. If this candidate is reasonably close to the desired specification, we should modify this component and then integrate into the program. Finally, we should record the total experience of this component, and we should insert this information into the component library’s experience base. The library's experience base is vital to the successful use of the library [14].

Software process models are used to organize a software development project. The Capability Maturity Model is the most popular software process model. Process models provide detailed reasoning about different features of the process, provide guidance to software developers and they help determining certain process steps, and provide a mechanism for process improvement. These tasks can be achieved with artificial intelligence process models by using knowledge representation schemes or using the Goal-Question-Metric technique [14].

There are lots of suggestions for using AI technology in software engineering. For example, "the tools for artificial intelligence development would be very useful for generating system prototypes in a general software engineering project. There are also a number of computationally intelligent approaches for supporting software engineers" [14].

"In last decides "process thinking" has been one of the major efforts to make software engineering as a professional engineering discipline"[15] . Processes are used in many widely known theories and essential practices of Software Engineering (SE) and closely related disciplines. CMMI is the most popular model between these process improvement practices.

There is a strict relationship between knowledge engineering and software engineering. Software Engineering is not only a technical discipline of its own. It is also a problem domain where technologies coming from other disciplines are relevant and can play an important role. One important example is knowledge engineering, a term that we use in the broad sense to encompass artificial intelligence, computational intelligence, knowledge bases, data mining, and machine learning. We see a number of typical software development issues that can benefit from these disciplines. We may divide them into four categories [16]:

- Planning, monitoring, and quality control of projects,

- The quality and process improvement of software organizations

- Decision making support,

- Automation.

When we consider process improvement activities in software engineering, we should use some artificial intelligence techniques in our works. If we want to upgrade the power of software systems, then we shall have to incorporate AI into practical software systems [17].

3.2 Computational Intelligence

Computational Intelligence (CI) describes several synergistic intelligent technologies that are effective in modeling systems, processes and decision making under uncertain conditions [14]. "Computationally intelligent system deals only with numerical data, has a pattern recognition component, and does not use knowledge in the AI sense; and additionally when it exhibit

(i) computational adaptivity;

(ii) computational fault tolerance;

(iii) speed approaching human-like turnaround, and

(iv) error rates that approach human performance ..." [18].

Computational intelligence technologies are complementary to one another. Some of the computational intelligence technologies are presented below [14]:

- Fuzzy sets and fuzzy logic model,

- Artificial neural networks,
- Genetic algorithms and
- Fractal sets and chaotic systems.

"One of the core philosophies of CI is to match a problem to the modeling technique best suited to solve it, instead of adopting a one-size-fits-all approach. Thus, fuzzy sets have seen extensive use in expert systems, while neural networks have been widely used as intelligent classifiers. There are also close relationships between CI and other technologies such as case based reasoning, machine learning and data mining [14]".

According to W. Duch [19], "AI is a part of CI focused on problems related to higher cognitive functions, while the rest of the CI community works on problems related to perception and control, or lower cognitive functions. Grand challenges on both sides of this spectrum are addressed".

3.3 Ontology-Supported Knowledge Representation and Software Processes

Ontologies are used for Knowledge-based systems to organize concepts and relationships between concepts. An ontology is at the heart of any knowledge-based system. A definition for an ontology is a logical theory. It gives an explicit partial account of a conceptualization. it is an intentional semantic structure that encodes the implicit rules constraining the structure of a piece of reality [20]. Every knowledge model has an ontological commitment. Every knowledge model has a partial account of the intended conceptualization of a logical theory. There are several types of ontologies and there are different researchers classify ontology types differently. Besides, there are different methodologies available for ontologies [21].

Ontology development is basis to most knowledge management works [22]. Ontological engineering has garnered increasing attention over the last few years [23]. Ontological engineering encompasses a set of activities conducted during conceptualization, design, implementation and deployment of ontologies. Ontological engineering covers topics including philosophy, metaphysics, knowledge representation formalisms, development methodology, knowledge sharing and reuse, knowledge management, business process modeling, commonsense knowledge, systematization of domain knowledge, information retrieval from the Internet, standardization, and evaluation [24]. It also gives us design rationale of a knowledge base. It helps define the essential concepts of the world of interest and It allows for a more disciplined design of a knowledge base, and enables knowledge accumulation [25].

Ontologies provide the basic structure around which knowledge bases can be built [26]. Each ontology is a system of concepts and their relations. The system defines the vocabulary of a problem domain and a set of constraints on how terms can be combined to model the domain [25].

Many similarities exist between ontological engineering and software engineering disciplines such as software architectures and software patterns. Many practitioners understand similarities between ontological engineering and the object-oriented paradigm, and similarities between phases of the ontology development and software development processes [25].

"Recently, the research on the ontology has been spread widely to be critical components in the knowledge management, Semantic Web, business-to-business applications, and several other application areas[27]. "

The development of software is one of the most knowledge intensive processes that occur in modern organizations and involves many intellectual assets, including developers' personal skills, artifacts, best practices, and process experiences. It is necessary to deploy these process assets through the organization. By doing this, developers can facilitate organizational memory and historical data to concentrate their creativity on solving technical problems [28][29]. For these purposes, we have design and organize our process assets appropriately and effectively. So, we have to use some ontology-based knowledge representation techniques. In the literature, there are lots of ontology supported knowledge presentation techniques for software processes such as ProKnowHow [30], BORE [31] and SPO [32].

BORE (Building an Organizational Repository of Experiences) is a prototype tool designed to further explore and refine the requirements for tools supporting experience based approaches [31].The BORE tool aims reusing organizational experience through packaging it in experience repositories [28][31].

ProKnowHow is developed to support the standard process tailoring for the projects. Allowing the knowledge acquired in this process to be shared. Also, ProKnowHow is based on a standard established software process and could collect and disseminate the knowledge acquired during standard process instantiation [28][30].

SPO (Software Process Ontology) is designed and extends to generate ontologies for specific process models, such as CMMI and ISO/IEC 15504. SPO is proposed to support software process definition and assessment [28][32].

According to Haihua, Liu, and Jin [28], software process knowledge can be generally categorized into three types:

Haihua, Liu, and Jin,

- Process experiences,
- Knowledge artifacts,
- Personal skills.

Each type of knowledge has its own ontologies that built manually. All of the ontologies' instances establish an organizational knowledge repository [28].

This categorization is presented in Figure 10:

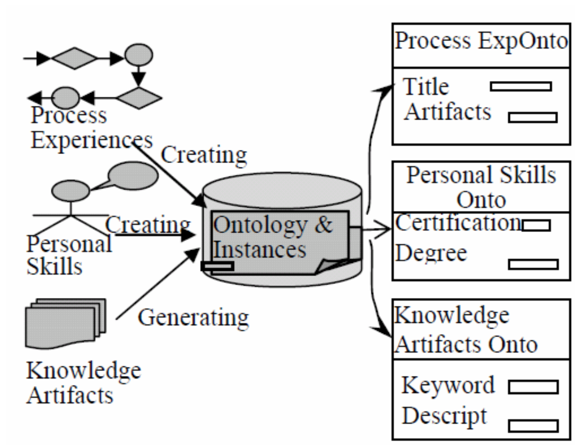

Figure 10: Conceptual Model of Software Process Knowledge Representation [28]

3.3.1 Ontology for Process Experiences

An organizational knowledge repository contains process experiences. We have to disseminate the process experiences through the whole organization.

Process experiences characteristics mainly consist of some issues presented below [28]:

- High dependency on specific project:

  There are some factors influence software product related to dependency on a specific project (such as application domain, team features, development technology and project size etc) [33].

- Hard to follow a given process model:

We can not apply all of existing process models to every kind of software practice.

- Reusable between similar projects:

We can use historical data for reference in future.

In the literature, there are lots of software process models. Although the existing software process models are different, they have some similarities. Normally, the models have a set of processes are classified into several domains, called "Subsystem" or "Category" [28].

These similarities are presented Table 3:

Table 3: Similarities of Existing Standard Software Process [32]

Model / Component	Sub System	Category	Process	Subprocess	Practice	Process Attribute
CMM		Category	Key Process Area		Key Practice	
CMMI		Category	Process Area	Specific Goal	Specific Practice	Generic Goal
ISO/IEC 15504		Category	Process	Compo nent Process	Basic Practice	Process Attributes
ISO 9001	Sub system		Main Topic Area		Manage ment Issue	

Haihua, Liu, and Jin [28] suggests a process experiences ontology presented in Figure 11. The proposed ontology defines the process model at the schema level and plays a similar role to conceptual data schemas in the database community.

In this ontology, they define some core classes to represent components in models, and properties to represent the relationship between components and the core classes are chosen from the shared concept in CMM, CMMI, ISO/IEC15504 and ISO 9001.

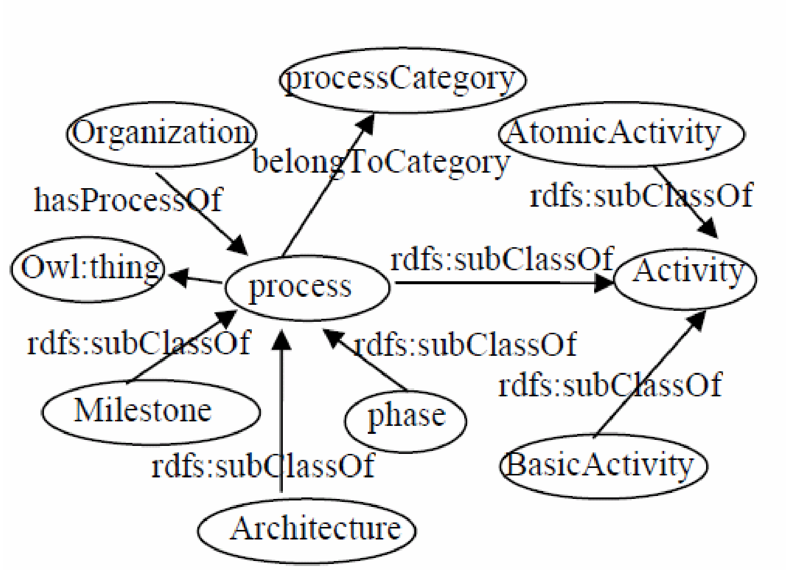

Figure 11: Process Experiences Ontology [28]

3.3.2 Ontology for Personal Skills

Personal skill is the most important factor to capture and represent knowledge. Because all software development activities performed by human. For this reason, we have to collect and store this knowledge and shared across the organization [28].

Main characteristic of personal skills presented below [28]:

- Tacit:

We have to elicit personal skills and It should be became explicit knowledge.

- High value:

Tacit knowledge is the most valuable asset for an organization.

- Unstable:

When a personal leave the organization, these valuable assets might lost.

Haihua, Liu and Jin [28] define personal skills ontology in a hierarchal structure of different kinds of skills a person can possess regarding the above characteristics. Skills are obtained from various sources, such as certified training, formal education or previous work experience [34]. Because skills have a great effect on the way employees performs tasks.

3.3.3 Ontology for Knowledge Artifacts

An organization's knowledge repository contains knowledge artifacts obtained throughout the projects.

There are four major artifact types [35]:

- Technical,
- Quality,
- Safety and
- Management.

Although there is a growing interest in tools and methods that support ontology automatic generation from document corpus, results are far from mature and practical [36].

Haihua, Liu, and Jin [28] developed an ontology to achieve following goals:

- To support useful process assets to be safely composed and organized into an knowledge repository,

- To represent the knowledge generated and acquired during process of software development,

- To support retrieval and dissemination of stored knowledge from organizational repository.

3.4 Recent Artificial Intelligence Applications for CMMI

Ontology- based applications seem to dominate for CMMI applications. In "Ontology-based Intelligent Decision Support Agent for CMMI Project Monitoring and Control"[37], Lee, Wang and Chen propose an ontology-based intelligent decision support agent (OIDSA) applied to project monitoring and control of Capability Maturity Model Integration (CMMI) Model. The work of the agent is based on the analysis and extraction of information from linguistic data sets. The OIDSA is composed of a natural language processing agent, a fuzzy inference agent, and a performance decision support agent. The fuzzy inference agent computes the similarity of the planned progress report and actual progress report, based on the CMMI ontology, the project personal ontology, and natural language processing results. The results provided by the OIDSA support an evaluation of the performance of Project members by project manager. "The evaluation shows that the Ontology-based Intelligent Decision Support Agent (OIDSA) for CMMI project monitoring and control can effectively evaluate the completed percentage of progress to reduce the human effort and the costs of the project [37]."

In another study entitled "An Intelligent Fuzzy Agent based on PPQA Ontology for Supporting CMMI Assessment [38]", based on the process and product quality assurance (PPQA) process area of CMMI, Wang and Lee propose an intelligent fuzzy agent based on PPQA ontology for supporting CMMI assessment. "The experiments show that the proposed approach can effectively operate for the summarization of the evaluation reports [37]."

In "Intelligent Estimation Agent Based on CMMI Ontology for Project Planning [39]", based on the process planning (PP) process area of CMMI, Lee, Wang, Yan et. al. propose an intelligent estimation agent based on

CMMI ontology for project planning. "The simulation results show that the proposed agent is feasible to estimate the total project cost [39]."

In "CRM Ontology Based on CMMI Project Planning for Business Applications [40]", based on the process planning (PP) process area of CMMI, Lee, Wang, Liu, and Lin propose a Customer Relationship Management (CRM) Ontology based on CMMI project planning for business application. "The CRM ontology describes the CRM project planning knowledge based on CMMI. It contains three categories, including establishing CRM project estimates, developing CRM project plan, and obtaining CRM project commitment. There are various concepts in the class layer and some instances in the instances layer defined in the ontology. The proposed ontology can be applied to construct an intelligent CRM system based on CMMI project planning for business applications [40]."

In "Selection Priority of Process Areas Based on CMMI Continuous Representation[41]", Huang and Han present a decision support model that assists managers in determining the priorities of the CMMI process areas based on the characteristics of the is being developed. "The proposed model was validated by using a repository and an example is presented to demonstrate the application of the model. Given the fact that hardly any research has yet been done on how to select the CMMI process areas to initialize process improvement, this study provides a starting point for the community in considering this important issue [41]."

In "An Intelligent Early Warning System for Software Quality Improvement and Project Management [42]", Liu, Kane and Bambroo discuss design, implementation, and evaluation of an experimental intelligent software early warning system based on fuzzy logic using an integrated set of software metrics. "It consists of the following components: software metrics database, risk knowledge base, intelligent risk assessment, and risk tracing. It helps to assess risks associated with the three problems from perspectives of product, process, and organization in the early phases of the software development process. It is capable of aggregating various partial risk assessment results into an overall risk indicator even if they may be conflicting. In addition, it can be used to analyze a risk by identifying its root causes through its risk tracing utility [42].”

In “Analysis of Ontology Development Methodology Based on and CMMI Level 4 [43]”, Lee, Choi and Kim describe relations between 5 steps of OTK (On to Knowledge) development process and process areas belong to CMMI level 4 and provide collaboration details how to adopt CMMI level 4 process areas to OTK (On to Knowledge) methodology for quality management of ontology development. “OTK methodology which is one of the most frequently used modeling method for ontology development nowadays but for effectiveness of development, it requires quality management of development itself. To provide quality management, CMMI reference model for process improvement suggests effective way of quality management by quantitative way. Especially level 4 process areas of CMMI provide practical approach to manage quality of projects which face quality problems of their projects [43].”

In “Software Quality in Artificial Intelligent System [44]”, Viyanasundaram and Srivatsa define the metrics to measure the quality of software in the architecture for an artificial intelligence system. “The proposed architecture for measurement consists of four components. These are task specification layer, problem solver layer, domain layer and an adapter layer. These four components are hierarchically organized in a layered fashion. In this architecture, the overall structure is decomposed into sub components, in a layered way such that a new layer can be added to the existing layer that can change the behavior of the system [44].”

In “KM Your Way to CMMI [45]”, Dayan and Evans suggest knowledge management principles to CMMI model.

In “An Intelligent PPQA Web Services for CMMI Assessment [46]”, Wang and Lee propose an an intelligent Process and Product Quality Assurance (PPQA) web services for Capability Maturity Model Integration (CMMI) assessment. “The intelligent web services are implemented by following the specific goals of PPQA process area of CMMI and contain an evaluating processes and work products service (EPWPS), a providing objective insight service (POIS), an ontological reasoning agent (ORA), and a message notification service (MNS). With the help of this intelligent web services, the stakeholders of the Project can get the noncompliance reports, evaluation logs, and quality assurance reports through the provided user interface. In addition, they also can receive an email to get the established reports during performing quality assurance activities. The simulation results show that the intelligent web services can support the PPQA of CMMI assessment effectively [46]."

In "An Ontology for CMMI-ACQ Model [47]", Sharifloo, Motazed, Shamsfard and Dehkharghani discuss an ontology developed to represent the CMMI-ACQ domain knowledge. "CMMI introduces a collection of best practices that helps organizations improve their processes and CMMI-ACQ as one of its constellations is to provide organizations with a resource of effective and proven practices to support acquisition process improvement. This ontology has been developed based on SUMO upper ontology. Through the paper, CMMI and CMMI-ACQ will be illustrated. Also the ontology, its development methodology, and its potential application will be described [47]."

In "How can we assess Knowledge Management? Constructing a Holistic Assessment Framework of KM [48]", Hung, Chou and Chen proposes two knowledge management maturity models as a means to help answer these and related questions. They have constructed a KMMM (Knowledge Management Maturity Model) Pyramid model from the perspective of knowledge management processes, and integrated knowledge management enablers into the model. "The Staged representation can be used to evaluate how well organizations perform in knowledge management, and provide maturity paths that organizations can follow. Continuous Representation is represented by a set of points in three dimensions. The applicability of this representation is assessed through a scale of 12 perspectives. These two representations provide different assessment approaches to help in determining how well organizations perform in terms of knowledge management, and provide maturity paths that organization can follow [48].” Figure 12 presents a Knowledge Management Maturity Pyramid Model. Figure 13 presents an element relationship diagram in KMMM.

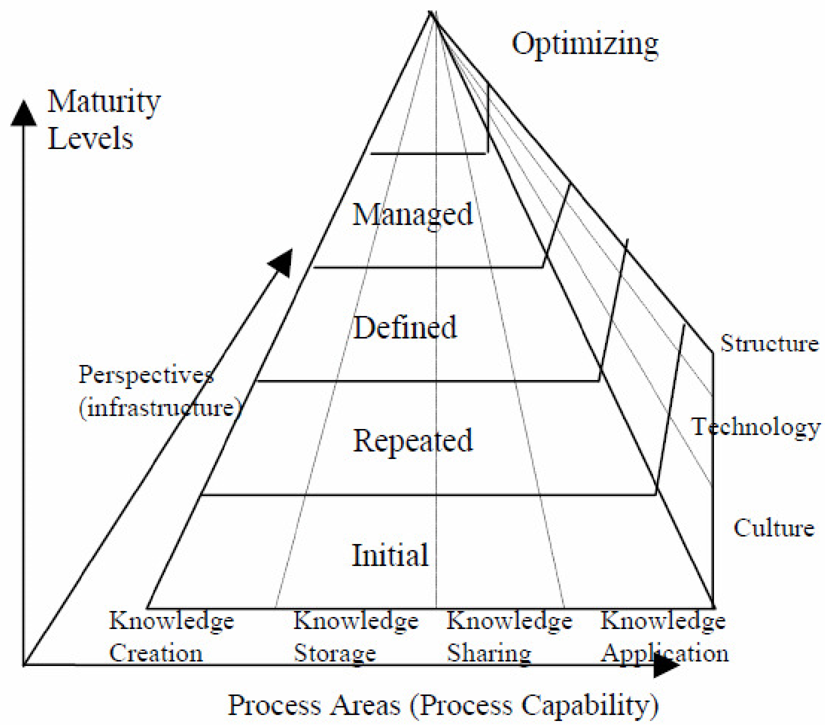

Figure 12: KMM (Knowledge Management Maturity) Pyramid Model [48]

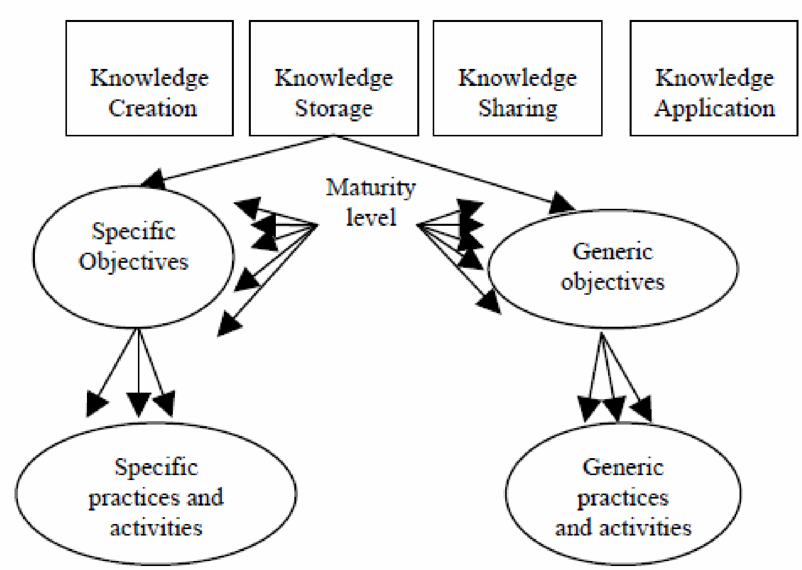

Figure 13: Element Relationship Diagram in KMMM [48]

3.4.1 Artificial Intelligence Applications for PPQA Process Area

"Ontology is an essential part of many applications. An ontology is a formal conceptualization of a real world, and it can share a common understanding of this real world. Supported by an ontology, both the user and the system can communicate with each other using a common understanding of a domain" [49], [50].

"A domain ontology defines a set of representational terms that we call concepts. Inter-relationships among these concepts describe a target world. The domain layer represents the domain name of an ontology and consists of various categories defined by domain experts. Every event comprises several concepts of class layer. In class layer, each concept contains a concept name an attribute set and an operation set for an application domain [49]."

In "An Intelligent Fuzzy Agent based on PPQA Ontology for Supporting CMMI Assessment[38]", based on the process and product quality assurance (PPQA) process area of CMMI, Wang and Lee propose an intelligent fuzzy agent based on PPQA ontology for supporting CMMI assessment.

The structure of domain ontology used in this work is presented in Figure 14:

The structure of ontology used in this work is composed of three layers:

- Domain Layer,
- Category Layer,
- Concept Layer.

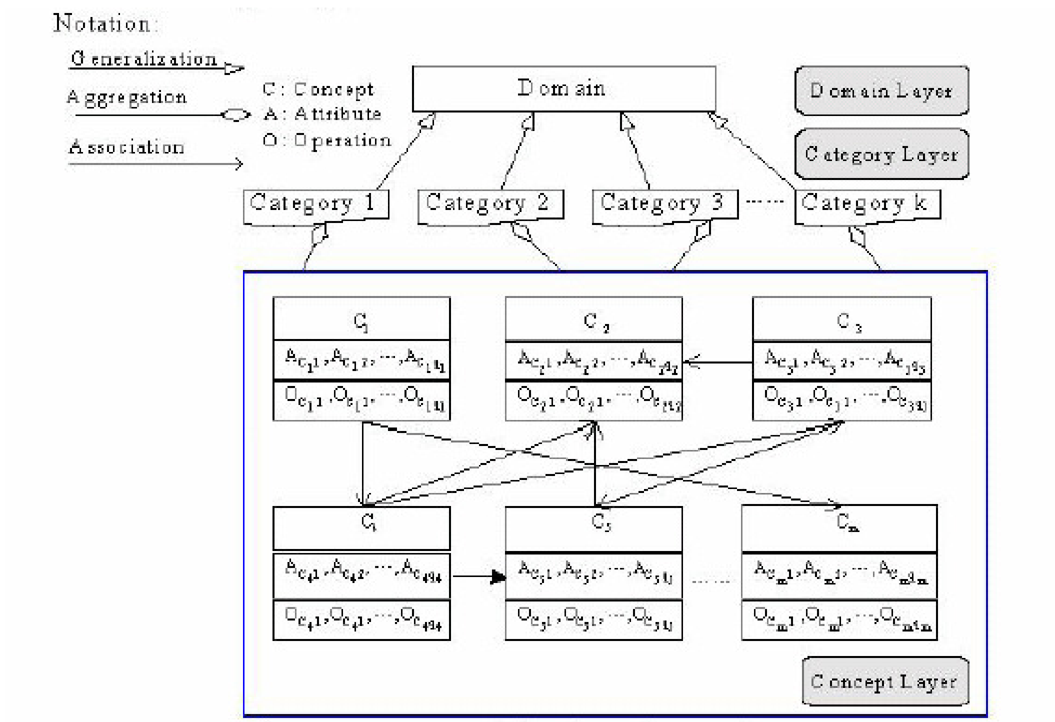

Figure 14: Structure of the Domain Ontology [38]

Domain name is presented in domain layer. For Process and Product Process Area (PPQA), domain name is "Software Process and Product Quality Assurance".

Domain experts can define various categories. For Process and Product Process Area (PPQA), category layer can be divided into two categories, such as

- Process Quality Assurance Category and
- Product Quality Assurance Category.

Concept layer contains:

- a concept name,
- an attribute set and
- an operation set.

Wang and Lee [38], construct the concept layer according to the five sub-layers, including

- who layer,
- when layer,
- what layer,
- where layer and
- how layer.

Wang and Lee propose a Process and Product Quality Assurance (PPQA) ontology.

Wang and Lee, structured the system architecture and the intelligent fuzzy agent based on PPQA Ontology for Supporting CMMI Assessment.

The structure of the PPQA ontology is shown in Figure 15.

System architecture of an intelligent fuzzy agent based on PPQA is presented in Figure 16.

System architecture steps are presented below [38]:

- Domain experts predefine PPQA ontology according to ontology generating system.
- The PPQA team members evaluate the processes and products according to defined processes.
- The PPQA team members identify and record noncompliance into evaluation reports.

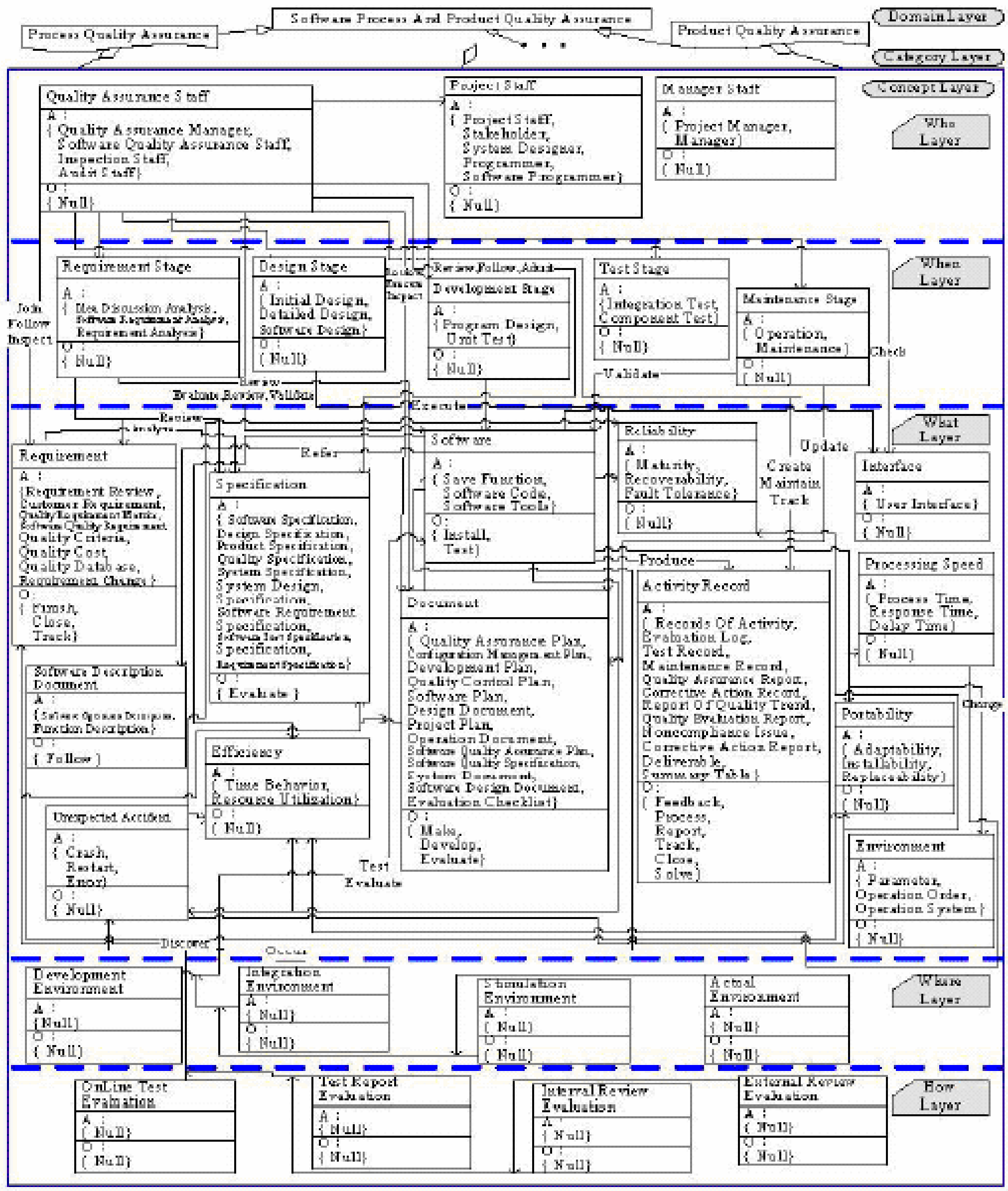

Figure 15: The Structure of the PPQA Ontology [38]

- Based on the Chinese Dictionary and the PPQA ontology, the intelligent fuzzy agent proceeds.

- The summarized report is stored in the PPQA report repository.
- With the key features of the evaluation reports, the related personnel can know if there is anything needed to be corrected for the project.

- The related personnel are able to retrieve the information stored in the PPQA report.

- The information presented on the PPQA reporting system is able to support the CMMI assessment tools.

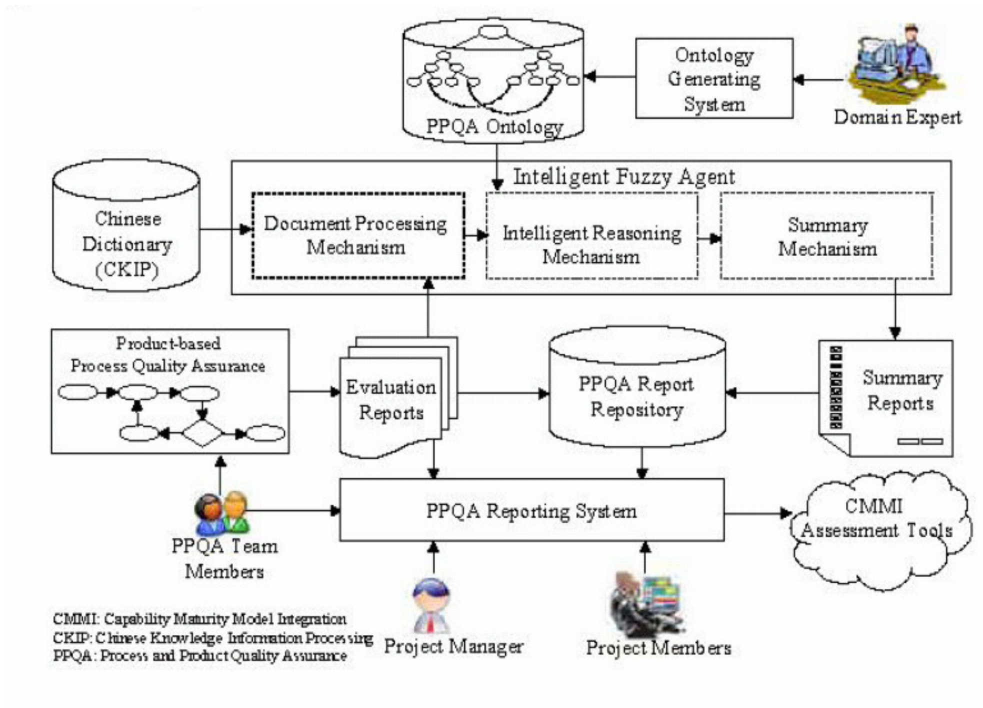

Figure 16: System Architecture of an Intelligent Fuzzy Agent Based on PPQA [38]

According to Wang and Lee, the intelligent fuzzy agent is composed of

- a document processing mechanism,
- an intelligent reasoning mechanism and
- a summary mechanism.

The document processing mechanism includes

- a Part-of-Speech (POS) tagger and

- a term filter.

The document processing mechanism tasks is presented below [38]:

- It tags the terms of evaluation reports,

- It filters meaningless terms to preserve the terms whose POS is noun or verb,

- It passes the results to the intelligent reasoning mechanism.

The conceptual similarity factors for making analysis of the Chinese terms are

- POS similarity,

- Term Word (TW) similarity and,

- Semantic Distance (SD) similarity,

The document processing mechanism calculates the conceptual similarity between any two Chinese terms. The POS similarity represents the path length between two nodes located on the tagging tree. The POS similarity represents the path length between two nodes located on the tagging tree. The tagging tree is adopted to calculate the conceptual similarity [38].

The intelligent reasoning mechanism makes use of the results of the document processing mechanism of the evaluation report to reason the term relation strength's membership degrees belonging to the PPQA ontology. The brief fuzzy rules pre-defined by domain experts are used [38].

The brief fuzzy rules pre-defined by domain experts are presented in Table 4.

Table 4: The Brief Fuzzy Rules Pre-defined by Domain Experts [38]

FUZZY RULES OF THE INTELLIGENT REASONING MECHANISM
Rule 1: if *POS* is **Low** and *TW* is **Low** and *SD* is **Low** then *TRS* is **Very Low**
Rule 2: if *POS* is **Low** and *TW* is **Low** and *SD* is **Medium** then *TRS* is **Low**
Rule 3: if *POS* is **Low** and *TW* is **Low** and *SD* is **High** then *TRS* is **Low**
⋮
Rule 19: if *POS* is **High** and *TW* is **Low** and *SD* is **Low** then *TRS* is **Low**
Rule 20: if *POS* is **High** and *TW* is **Low** and *SD* is **Medium** then *TRS* is **Medium**
Rule 21: if *POS* is **High** and *TW* is **Low** and *SD* is **High** then *TRS* is **High**
⋮
Rule 27: if *POS* is **High** and *TW* is **High** and *SD* is **High** then *TRS* is **Very High**

The summary mechanism has five sub-processes. These sub-processes are presented below [38]:

- The matched concept finder,
- Sentence patch extractor,
- Sentence patch filter,
- Sentence generator,
- Comparison rate computation,

The summary mechanism tasks are briefly presented below [38]:

- The matched concept finder picks up the terms pairs whose Term Relation Strength (TRS) values are over the threshold.
- The matched concept finder finds out the matched concept.

- Sentence patch extractor takes advantage of the matched concepts together with the PPQA ontology to search all the possible sentence paths.
- Sentence patch filter removes the redundant sentence paths.

- Sentence generator produces the key sentences of the evaluation report according to the retained sentence paths.

- Comparison rate computation calculates the computation rate of the evaluation report.

The Sentence Path Extractor extracts a set of possible sentence paths from the fuzzy ontology and sends the template set to Sentence Generator. The Sentence Generator produces a set of sentences from the class layer of the fuzzy ontology. Finally, the Sentence Filter filters noisy sentences and creates a set of summarized sentences. The Sentence Path Extractor uses the Depth-First-Search algorithm to look for possible sentence paths. The Sentence Generator generates a set of sentences based on the temporary fuzzy ontology. The Sentence Filter is used to filter the redundant sentences. It will generate a brief sentence set. The Sentence Filter will combine the common sentences and transfer them into semantic sentences [49].

Consequently, experimental results indicate that the ontology-based computational intelligent multi-agent can effectively summarize the evaluation reports for the CMMI assessment [37].

In “"A fuzzy ontology and its application to news summarization [49]”, Lee, Jian and Huang propose a fuzzy ontology and its applications to new summarization. In this work, they used similar ontology. “The domain ontology with various events of news is predefined by domain experts. The document preprocessing mechanism will generate the meaningful terms based on the news corpus and the Chinese news dictionary defined by the domain expert. Then, the meaningful terms will be classified according to the events of the news by the term classifier. The fuzzy inference mechanism will generate the membership degrees for each fuzzy concept of the fuzzy ontology. Every fuzzy concept has a set of membership degrees associated with various events of the domain ontology. In addition, a news agent based on the fuzzy ontology is also developed for news summarization. The news agent contains five modules, including a retrieval agent, a document preprocessing mechanism, a sentence path extractor, a sentence generator, and sentence filter to perform news summarization [49]."

3.4.2 Artificial Intelligence Applications for Project Monitoring and Control Process Area

In "Ontology-based intelligent decision support agent for CMMI project monitoring and control [38]", based on the project monitoring and control (PMC) process area of CMMI, Lee, Wang and Chen propose an ontology based intelligent decision support agent for CMMI project monitoring and control.

The structure of ontology used in this work is composed of three layers:

- Domain Layer,
- Category Layer,
- Concept Layer.

Domain name is presented in domain layer. For partial CMMI Level 2 process areas, domain name is "Partial CMMI Level 2 Process Areas".

Domain experts can define various categories. Partial CMMI Level 2 process areas, category layer can be divided into three categories, such as

- Requirement Management,
- Project Planning and

- Project Monitoring and Control.

Concept layer contains:

- a concept name
- an attribute set and
- an operation set.

Lee, Wang and Chen [51], construct the concept layer according to the five sub-layers, including

- who layer,
- when layer,
- what layer,
- where layer and
- how layer.

Lee, Wang and Chen [51], propose a CMMI ontology for some CMMI Level 2 process areas. The structure of the CMMI ontology is shown in Figure 17.

Lee, Wang and Chen [51], structured the system architecture for ontology-based intelligent decision support agent based on some CMMI process areas.

Structure of the ontology-based intelligent decision support agent presented in Figure 18.

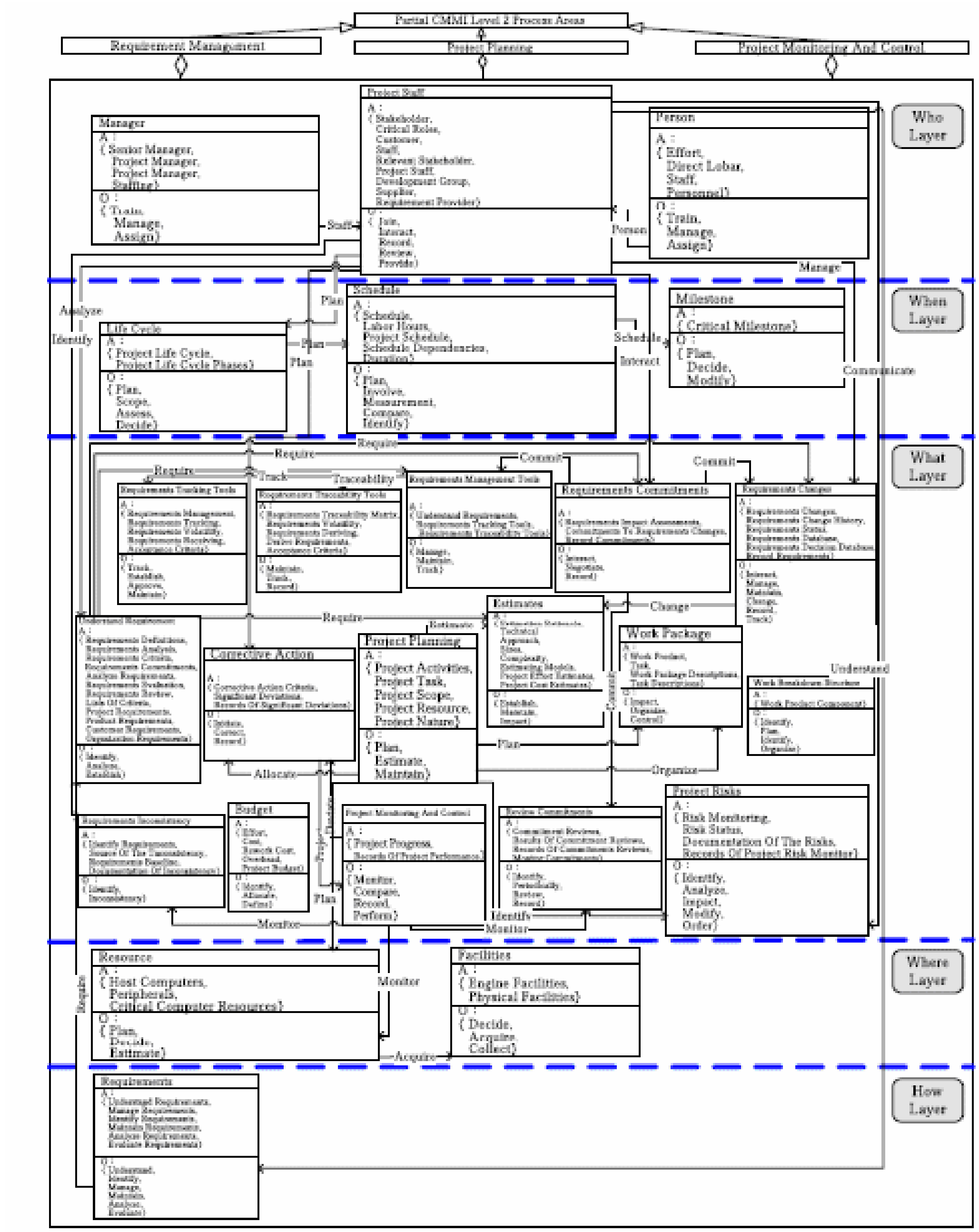

Figure 17: Structure of the CMMI Ontology for RM, PP and PMC [51]

System architecture steps are presented below [51]:

- Domain experts (CMMI experts and project domain experts) predefine CMMI ontology according to ontology generating system.

- The project members periodically fill in the planned and actual progress reports

- The project members store planned and actual progress reports into project progress repository.

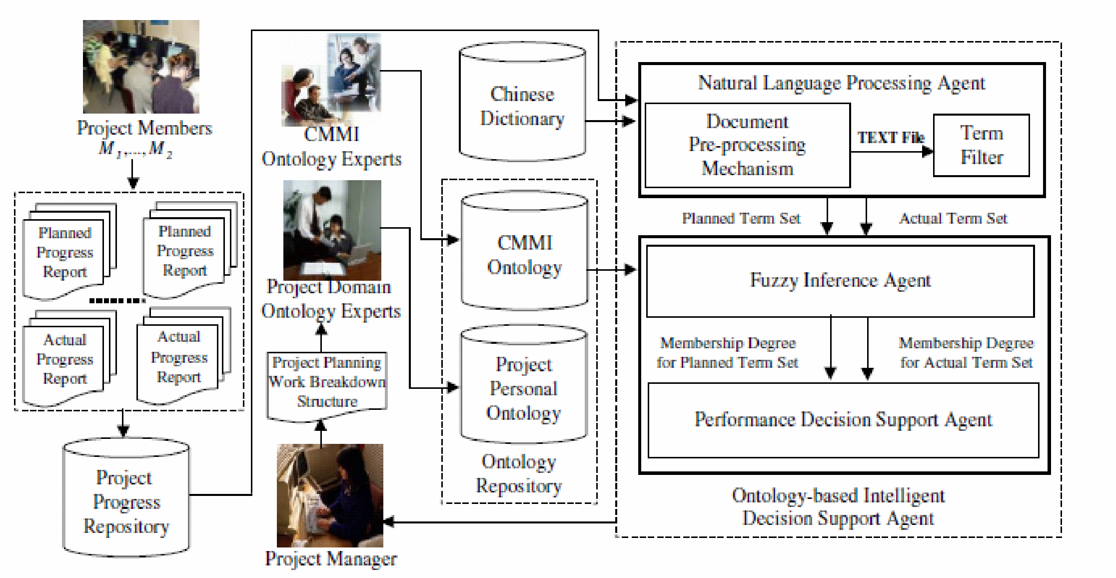

Figure 18: Structure of the Ontology-based Intelligent Decision Support Agent [51]

- Based on the Chinese Dictionary, the natural language processing agent collects the progress reports from the project progress repository.

- The document pre-processing mechanism tags the terms of progress reports with Part-of-Speech (POS) such as noun, verb, and adjective.

- The term filter keeps the meaningful terms whose POS is noun or verb.

- The term filter passes these filtered terms to the fuzzy inference agent.

- The fuzzy inference agent makes use of the meaningful term sets of the planned term set and the actual term set, to infer the membership degrees belonging to the CMMI ontology.

- According to the membership degrees for planned term set and actual term set, the performance decision support agent is able to measure the progress as a completed percentage of project activities.

- The project manager regularly reviews the completed progress percentage of project members to identify and document if there are significant issues and deviations from the plan, while evaluating the performance of project members.

- The project domain expert retrieves the work breakdown structure of the project planning to construct and modify the project personal ontology.

According to Lee, Wang and Chen, Ontology-based Intelligent Decision Support Agent is composed of

- a natural language processing agent,

- a fuzzy inference agent and

- a performance decision support agent[8].

The document pre-processing mechanism includes

- a Part-of-Speech (POS) tagger and

- a term filter.

The document processing mechanism tasks is presented below [51]:

- It tags the terms of evaluation reports,

- It filters meaningless terms to preserve the terms whose POS is noun or verb,

- It passes the filtered terms to the fuzzy inference agent.

The conceptual similarity factors for making analysis of the Chinese terms are

- POS similarity,

- Number similarity and

- Distance similarity [51]:

The POS similarity represents the path length between two nodes located on the tagging tree. The tagging tree is adopted to calculate the conceptual similarity in POS between any two Chinese terms [51]:

The Number similarity represents the value of the conceptual similarity between any two Chinese terms according to these three Chinese characteristics [51]:

- The more identical words in both terms in the pair, the more similar the terms are to each other in semantic meaning,

- Terms in a pair with both identical and continuous words have much greater semantic similarity than those in a pair without identical or continuous words and Terms in a pair with identical starting or ending words have a strong semantic similarity. The Distance similarity represents the semantic distance in the same layer of domain ontology for any term pair.

In the Distance similarity computation, domain experts pre-define the concept layer of CMMI ontology as five sub-layers, including the who layer, the when layer, the what layer, the where layer, and the how layer [51].

The architecture of fuzzy inference agent in the ontology-based intelligent decision support agent consists of:

- the input linguistic layer,
- input term layer,
- rule layer,
- output term layer and
- output linguistic layer.

Architecture of Fuzzy Inference Agent for the Ontology-based Intelligent Decision Support Agent is presented in Figure 19.

Layer 1 (Input Linguistic Layer):

This layer includes the Chinese term set of planned progress report, actual progress report, and CMMI ontology. The nodes in this layer directly transmit the values of POS similarity, Number similarity, and Distance similarity for the term sets to the next layer.

Layer 2 (Input Term Layer):

This layer performs the membership functions to compute the membership degrees for all terms derived from the retrieved planned term set, actual term set, and all concepts of CMMI ontology. There are three fuzzy variables, including POS similarity, Number similarity, and Distance similarity, considered in this layer for each term's property.

Layer 3 (Rule Layer):

Rule layer, represents a fuzzy rule. The main task of this layer is to be responsible for using AND operator to combine the matching degree of each fuzzy rule's condition. The rules are defined by domain expert's knowledge. These rules are presented below:

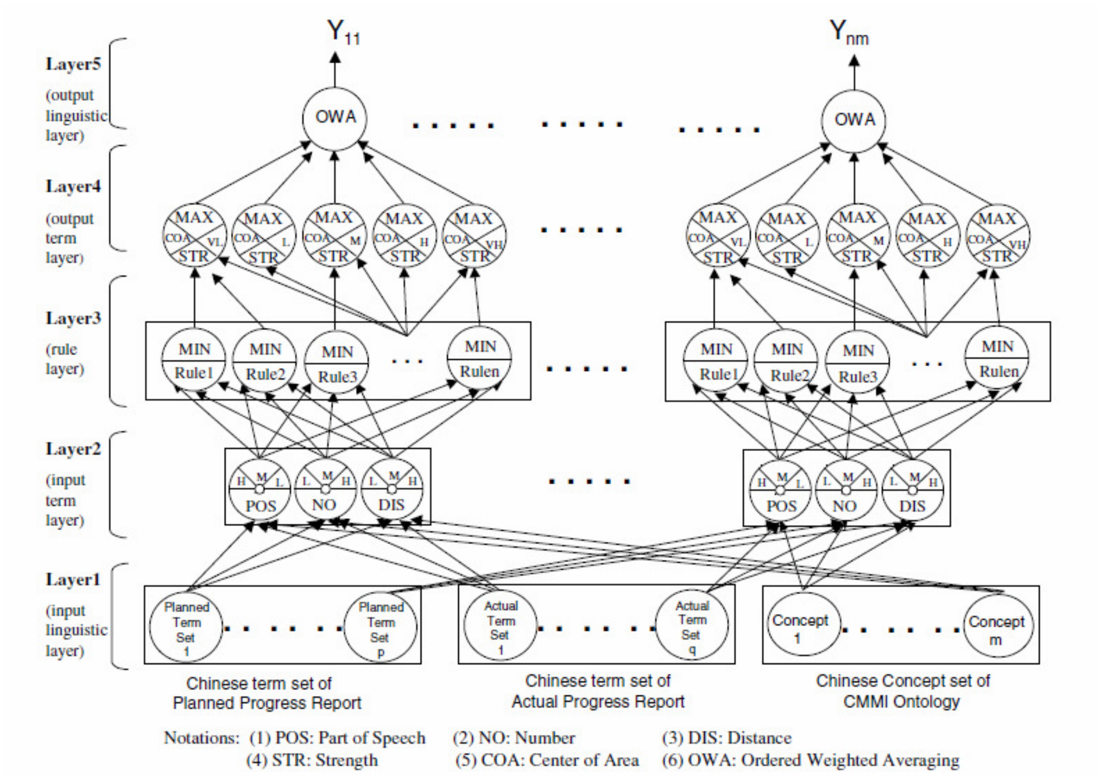

Figure 19: Architecture of Fuzzy Inference Agent for the Ontology-based Intelligent Decision Support Agent [51]

Layer 4 (Output Term Layer):

This layer performs the consequent of the fuzzy rules. Fig. 7 shows the structure of the node in the output term layer for the fuzzy rules triggering the fuzzy set Strength_Low. Take the Rule2, Rule3, Rule4, Rule5, Rule7, Rule10, Rule11, Rule13, and Rule19, as an example. They all trigger the same fuzzy set, Strength_Low [51].

Layer 5 (Output Linguistic Layer):

This layer performs the final strength value using the ordered weighted averaging aggregation operator. Membership Functions of the Fuzzy Variable are presented in Figure 20.

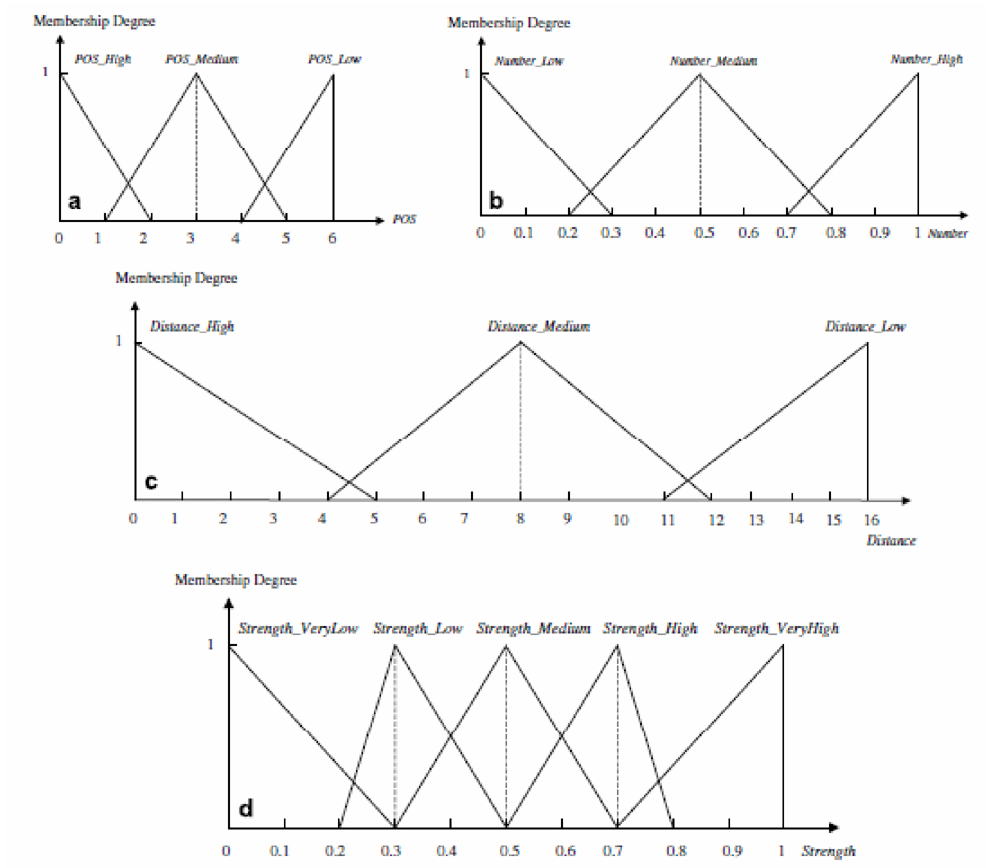

Figure 20: Membership Functions of the Fuzzy Variable: (a) POS, (b) Number, (c) Distance, and (d) Strength [52]

The Performance Decision Support Agent [51]:

The Performance Decision Support Agent measures the completed percentage of project progress based on the output of the fuzzy inference agent. The smaller the difference in evaluated percentage between project members and the ontology-based intelligent decision support agent is, the better the performance of the ontology-based intelligent decision support agent is [51].

Fuzzy inference rules for strength are presented in Table 5.

Table 5: Fuzzy Inference Rules for Strength [51]

Rule	Fuzzy Variable			
	POS	Number	Distance	Strength
1	L	L	L	VL
2	L	L	M	L
3	L	L	H	L
4	L	M	L	L
5	L	M	M	L
6	L	M	H	M
7	L	H	L	L
8	L	H	M	M
9	L	H	H	H
10	M	L	L	L
11	M	L	M	L
12	M	L	H	M
13	M	M	L	L
14	M	M	M	M
15	M	M	H	H
16	M	H	L	M
17	M	H	M	H
18	M	H	H	H
19	H	L	L	L
20	H	L	M	M
21	H	L	H	H
22	H	M	L	M
23	H	M	M	H
24	H	M	H	H
25	H	H	L	H
26	H	H	M	H
27	H	H	H	VH

VL: Very Low, L: Low, M: Medium, H: High, VH: Very High.

Structure of the Node of the Output Term Layer is presented in Figure 21.

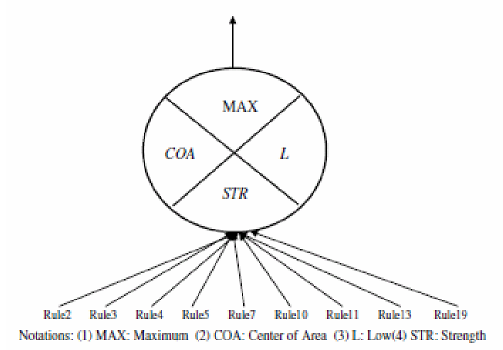

Figure 21: Structure of the Node of the Output Term Layer [51]

3.4.3 Artificial Intelligence Applications for Project Planning Process Area

In "Intelligent Estimation Agent Based on CMMI Ontology for Project Planning [39]", based on the process planning (PP) process area of CMMI, Lee, Wang, Yan, Lo, Chuang, and Lin propose an intelligent estimation agent based on CMMI ontology for project planning.

Lee, Wang, Yan, Lo, Chuang and Lin propose a novel structure of the domain ontology different from their previous work. The structure of domain ontology used in this work is presented in Figure 22.

The structure of ontology used in this work is composed of three layers:

- Concept Layer,
- Relation Layer,
- Instance Layer.

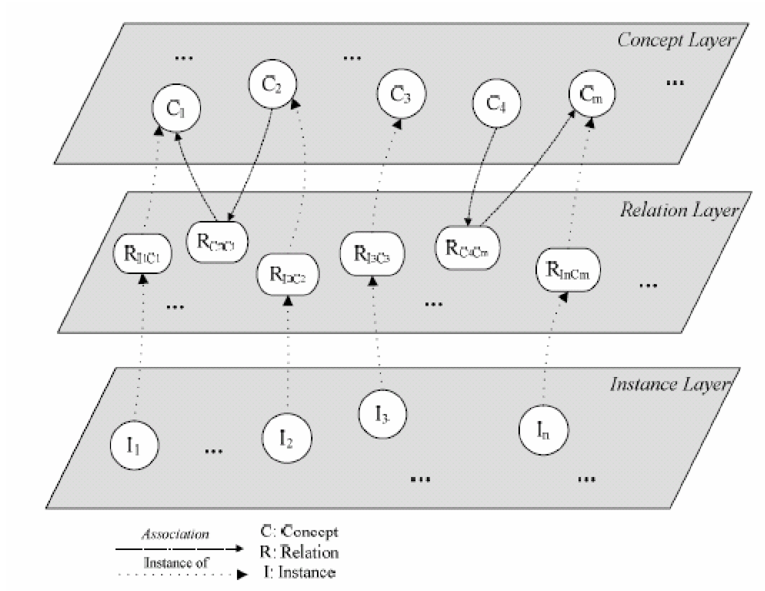

Figure 22: Structure of the Domain Ontology for an Intelligent Estimation Agent [39]

There are many concepts, such as “C1 ,C2 , ...,Cm ” defined in the concept layer. In the relation layer, each relation represents the inter-conceptual relations of the domain ontology. There are two types of relations used in the proposed ontology. These are “association” and the “instance-of”. An association relation represents the inter-relation among concepts. “Instance-of” represents the relation between a concept and an instance. The instance layer comprises some instances like “ I1 , I 2 , ..., I n [39].”

Lee, Wang, Yan, Lo, Chuang and Lin apply the structure of the domain ontology, shown in Figure 23, to project planning according to Table 6.

Table 6 lists the names of the concepts, instances, and relations of the project planning ontology [39]. For example, there is an instance-of relation between the concepts “Resources” and “Needed knowledge and skill” called “Include.” The association relation between the concept “Risks” and the instance “Risk impacts” is “Identify.”

Table 6: Practice to Goal Relationship of Project Planning

Specific Goal	Specific Practice
Establish Estimates	Estimate the Scope of the Project
	Establish Estimates of Work Product and Task Attributes
	Define Project Life Cycle
	Determine Estimates of Effort and Cost
Develop a Project Plan	Establish the Budget and Schedule
	Identify Project Risks
	Plan for Data Management
	Plan for Project Resources
	Plan for Needed Knowledge and Skills
	Plan Stakeholder Involvement
	Establish the Project Plan
Obtain Commitment to the Plan	Reviews Plans that Affect the Project Reconcile work and Resource Levels
	Obtain Plan Commitment

Figure 23 shows an example of the domain ontology on project planning [39]:

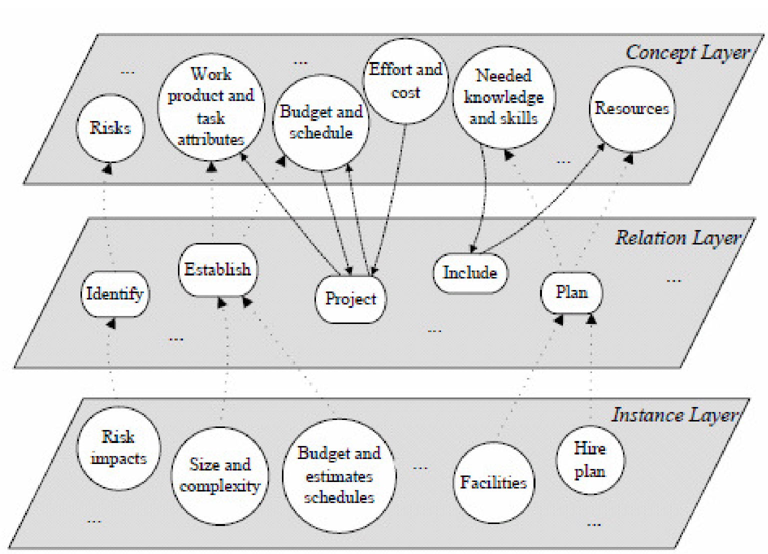

Figure 23: Example of the Domain Ontology on Project Planning [39]

Table 7: Names of the Concepts, Instances, and Relations of the Project Planning Ontology [39]

Concept Name	**Instance Name**	**Instance of Relation Name**
Scope	Work breakdown structure	Estimate
Work product and task attributes	Size and complexity	Establish
Life cycle	Project life cycle phases	Define
Effort and Cost	Effort and cost estimates	Determine
Budget and schedule	Budget and estimates of schedules	Establish
Risks	Risk impacts	Identify
Data management	Data requirements lists	Plan
Resources	Facilities	Plan
Needed Knowledge and skills	Hire plan	Plan
Stakeholder involvement	Involvement plan	Plan
Project plan	Overall project plan	Establish
Project-affecting plan	Review records	Review
Resource levels	Revised project plan	Reconcile
Commitment	Commitment documents	Obtain
Concept Name Association	Concept Name	Relation Name
Risks	Project-affecting plan	Reconcile
Life cycle	Project plan	Determine
Work product and task attributes	Effort and cost	Project
Resources	Needed knowledge and skills	Include
Work product and task attributes	budget and schedule	Project

Figure 24 shows a CMMI-based Project Planning Ontology [39]:

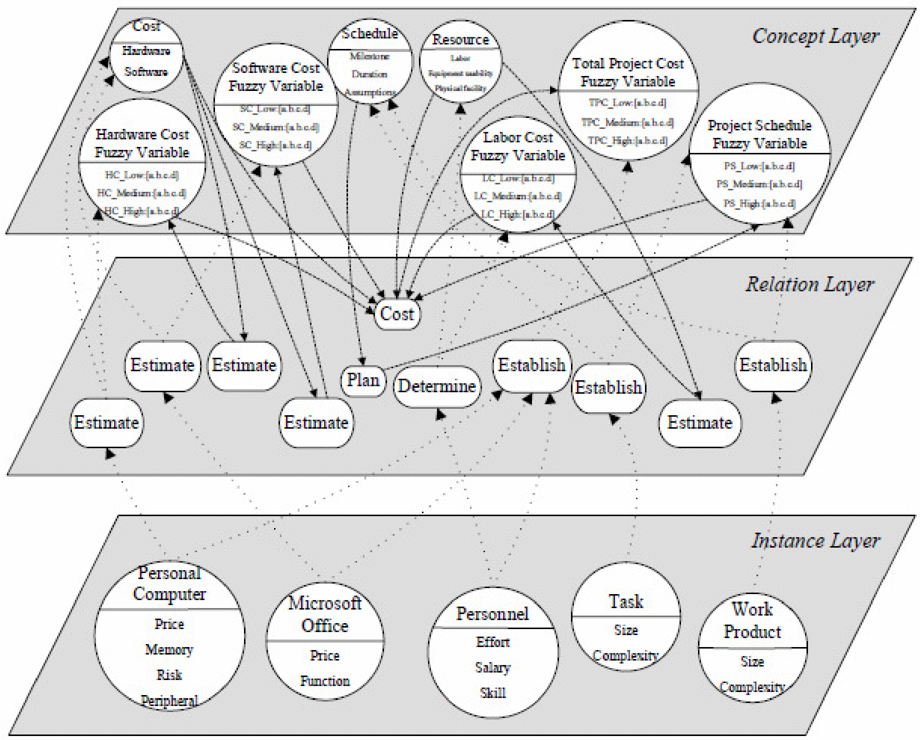

Figure 24: CMMI-based Project Planning Ontology [39]

Based on the structure of the domain ontology presented above, a CMMI-based project planning ontology applied to the proposed ontology-based intelligent estimation agent is built up [39].

Project cost estimation is very important factor for project management. Cost estimation reduces the risk of project failure and improves the feasibility of a project. In this work, Lee, Wang, Yan, Lo, Chuang, and Lin propose an ontology-based intelligent estimation agent, including a CMMI-based project planning ontology and a fuzzy cost estimation mechanism, for the total project cost estimation. The proposed agent combines the fuzzy inference, Capability Maturity Model Integration (CMMI), and the ontology. The fuzzy cost estimation mechanism infers the total project cost and stored the results to the project estimation repository [39].

Figure 25 shows the structure of the ontology based intelligent estimation agent for the total project cost, including a CMMI-based project planning ontology and a fuzzy cost estimation mechanism[39].

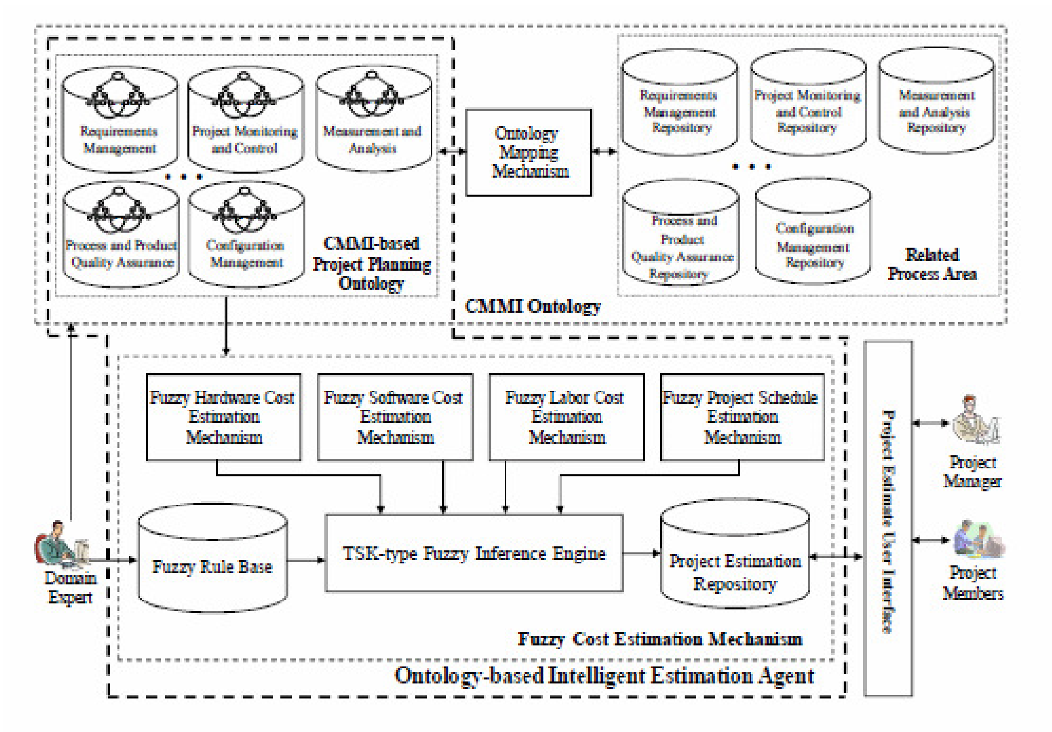

Figure 25: Structure of the Ontology-based Intelligent Estimation Agent [39]

System architecture steps are presented below [39]:

- Domain experts predefine the CMMI ontology and the fuzzy rule base.

- The ontology mapping mechanism generates the corresponding CMMI-based project planning ontology after mapping the information stored in the requirements management repository, project monitoring and control repository, measurement and analysis repository, process and product quality assurance repository, and configuration management repository.

- The CMMI-based project planning ontology consists of requirements management ontology, a project monitoring and control ontology, a measurement and analysis ontology, a process and product quality assurance ontology, and configuration management ontology.

- The fuzzy hardware cost estimation mechanism retrieves the information stored in the CMMI-based project planning ontology and then uses a fuzzy model to infer the estimates of the hardware cost.

- The fuzzy software cost estimation and the fuzzy labor cost estimation mechanisms run the similar functions for the estimation of the software cost and labor cost, respectively.

- The total project cost is inferred using a fuzzy inference engine according to the predefined fuzzy rule base and the output of the fuzzy Project schedule estimation mechanism.

- The total project cost is stored in the project estimation repository and presented to the project manger and project members through a provided user interface.

In this way, we can decrease the complexity of estimating cost and the workload of the estimators significantly.

Fuzzy cost estimation mechanism includes:

- fuzzy hardware cost estimation mechanism,

- a fuzzy software cost estimation mechanism,

- a fuzzy labor cost estimation mechanism,

- a fuzzy project schedule estimation mechanism,

- a fuzzy rule base,

- a fuzzy inference engine and

- a project estimation repository.

The fuzzy hardware cost estimation mechanism infers the hardware cost of the project according to the quantity of

- the high-cost,
- medium-cost and
- low-cost hardware.

There are three input fuzzy variables,

- low-cost hardware quantity (LCHQ),
- medium cost hardware quantity (MCHQ) and
- high-cost hardware quantity (HCHQ).

Additionally, one output fuzzy variable, hardware cost (HC) is adopted for the fuzzy hardware cost mechanism.

The linguistic terms adopted for LCHQ, MCHQ, and HCHQ fuzzy variables are Small, Medium, and Large.

There are five linguistic terms, including Very Low, Low, Medium, High, and Very High, in HC fuzzy variable.

A complete set of fuzzy rules for the estimation of the hardware cost is shown in Table 8.

The definitions of the fuzzy inference for the fuzzy software cost estimation mechanism and the fuzzy labor cost estimation mechanism are similar to the ones of the fuzzy hardware cost estimation mechanism.

Table 8: Fuzzy Rules for Hardware Cost [39]

Rule	Input Fuzzy Variable			Output Fuzzy Variable
	LCHQ	MCHQ	HCHQ	HC
1	Small	Small	Small	Very Low
2	Small	Small	Medium	Low
3	Small	Small	Large	Low
4	Small	Medium	Small	Low
5	Small	Medium	Medium	Low
6	Small	Medium	Large	Medium
7	Small	Large	Small	Low
8	Small	Large	Medium	Medium
9	Small	Large	Large	High
10	Medium	Small	Small	Low
11	Medium	Small	Medium	Low
12	Medium	Small	Large	Medium
13	Medium	Medium	Small	Low
14	Medium	Medium	Medium	Medium
15	Medium	Medium	Large	High
16	Medium	Large	Small	Medium
17	Medium	Large	Medium	High
18	Medium	Large	Large	High
19	Large	Small	Small	Low
20	Large	Small	Medium	Medium
21	Large	Small	Large	High
22	Large	Medium	Small	Medium
23	Large	Medium	Medium	High
24	Large	Medium	Large	High
25	Large	Large	Small	High
26	Large	Large	Medium	High
27	Large	Large	Large	Very High

CHAPTER 4

SUMMARY and CONCLUSION

Recently, as the process improvement works gain more acceptance, miscellaneous process improvement models are being developed. With increasing competition, organizations seek to improve software quality by improving their management, support and engineering processes. Capability Maturity Model Integration (CMMI) is the most popular process improvement maturity model for the development of products and services. This model contains best practices that address development and maintenance activities applied to product and services.

CMMI focuses on improving processes in an organization. It aims a powerful process improvement mechanism using best practices and main goal is organizational excellence in software development.

In this thesis, we first review CMMI concepts and CMMI-DEV v.1.2 process areas. We next survey AI techniques and investigate how to improve our CMMI processes using those techniques. For this purpose, we review the most recent artificial intelligence techniques used in conjunction with CMMI.

As a result of literature research, we saw that some CMMI processes have been successfully applied using AI techniques.

These CMMI Process Areas are presented below:

- Product and Process Quality Assurance (PPQA)

- Project Monitoring and Control (PMC)
- Project Planning (PP)
- Configuration Management (CM)
- Measurement and Analysis (MA)

As a result of this survey, we have seen that we can use similar artificial intelligent techniques to improve our processes. Specially, using CMMI based ontologies and some intelligent agents (fuzzy inference agent, intelligent decision support agent, natural language processing agent, intelligent estimation agent etc.), we can improve our CMMI processes.

Finally, we have observed that selected Level 2 process areas are suitable for implementation with the similar artificial intelligence techniques and we can apply these techniques to other CMMI process areas (Quantitative Project Management, Requirements Development, Organizational Process Focus, Organizational Process Performance, Risk Management etc.) for future research.

REFERENCES

1. **AHERN, D.M., CLOUSE A., TURNER, R.** et. al (2008), *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*, Third Edition, Addison Wesley.

2. **MUTAFELLIJA, B., STROMBERG, H.** (2009), *Process Improvement with CMMI v.1.2 and ISO Standards*, CRS Press, Taylor Francis Group.

3. **FRENZEL, A.** (2005), *Ontology-based Architecture to support Software Process Improvement*, Brandenburg University of Technology Cottbus, Faculty of Mathematics, Natural Science and Computing Internet-Technologies.

4. CMMI Product Team (2006), *CMMI for Development, Version 1.2,* Carnegie Melon University, Software Engineering Institute.

5. **PIKKARAINEN, M., MANTYNIEMI A.** (2006), *An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies*, SPICE 2006 conference, Luxemburg.

6. **SHRUM, S., PHILLIPS, M.**(2004), *CMMI Overview for Executives*, Carnegie Mellon University, Software Engineering Institute, Pittsburg.

7. **PERSSE, J.R.** (2006), *Process Improvement Essentials*, O'Reilly.

8. **AHERN D.M., ARMSTRONG, J., CLOUSE A.,** et. al. (2005), *CMMI-SCAMPI Distilled Appraisals for Process Improvement*, Addison Wesley Professional.

9. www.sei.cmu.edu/about/press/CMMI-ACQ/CMMIhistory.pdf.

10. http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration.

11. **KULPA M.K., JOHNSON, K.A**. (2008), *Interpreting the CMMI, A Process Improvement Approach, Second Edition*, CRC Press.

12. **CHRISSIS M.B., KONRAD, M., SHRUM, S**. (2007), *CMMI Second Edition, Guidelines for Process Integration and Product Improvement,* SEI Series in Software Engineering, Addison Wesley.

13. **BARSTOW, D**.(1987), *Artificial Intelligence and Software Engineering,* Schlumberger-Doll Research Old Quarry Road, ACM.

14. **DICK, S., KANDEL, A.** (2005), *Computational Intelligence In Software Quality Assurance*, Series In Machine Perception Artificial Intelligence, Volume 63, World Scientific, CRC Press.

15. **KETO, H., PALOMAKİ, J.** (2007), *Applying the Process-Ontological Model in Practice*, Tampere University of Technology/Pori Department of Information Technology, Finland.

16. **BRIAND, L.C**. (2002), *On the many ways Software Engineering can benefit from Knowledge Engineering*, Software Quality Engineering Laboratory, Carleton University, Department of Systems and Computer Engineering, ACM SEKE, Italy.

17. **PARTRIDGE, D.** (1998), *Artificial Intelligence and Software Engineering – Understanding the Promise of the Future*, AMACOM, American Management Association.

18. **BEZDEK, J.C**.(1994), *What is computational intelligence?*, Computational Intelligence: Imitating Life, New York: IEEE Press, 1994

19. **DUCH, W**. (2007), *What is Computational Intelligence and what could it become*?, Department of Informatics, Nicolaus Copernicus University, and School of Computer Engineering, Nanyang Technological University, Singapore.

20. **GUARINO, N., GIARETTA, P.** (1995), *Ontologies and knowledge bases: toward a terminological clarification*, in Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, N. Mars (Editor), 25-32, ISO Press: Amsterdam, The Netherlands.

21. **DAVIS, S.D., WALTON, T.B**.*(2004)*, *Engineering Knowledge*, ACMSE '04, Huntsville, Alabama, USA.

22. **WHITE, K.F., LUTTERS, W.G.** (2007), *Structuring Cross-Organizational Knowledge Sharing,* Department of Information Systems, UMBC, ACM, Florida, USA.

23. **CHANDRASEKARAN, B., JOSEPHSON, J.R., BENJAMINS, V.R.** (1999), *What are ontologies, and why do we need them*?, IEEE Intelligent Systems 14, 1 20–26.

24. **MIZOGUCHI, R.** (1998), A *step towards ontological engineering*, Proceedings of The 12th National Conference on AI of JSAI, 24–31.

25. **DEVEDZIC, V.** (2002), *Understanding Ontological Engineering*, University of Belgrade in Yugoslavia, Vol. 45, No. 4 Communications Of The ACM.

26. **SWARTOUT, W., TATE, A.** (1999), *Ontologies*, Guest Editors' Introduction, IEEE Intelligent Systems 14, 1, Special Issue on Ontologies, 18–19.

27. **LEE, C.S.** (2005), *Introduction to the Applications of Domain Ontology*, Department of Computer Science and Information Engineering, National University of Tainan, Taiwan.

28. **HAIHUA, J.H., LIU, Y.C., JIN, M.** (2007), *A Framework of Ontology-supported Knowledge Representation in Software Process*, School of Computer Science & Engineering, Beihang University, China.

29. **BIRK, A., SURMANN, D., ALTHOFF, K.**(1999), *Applications of knowledge acquisition in experimental software engineering*. Proc. of the 11th European Workshop on Knowledge Acquisition, Modeling, and Management, pp. 67- 84.

30. **FALBO, A**.(2002), *Managing Software Process Knowledge*, Proc. of the International Conference on Computer Science, Software Engineering, Information Technology, e- Business, and Applications, Foz do Iguazu, Brazil, June.

31. **SCOTT, R.H.** (2003), Tool *support for experience based software development methodologies*, Advances in Computers, 59:29-82.

32. **LIAO, L., YUZHONG, Q., H., LEUNG, K.N.(***2005), A software process ontology and its application*, Proc. of the 4th International Semantic Web Conference, Galway, Ireland.

33. **SCACCHI, W.** (2000), *Understanding software process redesign using modeling, analysis and simulation*, Software Process Improvement and Practice, 5:183-195.

34. **NOUR, P.** (2003), *Ontology-based Retrieval of Software Engineering Experiences*, Master Thesis, Department of Computer Science, University of Galary.

35. **CERAVOLO, P., DAMIANI, E., MARCHESI, M.** et. al. (2003), *An Ontology-based Process Modeling for XP*, Proc. of the Tenth Asia-Pacific Software Engineering Conference, pp. 236-242.

36. **ABECKER, A., BERNARDI, A., HINKELMANN, K.,** et. al. (1998), *Toward technology for organizational memories*, IEEE Intelligent systems, 13: 40-48.

37. **LEE, C.S., WANG, M.H.** (2007), *Recent Research on Ontology Applications*, Department of Computer Science and Information Engineering, National University of Tainan, Taiwan, ISIS.

38. **WANG, M.H., LEE, C. S.** (2007), *An Intelligent Fuzzy Agent based on PPQA Ontology for Supporting CMMI Assessment*, Nat. Univ. of Tainan, Tainan, Fuzzy Systems Conference.

39. **LEE, C.S., WANG, M. H., CHEN, J.J.** et. al. (2008), *Intelligent Estimation Agent Based on CMMI Ontology for Project Planning*, IEEE International Conference on Systems, Man and Cybernetics, SMC.

40. **LEE, C.S., WANG, Y.C., LIU, W.M.** et. al. (2007), *CRM Ontology Based on CMMI Project Planning for Business Applications*, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong.

41. **HUANG, S.J., HAN, W.M**.(2006), *Selection Priority of Process Areas Based on CMMI Continuous Representation*, Science Direct, Information & Management.

42. **LIU, X. F., KANE, G., BAMBROO, M.**(2006), *An Intelligent Early Warning System for Software Quality Improvement and Project Management*, Science Direct, The Journal of Systems and Software.

43. **LEE, S.H., CHOI, H.J., KIM, S.** (2009), *Analysis of Ontology Development Methodology Based on and CMMI Level 4*, School of Engineering, Information and Communications University, Korea.

44. **VIYANASUNDARAM, B., SRIVATSA, S.K.** (2007), *Software Quality in Artificial Intelligent System*, Information Technology Journal 6, Asian Network for Scientific Information.

45. **DAYAN, R., EVANS, S.** (2006), *KM Your Way to CMMI*, Journal of Knowledge Management.

46. **WANG, M.H., LEE, C.S.** (2008), *An Intelligent PPQA Web Services for CMMI Assessment*, Eighth International Conference on Intelligent Systems Design and Applications.

47. **SHARIFLOO, A.A., MOTAZED, Y., SHAMSFARD, M.**, et. al. (2008), *An Ontology for CMMI-ACQ Model*, Information and Communication Technologies: From Theory to Applications.

48. **HUNG, Y.H, CHOU, S.T., CHEN, Y.S.** (2005), *How can we assess Knowledge Management? Constructing a Holistic Assessment Framework of KM*, European Conference on Knowledge Management.

49. **LEE, C.S., JIAN, Z. W., HUANG, L. K.** (2005), *A fuzzy ontology and its application to news summarization*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 35, no. 5, pp. 859-880.

50. **LEE, C.S., KAO, Y.F., KUO, Y. H.,** et. al. (2006), *Automated ontology construction for unstructured text documents*, Data & Knowledge Engineering.

51. **LEE, C.S., WANG, M.H., CHEN, J.J.** (2007), *Ontology-based Intelligent Decision Support Agent for CMMI Project Monitoring and Control*, International Journal of Approximate Reasoning.