Building New Tool for Software Project Planning Based on Meta-modeling Concepts and CMMI Meta-Standards

Husam Bofaoor[1], Bassem KOSAYBA[2]

Abstract – CMMI (Capability Maturity Model Integration) for development is a quality system targeting to improve the software development processes. It allows to measure organization maturity level and to improve its capacity in software development.

The software industry became a vital economic sector that employs a wide range of people. A quality assurance system becomes a real need to ensure the quality of software and to increase abilities of software companies to meet the requirements of customers. MDE (Model Driven Engineering) is an approach that aims to automate the use of models in order to build more complex systems. Our paper aims to use a MDE approach to develop tools supporting any software company to adjust their software development process and then to meet the CMMI goals.

In this paper, we suggest a new structure of software development process in several models. We look at the whole software development process as a complex system that we should manage and monitor. We show in this paper how to specify a meta-model for each step of this system. So, we can take the system different functionalities to design it through the definition of its related models. After that, our MDE framework builds the system through the model transformations. These transformations permit the integration of system different functionalities. We applied our approach to design new tool for software project planning that cope with CMMI meta-standards.

One of the advantages of our approach is that it is very resilient and it can support the software process development for several software companies and fit rapidly the changes in their software development process.

Keywords:

CMMI (Capability Maturity Model Integration), MDE (Model Driven Engineering), GMF (Graphical Modeling Framework).

I. Introduction

Software companies depend more and more on enhancing and improving their work procedures and processes to achieve their quality objectives.

CMMI offers to software companies the ability to improve their software quality system by exercising specific practices related to specific and generic goals to reach certain required maturity level in software development process [1][2].

CMMI is a quality standard whose target is to improve the software development process. It allows to measure the organization maturity level and to improve its capacity of software development. This meta-standard directs software companies to establish theirs processes in order to develop high quality software products. CMMI defines how to enhance and evaluate processes among a set of activities called “Process Areas” which fully support software development operations (like Project Planning and Management, etc.) [1].

CMMI meta-model allows software companies to measure their maturity level and then to determine the needed set of best practices to improve their capacity of software development, and to enhance quality of their processes and products [2].

So, CMMI meta-model offers:

- □ Defined structure of practices to follow in order to reach a selected maturity level.
- □ Documented procedures and processes to enhance selected maturity level.
- □ Set of control and evaluation processes to validate improvements.

It is important to note that CMMI identifies a set of practices for a software project development and gives procedure examples to follow [3], but it doesn't guarantee the complete success of software projects.

We should also remember that models are abstractions of systems. High levels of abstraction allow easier understanding and handling of systems. Moreover, new approaches like the MDE [4] aim to automate the use of models. MDE encourages the identification and the separation between the different system functionality. MDE structures the application development in several models and model transformations. In MDE framework, we specify a meta-model for each system aspect. So, we can design separately the system different functionalities through the definition of system models [4].

After that, the MDE framework builds the whole system through the model transformations [4]. These transformations permit the integration of system different functionalities. Briefly, MDE allows us to specify a methodology for defining problems and how to go towards solutions.

In this paper, we provide a model-driven framework of software development process with respect to CMMI guidelines [1]. This framework is supported by modeling tools which are automatically produced themselves by meta-models. Each meta-model defines the concepts related to each CMMI guideline. We will discuss in the remaining parts of this paper how to meta-model some of CMMI guidelines. Then, we will show how to organize the produced framework tools.

Section 2 of this paper presents briefly CMMI maturity levels of and their related Process Areas. Section 3 extracts requirements of our tool for software project planning and management. Section 4 describes Meta-Model that we use to build our tool. Finally, in section 5, we present our implementation strategy of our suggested tool.

II. CMMI maturity levels (MLs):

CMMI for development has 22 process areas distributed into 4 categories [1]: Process Management, Project Management, Engineering and Support as it is described in the table of figure 1.

ML	**Process Management**	**Project Management**	**Engineering**	**Support**
2		-Project Planning (PP) -Project Monitoring and Control(PMC) -Supplier Agreement Management (Sam)	-Requirement Management (REQM)	-Configuration Management (CM) -Process & Product Quality Assurance (PPQA) -Measurement and Analysis (MA)
3	-Organizational Process Focus (OPF) -Organizational Process Definition (OPD) -Organizational Training (OT)	-Integrated Project Management (IPM) -Risk Management (RSKM)	-Requirements Development (RD) - Technical Solution (TS) -Product Integration (PI) -Validation (VAL) -Verification (VER)	-Decision Analysis and Resolution (DAR)
4	-Organizational Process Performance (OPP)	-Quantitative Project Management (QPM)		
5	-Organizational Innovation and Deployment (OID)			-Causal Analysis and Resolution (CAR)

Figure 1: CMMI for Development process Areas.

The CMMI maturity level 2 has 7 process areas. One of them is project planning process area.

As we know, project management phase is one of the important stages in the development of software products. Most of the software companies suffer from weaknesses in managing their software development projects to deliver products of high quality. The majority of these weaknesses are the result of bugs in work processes implementation. So, CMMI can give us a new thinking manner to establish work processes to manage software projects in order to get rid of problems and obstacles that could occur during software development process.

II.1. Scope of our work

CMMI contains five levels, each level contains many of process areas.

Every process area is supported by a set of generic and specific goals, each goal has a set of practices.

In level 2 of CMMI cover all management process in any software company, this level contains seven process areas.

Figure 2 shows an outline of the process areas related to project management and their relationships [1].

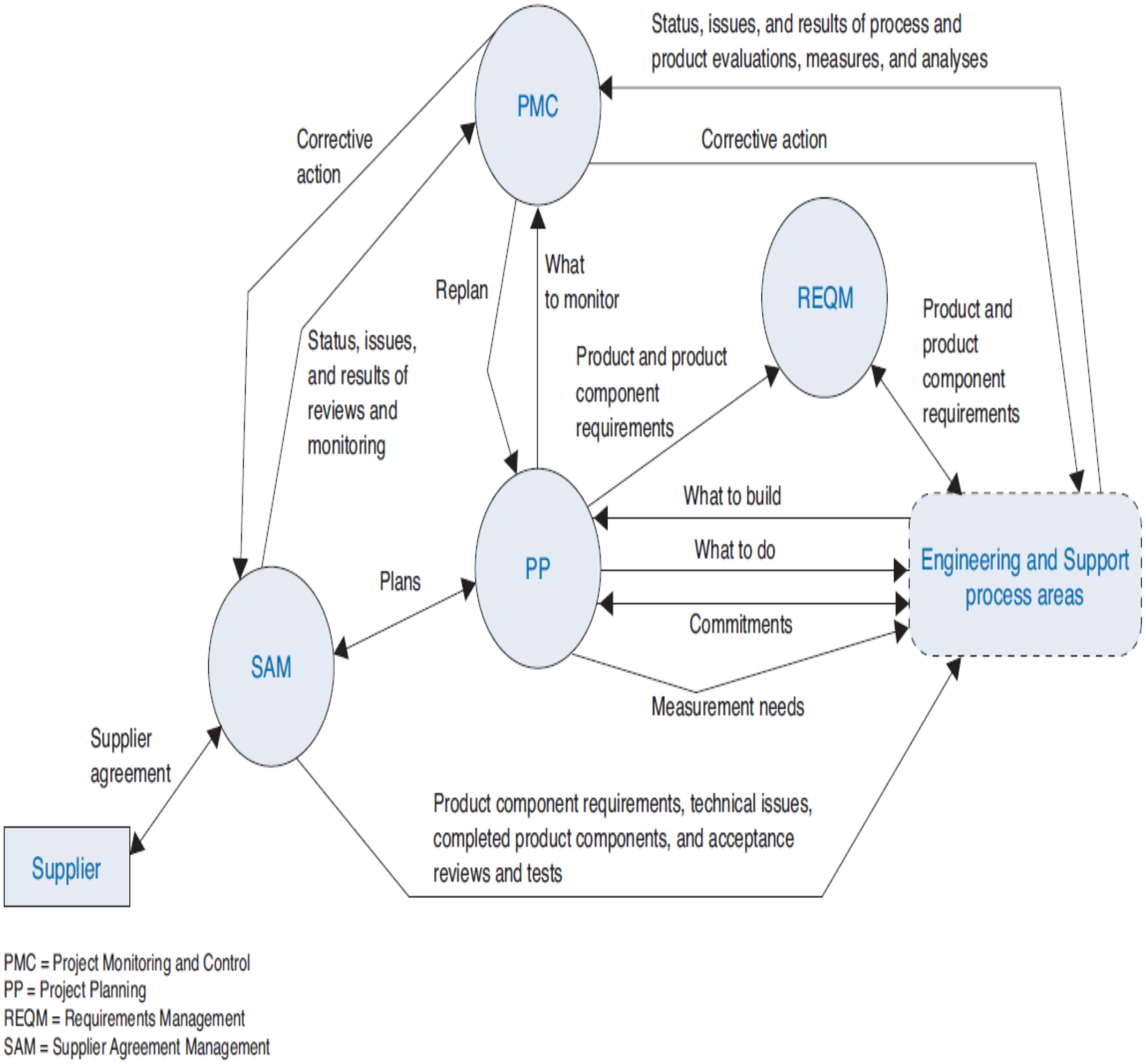

PMC
Status, issues, and results of process and product evaluations, measures, and analyses
Corrective action
Corrective action
Replan
What to monitor
REQM
Status, issues, and results of reviews and monitoring
Product and product component requirements
Product and product component requirements
What to build
What to do
Commitments
Plans
PP
Engineering and Support process areas
SAM
Measurement needs
Supplier agreement
Product component requirements, technical issues, completed product components, and acceptance reviews and tests
Supplier
PMC = Project Monitoring and Control
PP = Project Planning
REQM = Requirements Management
SAM = Supplier Agreement Management

Fig 2 Process Areas related to project Planning

II.2 Weaknesses in software project management

The process of project management may suffer from several problems, such as:

1- Delays in product delivery.
2- Misunderstanding of customer requirements.
3- Non-accurate estimation of project cost and time.
4- Lack of teamwork commitment according to software engineering principles.

To overcome these obstacles, we should introduce improvements in project management work processes and develop new work platform for project planning according to CMMI recommendations. .

III. Requirements of our tools:

The process of manufacturing of a software tool for project planning and management must be based on several important requirements [3]. These requirements are:

III.1. Work Breakdown Structure (WBS)

WBS helps in determining the list of actions that describe all project activities. These activities are not the same and vary from one project development stage to another. So, these activities must be determined carefully for each stage, to provide project manager and company administration the ability to fully control their projects. Although it is not easy to determine these activities at the beginning of the project, this task becomes more straight forward at advanced stages of project development process.

III.2. Project Schedule

Project Schedule is a timeline of project execution. It depends directly on the list of activities specified previously during WBS phase.

In addition to project activities, we have many other factors that play an essential role in determining project schedule, like:

- Start and end dates of each activity.
- which resources are assigned to perform it.
- availability dates of assigned resources.
- dependencies between activities.

Any change at the level of project activities and any change at the level of time estimation for project activities, will necessarily affect project schedule.

III.3. Estimates

Estimation process plays an essential role in project planning. This process aims to estimate cost and effort that needed to complete each activity of the project.

Estimations are determined by using many methodologies such as: Planning Poker, Function points, use-case points [3].

Estimation should be done at the level of:

1- Project activities.
2- Management, training and follow-up tasks.
3- Change requests.

For each one of these estimation operations should fallow specific steps to provide good results.

We should also take into consideration that any change in WBS project activities will affect their estimation which will in its turn affect project schedule.

III.4. Tracking and monitoring project activities

This activity is used to monitor and track all changes and projects' activities, such as: detections of non-compliances, change requests, bugs, defects… This activity aim to ensure the good study of all discovered issues related to the project and to perform the appropriate amendments as required.

IV. Using Meta-Modeling Concepts to Build our Tool:

We target to define an approach to build development processes conform to CMMI recommendations and dedicated for several companies. We want that built process to be easily modified according to the changes of company work and/or to be easily adapted to other companies. So, we create a meta-model for each adopted CMMI process area. Then, we produce modeling tools starting of the meta-models for all the desired phases. The models defined using some produced modeling tools are inputs to other produced tools. The CMMI recommendations are reflected into the built process by using meta-modeling techniques. This approach is flexible because we can easily change the process tools by introducing the changes in the meta-models that drive the development process. Then, we re-produce the modeling tools.

Now, we will define the conceptual diagrams of project planning procedures and practices as recommended by CMMI. Then, we will apply meta-modeling techniques to build our process tools as it is shown in the following sections.

IV.1. Conceptual Diagrams compatible with Project Planning Process Area

Figure 3 shows the conceptual diagram related to CMMI project planning process area. This diagram contains the following classes:

- Estimation
- Project phases
- Project schedule
- Procedures
- CMMI Pas
- PA Goals
- Goal Practices

To build a project schedule we need to define its tasks. For us, these tasks are extracted as follows: We start with the class of project phases which are the definition of key stages of the project. Each project phase is associated with class “Procedures” that represents a range of activities (Activities) to be executed sequentially and interconnected one to many. Procedures in their turn are defined by work stages containing a number of actions.

Activities depend on defined goals and practices in the CMMI Platform.

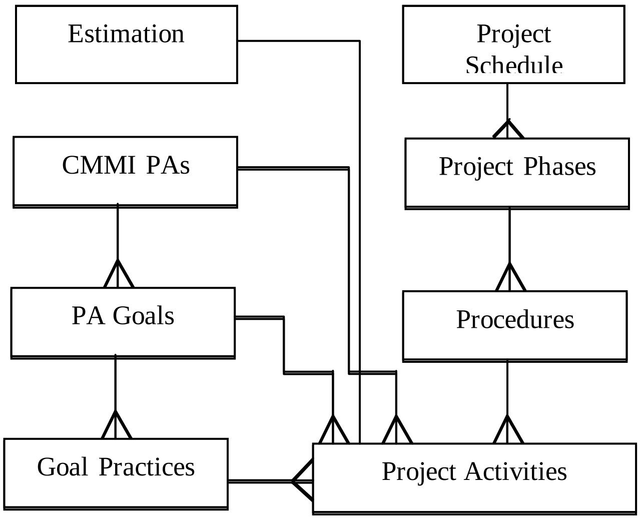


Figure 3: Conceptual diagram of CMMI project planning PA.

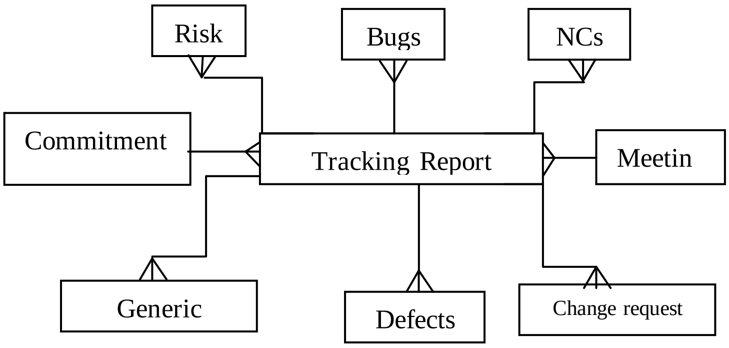


Figure 4: Conceptual diagram to generate tracking reports.

Figure 4 presents the conceptual diagram to generate tracking reports. This report should contain:

- Basic data date of the report and project stage.
- Automatically collected data about: risk, general problems, inconsistencies, deficiencies, change requests, defects.

IV.2 Specific goals of project planning

- SG 1: Establish Estimates

Estimates of project planning parameters are established and maintained [1].

Specific practice Name	Classes that apply the specific practice in our tool
Establish a top-level work breakdown structure (WBS) to estimate the scope of the project.	Project Activities
Establish and maintain estimates of work product and task attributes.	-Estimation -Procedures→Project Activity Link(1 → n)
Define project lifecycle phases on which to scope the planning effort.	-procedures -Project Phases -Project Schedule -Procedures→Project Activity Link(1 → n) -Project Phases → Procedures Link(1 → n) -Project Schedule → Project Phases Link(1 → n)
Estimate the project’s effort and cost for work products and tasks based on estimation rationale.	Estimation

- SG 2 Develop a Project Plan:

A project plan is established and maintained as the basis for managing the project [1].

Specific practice Name	Classes that apply the specific practice in our tool
Establish and maintain the project’s budget and schedule.	-Project Schedule -Estimation
Identify and analyze project risks.	-Risk
Plan for the management of project data.	-CMMI PAs →Project Activity Link(1 → n) -PA Goals →Project Activity Link(1 → n) -Goal Practices →Project Activity Link(1 → n)
Plan for resources to perform the project.	Schedule
Plan for knowledge and skills needed to perform the project.	- Project Schedule - Procedures
Plan the involvement of identified stakeholders.	- Project Schedule - Project Activity
Establish and maintain the overall project plan.	- Project Schedule - Tracking Report

1 n

- SG 3 Obtain Commitment to the Plan

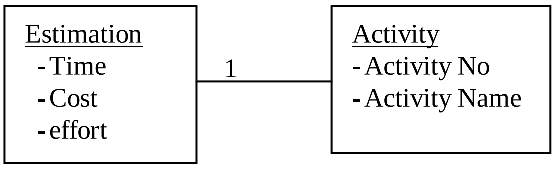
Commitments to the project plan are established and maintained [1].

Specific practice Name	Classes that apply the specific practice in our tool
Review all plans that affect the project to understand project commitments.	-Project Schedule -Estimation
Adjust the project plan to reconcile available and estimated resources.	-Generic Issues - Meeting
Obtain commitment from relevant stakeholders responsible for Performing and supporting plan execution.	-Commitment

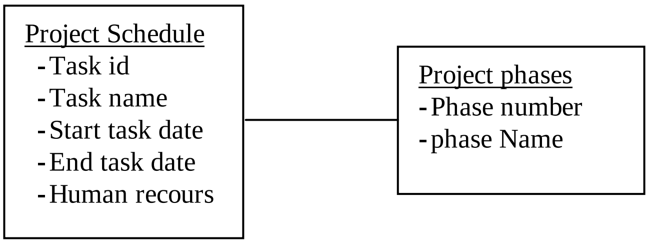
V. Implementation of our tools

We use Eclipse Graphical Modeling Framework (GMF) to produce our CMMI tools. GMF produces graphical modeling tools starting from EMF (Eclipse Modeling Framework) meta-models [5]. So, we have to express our business requirements as EMF meta-models. Then, we are explaining sub relationships and attributes between classes that we need it to build our tools.

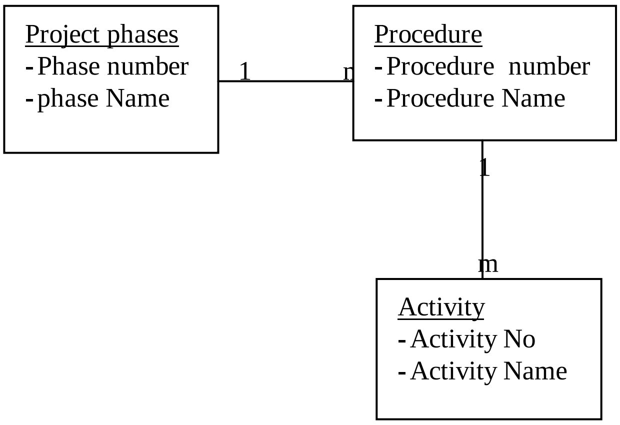
- In the project planning we defined class activity, this class relation with class estimation by one to one, in class estimation we define three attributes, these attributes are time, cost and effort.



- Also we defined class Project schedule, this class contains attributes, these attributes are task id, task name, start task date, end task date and human resources. This class relation one to many with class Project phases, the project phases contains attributes, these attributes are phase number and phase name.



- We define class Procedure, we define two attributes in this class, these attributes are procedure number and Procedure name. The class Procedure relations one to many with class Project Activities, also relations with class Project phases by many to one.



V.1. How transform diagrams to generate code

First we used hierarchical diagram presented in figure 5 to handle any CMMI process area.

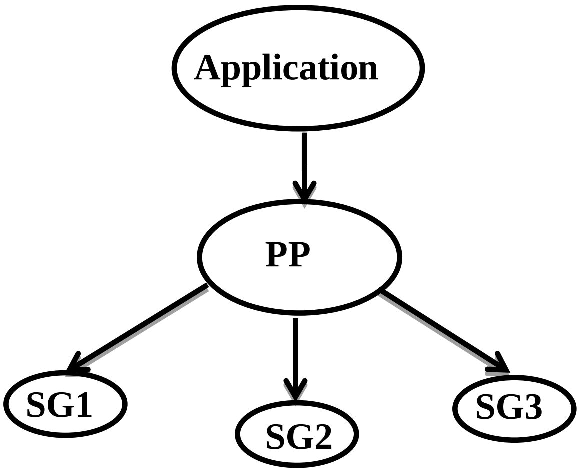
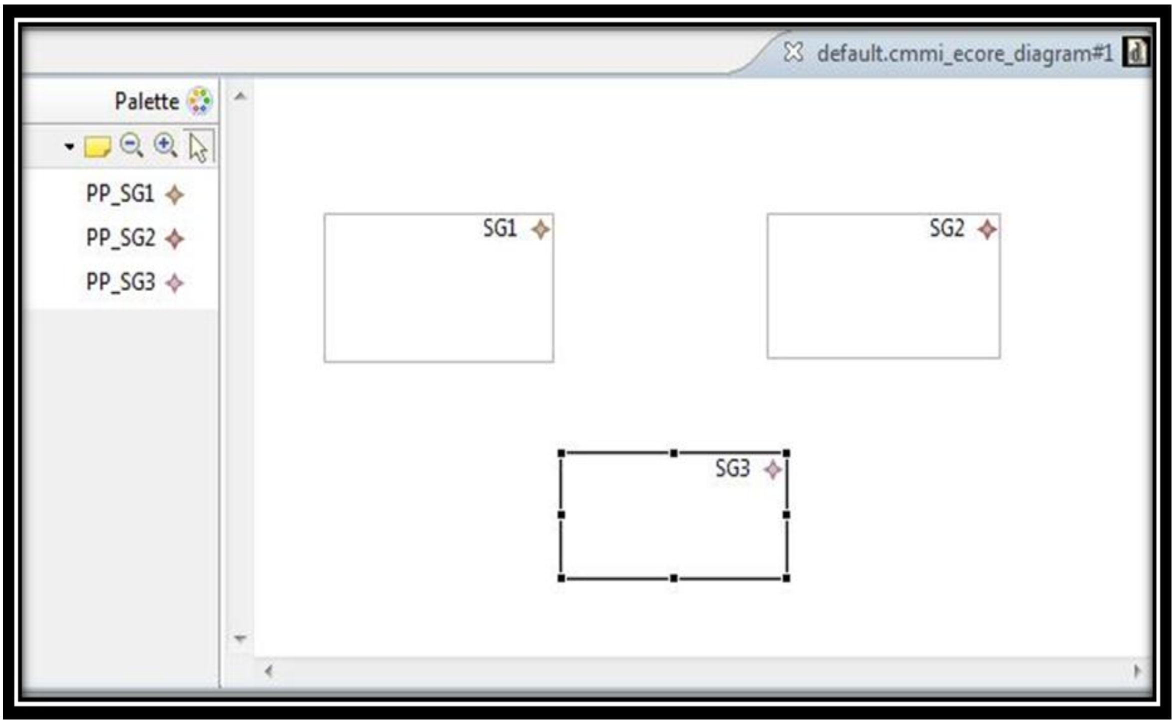


Figure 5: Hierarchical diagram to handle a project planning PA.

We show in the following sections how we did to integrate GMF with CMMI meta-model previously presented.

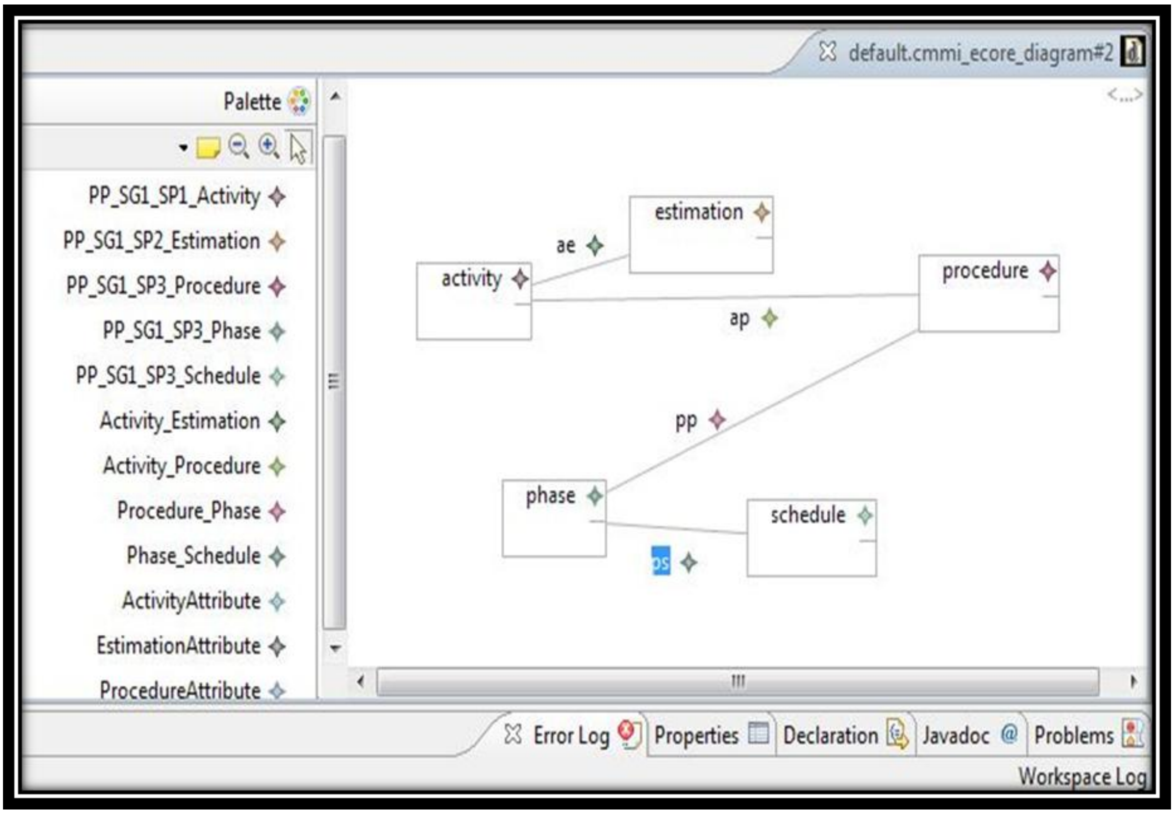
so, we will be able to build our special model according to CMMI meta- model taking into account all constraints in this meta- model.

we can drag and drop any specific goal we need.

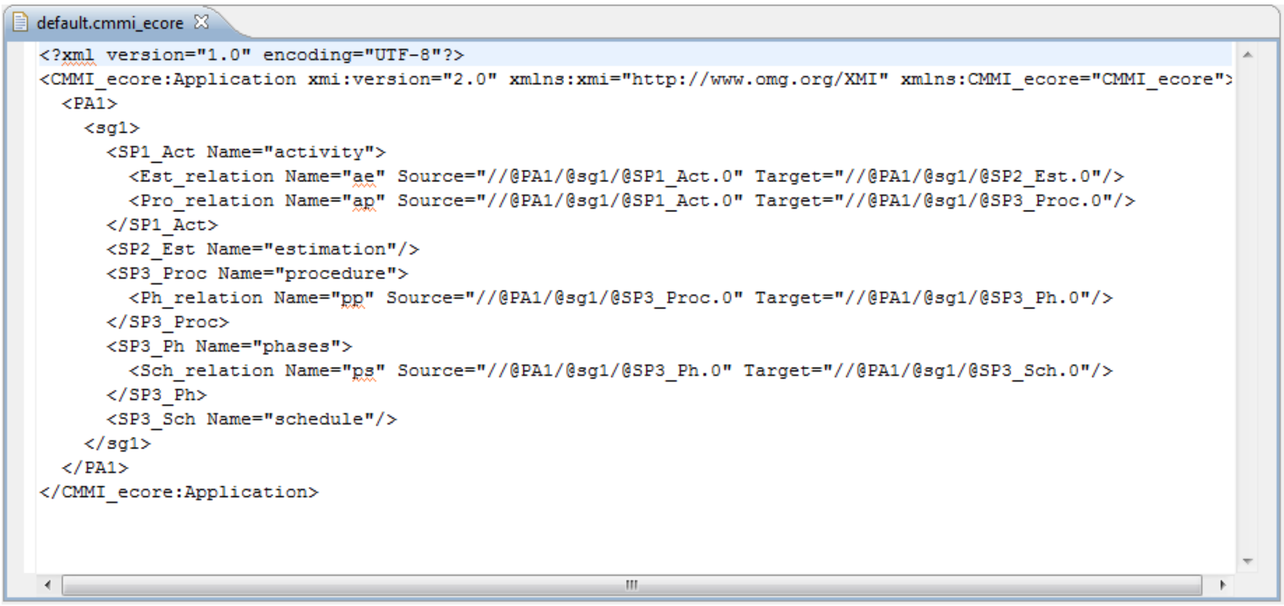


`

By double click on SGs we will have PP_SG1 execution interface. So, we can drag and drop any class related to any attribute and any relation we need to take into account for meta model constraints.



Finally, after building our special model we can a XML file reflecting this model and we can use it to generate source code of it:



```
<?xml version="1.0" encoding="UTF-8"?>
<CMMI_ecore:Application xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:CMMI_ecore="CMMI_ecore">
  <PA1>
    <sg1>
      <SP1_Act Name="activity">
        <Est_relation Name="ae" Source="//@PA1/@sg1/@SP1_Act.0" Target="//@PA1/@sg1/@SP2_Est.0"/>
        <Pro_relation Name="ap" Source="//@PA1/@sg1/@SP1_Act.0" Target="//@PA1/@sg1/@SP3_Proc.0"/>
      </SP1_Act>
      <SP2_Est Name="estimation"/>
      <SP3_Proc Name="procedure">
        <Ph_relation Name="pp" Source="//@PA1/@sg1/@SP3_Proc.0" Target="//@PA1/@sg1/@SP3_Ph.0"/>
      </SP3_Proc>
      <SP3_Ph Name="phases">
        <Sch_relation Name="ps" Source="//@PA1/@sg1/@SP3_Ph.0" Target="//@PA1/@sg1/@SP3_Sch.0"/>
      </SP3_Ph>
      <SP3_Sch Name="schedule"/>
    </sg1>
  </PA1>
</CMMI_ecore:Application>
```

VI. Conclusion and Future Prospects

We proposed in this paper a resilient manner to support the CMMI standard. This manner may be adapted according to the company needs. We can easily add new phase or modify any phase concepts used in defining the software product model.

We experienced our approach on the Project planning process area in the CMMI maturity level 2. So, we expressed the Project Management CMMI Process Area in several meta-models. There is an ordering relation between these meta-models. Also, we produced graphical modeling tools starting from these meta-models. We used GMF to generate the graphical tools

Exact description of work processes helps us to build appropriate class diagrams, and thus build the application that meets all the requirements of the procedure.

References

[1] Mary Beth Chrissis And Mike Konrad And Sandy Shrum, CMMI For Development, Version 1.2. CMMI Second Edition Guidelines for Process Integration and Product Improvement.

[2] Ahern, Dennis M.; Clouse, Aaron; & Turner, Richard. CMMI, Distilled: A Practical Introduction to Integrated Process, Improvement, Second Edition. Boston: Addison-Wesley,2003.

[3] SEI 2006b CMMI Product Development Team. SCAMPI v1.2, Standerd CMMI Appraisal Method for Process Improvement, version 1.2: Method Definition Document.

[4] The Essence of Multilevel Metamodeling, by Colin Aktinson and Thomas K¨uhne.

[5] The Eclipse Graphical Editor Framework (GEF), Pearson Education, by Dan Rubel, Jaime Wren, Eric Clayberg, 2012.

Authors' information

1. **Husam bofaoor** received his Engineering degree in Damascus University, faculty of information technology engineering, branch of Systems and networks computer in 2009, and study up to now, Master Management Technology (MTM) In Syrian Virtual University.

2 Bassem KOSAYBA PhD degree in Software Engineering (Model Driven Engineering) from USTL (University of Sciences and Technology of Lille) in 2006.
Master Degree in Parallel and Distributed Systems from USTL in 2001.
Master Degree in Computer and Control from Damascus University in 1999.
Engineering Degree in Electronics from Damascus University in 1997.
He is now an associated professor at Information Technology Engineering Faculty, Damascus University. His research interests include Software Engineering (MDE) and Distributed Systems.