UML Modeling of Five Process Maturity Models

Version 1

LQL-2003-TR-02

2003

Simon Alexandre

Naji Habra

CETIC - FUNDP 2003

Table of contents

Table of figures

1 Introduction

This technical report presents an UML class diagram modeling of five process improvement models. The main objective is to propose a common graphical representation of different models to facilitate their comprehension and comparison.

The five selected models are the Software Capability Maturity Model 1.1, Trillium 3.0, ISO/IEC 15504 (SPICE), Bootstrap 3.0 and the two representation of the Capability Maturity Model Integration 1.1 for Software engineering. These models have been selected for several reasons. Three of them, (SW-CMM, ISO/IEC 15504 and CMMI), because of their representativeness and wide use across international organizations. Trillium and Bootstrap have been integrated in this report to include least-known models and allowing comparisons with the others. In the report, models are presented according to the date of release.

The report is structured as follow. The second section presents general information about the five models. The third section contains the proposed modeling for each of the models. In the last section, we present the concluding remarks and our future works.

2 Selected models

2.1 Capability Maturity Model for Software 1.1 (SW-CMM) SEI (1993)

In November 1986, the Software Engineering Institute (SEI) began, in association with the Mitre Corporation, the development of a process maturity framework to help organizations to improve software processes[1]. One year later, in September 1987, the SEI presented the outline of the future framework detailed later in Watts Humphrey's book *Managing the Software Process* [HUMP89]. The first version of the Capability Maturity Model for software, version 1.0, was reviewed and used in 1991 and 1992. The current and last version of the SW-CMM, version 1.1, was released in 1993 [PAUL93].

The CMM presents a set of recommended practices based on knowledge acquired from software process assessments and other feedback from both industry and government. The CMM provides software organizations with guidance on how to increase control on their software process. This framework (see Figure 1) guide organizations in selecting process improvement strategies by evaluating the current process maturity and identifying the most critical issues for process improvement.

The CMM is organized in five maturity levels (Initial, Repeatable, Defined, Managed, Optimizing). A maturity level is a "*well-defined evolutionary plateau toward achieving a mature software process.(...) Each level comprises a set of process goals that, when satisfied, stabilize an important component of the software process. Achieving each level of the maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organization. (...)*" [PAUL93A] (P.7). Each of these maturity levels indicate the process capability of an organization which is defined as describing "*the range of expected results that can be achieved by following a software process. The software process capability of an organization provides one means of predicting the most likely outcomes to be expected from the next software project the organization undertakes.*" [PAUL93A] (P.3).

The maturity levels 2 to 5 contains different Key Process Area which identify "*a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability.*" [PAUL93A] (P.30). Each Key Process Area must achieve well-defined goals to reach the next maturity level.

A Key Process Area contains a set of Key Practices which "*describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area.*" [PAUL93A] (P.39).

These Key Practices are organized into five categories called Common Features which "*are attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting.*" [PAUL93A] (P.38).

- Commitment to Perform: "*describes the actions the organization must take to ensure that the process is established and will endure. Commitment to Perform typically involves establishing organizational policies and senior management sponsorship.*" [PAUL93A] (P.38)

- Ability to Perform: "*describes the preconditions that must exist in the project or organization to implement the software process competently. Ability to Perform typically involves resources, organizational structures, and training.*" [PAUL93A] (P.38)

- Activities Performed: "*describes the roles and procedures necessary to implement a key process area. Activities Performed typically involve establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary.*" [PAUL93A] (P.38)

- Measurement and Analysis: "*describes the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed.*" [PAUL93A] (P.38)

---

[1] This description of SW-CMM come from [PAUL93B] and [PAUL99]

- Verifying Implementation: "*describes the steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance.*" [PAUL93A] (P.38)

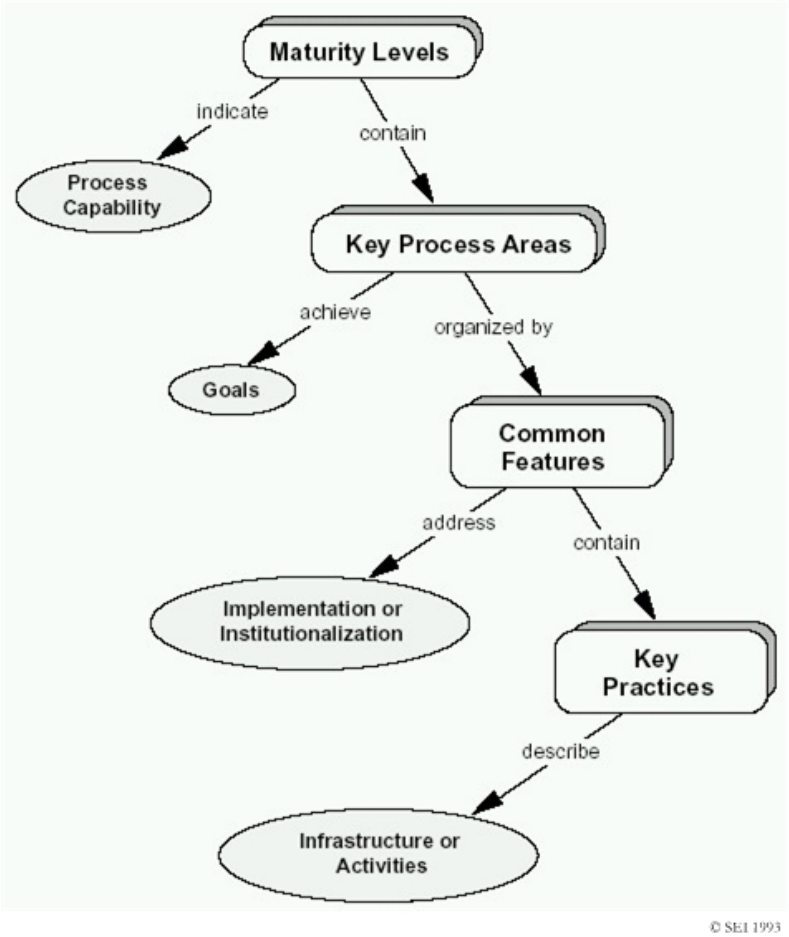

Figure 1 SW-CMM Version 1.1.

By organizing the CMM into maturity levels the software process improvement must follows the defined steps defined by the framework. The CMM is at a sufficient level of abstraction that it does not unduly constrain the way the process is implemented by an organization. So the CMM can be used in different ways: to understand the way to launch an improvement program, to develop CMM-based appraisal methods meeting specific needs, to identify the risks of selecting contractors, to identify strengths and weaknesses in organizations, to guide Software Engineering Process Group (SEPG) in organizations.

2.2 Trillium 3.0 Bell Canada (1994)

Trillium has been developed by a consortium of telecommunications companies headed by Bell Canada.[2] Trillium is based on the Software Engineering Institute *Capability Maturity Model* version 1.1. and covers all aspects of the software development life-cycle, most system and product development and support activities. Trillium has also been designed to be applied to embedded software systems such as telecommunications systems.

The goal of Trillium is to provide a mean to initiate and guide a software process improvement programme. Trillium provides a set of key industry practices which can be used to improve an existing process or lifecycle.

The architecture of Trillium differs from CMM v1.1. in several points (see Figure 2) :

- Model architecture based on roadmaps, rather than key process areas,
- Product perspective, rather than software,
- Wider coverage of capability impacting issues,
- Customer focus, technological maturity, and a telecommunications orientation.

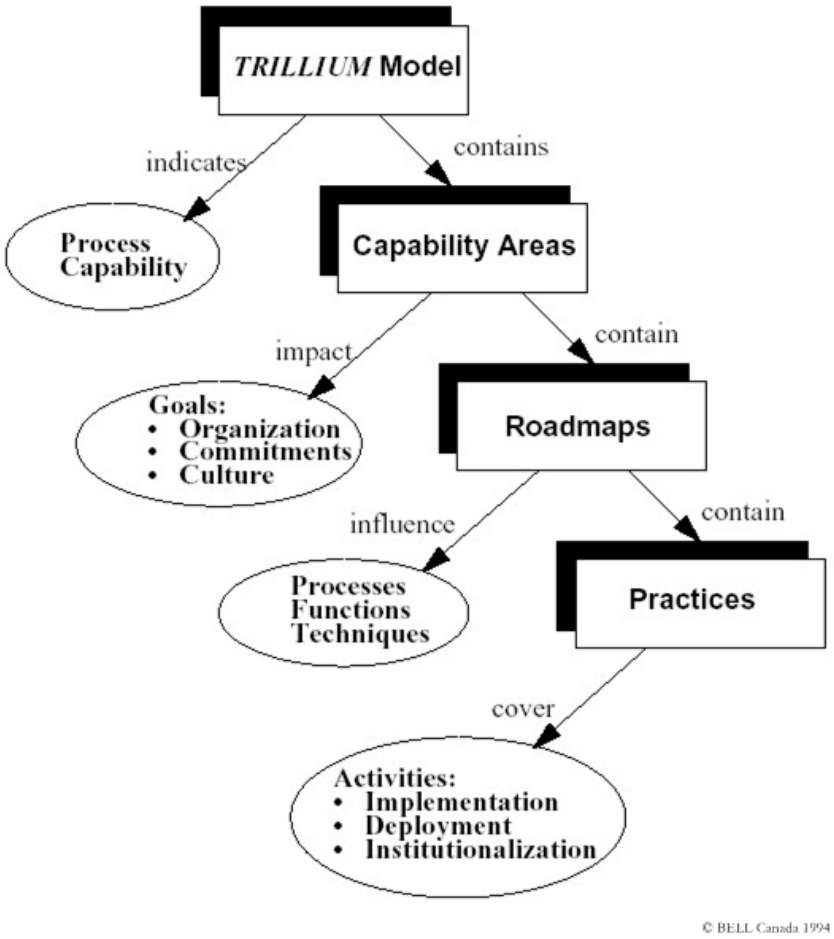

Figure 2 Trillium 3.0

[2] This description come from [ZAHR97] and [TRIL94]

The version of Trillium covers all CMM v1.1 activities and abilities and some of the commitments, measurements and verifications. The model incorporates the intent of others norms and standards:

- ISO9001: 1994 International Standard,
- ISO9000-3: 1991 Guideline,
- Bellcore TR-NWT-000179 Issue 2, June, 1993,
- Bellcore TA-NWT-001315 Issue 1, December, 1993,
- Relevant parts of the Malcom Baldrige National Quality Award, 1995 Award Criteria,
- IEEE Software Engineering Standards Collection, 1993 Edition,
- IEC Standard Publication 300:1984.

Trillium provides guidelines for improving an organization capability which is defined as "*The ability of a development organization to consistently deliver a product or an enhancement to an existing product: that meets customer expectations, with minimal defects, for the lowest life-cycle cost and in the shortest time.*" [TRIL94]

There are five Trillium levels (like CMM maturity levels) specified in the model which are: Unstructured, Repeatable and Project Oriented, Defined and Process Oriented, Managed and Integrated, Fully Integrated.

There are 8 capability areas in Trillium containing several roadmaps. A capability area "*contains practices at multiple Trillium levels*". [TRIL94] A roadmap is "*a set of related practices that focus on an organizational area or need, or a specific element within the product development process."* [TRIL94] (P.25) Each roadmap contains practices assigned at different Trillium levels as follow:

- Practices that are fundamental for the successful conclusion of a development project are assigned to level 2.
- Practices that are organization-wide in scope or fundamental to the continuous improvement of the development process are assigned to level 3.
- Practices that deal with CASE technology or characterize advanced process maturity (for example change management, integration of defect prevention, statistical process control and advanced metrics) are assigned to level 4.
- Practices that deal with advancing technology as it applies to process automation, formal methodologies and strategic utilization of organization repositories.

2.3 ISO/IEC 15504 (SPICE) (1998)

In June 1991 in London, the Joint Technical Committee 1/Sub-Committee 7 (JTC1/SC7) of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC) decided to launch a new working group (WG10) to develop an international standard on process assessment.[3] In 1993, the SPICE (Software Process Improvement and Capability dEtermination) project was established. During the following years several drafts based on existing process models like the CMM were released. In 1998 the first version of the standard was released.

This standard was intended to be used in these following modes:

- Capability determination
- Process improvement
- Self-assessment

There are 3 potential users of the standard:

- Purchasers
- Acquirers
- Software suppliers

The standard is composed of 9 different parts.

Part 1: Concepts and introductory guide (informative) is an entry point into ISO/IEC TR 15504. It describes how the parts of the suite fit together, and provides guidance for their selection and use. It explains the requirements contained within ISO/IEC TR 15504 and their applicability to the performance of an assessment.

Part 2: A reference model for processes and process capability (normative) of ISO/IEC TR 15504 defines a two dimensional reference model for describing processes and process capability used in a process assessment. The reference model defines a set of processes, defined in terms of their purpose and outcomes, and a framework for evaluating the capability of the processes through assessment of process attributes structured into capability levels. Requirements for establishing the compatibility of different assessment models with the reference model are defined.

Part 3: Performing an assessment (normative) of ISO/IEC TR 15504 defines the requirements for performing an assessment in such a way that the outcomes will be repeatable, reliable and consistent.

Part 4: Guide to performing assessments (informative) of ISO/IEC TR 15504 provides guidance on performing software process assessments, interpreting the requirements of ISO/IEC TR 15504-2 and ISO TR 15504-3 for different assessment contexts. The guidance covers the selection and use of a documented process for assessment; of a compatible assessment model(s); and of a supporting assessment instrument or tool. This guidance is generic enough to be applicable across all organizations, and also for performing assessments using a variety of different methods and techniques, and supported by a range of tools.

Part 5: An assessment model and indicator guidance (informative) of ISO/IEC TR 15504 provides an exemplar model for performing process assessments that is based upon and directly compatible with the reference model in ISO/IEC TR 15504-2. The assessment model(s) extend the reference model through the inclusion of a comprehensive set of indicators of process performance and capability.

---

[3] This description comes from [ISO98] and [ZAHR97].

Part 6: Guide to competency of assessors (informative) of ISO/IEC TR 15504 describes the competence, education, training and experience of assessors that are relevant to conducting process assessments. It describes mechanisms that may be used to demonstrate competence and to validate education, training and experience.

Part 7: Guide for use in process improvement (informative) of ISO/IEC TR 15504 describes how to define the inputs to and use the results of an assessment for the purposes of process improvement. The guide includes examples of the application of process improvement in a variety of situations.

Part 8: Guide for use in determining supplier process capability (informative) of ISO/IEC TR 15504 describes how to define the inputs to and use the results of an assessment for the purpose of process capability determination. It addresses process capability determination in both straightforward situations and in more complex situations involving, for example, future capability. The guidance on conducting process capability determination is applicable either for use within an organization to determine its own capability, or by an acquirer to determine the capability of a (potential) supplier.

Part 9: Vocabulary (normative) is a consolidated vocabulary of all terms specifically defined for the purposes of ISO/IEC TR 15504.

ISO/IEC15504-2 defines a reference model that provides a basis for rating the capability of processes, based on their achievement of defined process attributes. This reference model proposes architecture for software processes. It comprises a two-dimensional approach to the evaluation of process capability. The first, the process dimension describes the processes to be assessed and the second, the capability dimension, describes the scale for measurement of capability.

From the process dimension point of view, processes, in the reference model, are grouped into five categories:

Customer-Supplier: Processes that directly impact the customer, support development and transition of the software to the customer, and provide for the correct operation and use of the software product and/or service.

Engineering: The Engineering process category consists of processes that directly specify, implement, or maintain the software product, its relation to the system, and its customer documentation. In circumstances where the system is composed totally of software, the Engineering processes deal only with the construction and maintenance of such software.

Support: Processes that may be employed by any of the other processes (including other supporting processes) at various points in the software life cycle.

Management: Processes that contain generic practices that may be used by anyone who manages any type of project or process within a software life cycle.

Organization: Processes that establish the business goals of the organization and develop process, product, and resource assets that, when used by the projects in the organization, will help the organization achieve its business goals.

From the capability dimension, the reference model provides six capability levels. A capability level is "*a set of attribute(s) that work together to provide a major enhancement in the capability to perform a process. Each level provides a major enhancement of capability in the performance of a process. The levels constitute a rational way of progressing through improvement of the capability of any process.*". The six levels are: Incomplete, Performed, Managed, Established, Predictable, and Optimizing.

2.4 Bootstrap 3.0 (1997)

The Bootstrap method is the result of a European Community project (ESPRIT project 5441) from 1991 to 1993.[4] The goal of the project was to speed up the application of software engineering technology in the European software industry. The methodology was designed to determine the maturity of an organization, its strengths and weaknesses and the improvement guidelines. Since 1993 the project has been managed by the Bootstrap Institute. In 1997 the Bootstrap version 3.0 was released. This version has been developed to be fully compliant with ISO/IEC 15504.

The objectives of the Bootstrap methodology are:

- provide support to the evaluation of process capability against a set of recognised software engineering best practices;
- include internationally recognised software engineering standards as sources for identification of best practices;
- support the evaluation of how the reference standards have been implemented in the assessed organization;
- assure the evaluation is reliable and repeatable;
- identify strengths and weaknesses in the organization's processes;
- provide benchmarking data for assessment results comparison;
- provide results that form a suitable and reliable basis for improvement planning;
- plan improvement actions that support achievement of the organization's goals;
- help increasing process effectiveness while implementing standard requirements in the organization.

The Bootstrap is presented as an assessment model including three dimensions: a process dimension, a capability dimension and a technology support dimension. The process dimension is called the bootstrap process model. It integrates requirements from several standards like ISO9001 (1989), ISO9000-3(1991), ISO12207 (1995), ISO15504 (1993), ESA PSS-05-0 (1991) and the CMM.

The Bootstrap process architecture reflects a tree structure that contains: process categories, processes and best practices. The Bootstrap 3.0 architecture contains three main process categories (see Figure 3):

- Organization
- Methodology
- Technology

[4] This description comes from [KUVA99], [HAAS94], [STIE97], [ZAHR97].

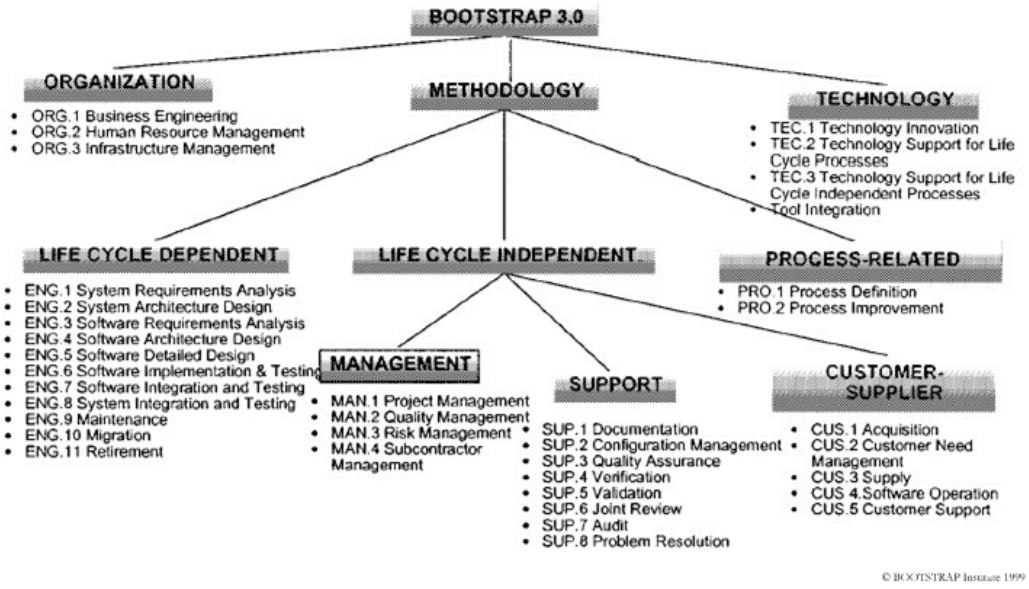

Figure 3 Bootstrap 3.0

2.5 Capability Maturity Model Integration 1.1 (CMMI) for Software Engineering SEI (2002)

2.5.1 CMMI

Since the first version in 1991, several CMM's have been developed for different disciplines[5]. The most notables are the models for systems engineering, software engineering, software acquisition, workforce management and development, and integrated product and process development. These models were very useful for the organizations but multiple uses inside an organization became problematic. Many organizations would like to focus their improvement efforts across the disciplines in their organizations. However, the differences among these disciplines-specific models, including their architecture, content, and approach, have limited these organizations ability to focus their improvements successfully. The use of multiple models in an organization is also costly in terms of appraisal, trainings and improvement activities.

There are three main sources models used in CMMI development:

- Software: SW-CMM, draft version 2(C)
- System engineering: EIA/IS 731
- Integrated product and process development: IPD CMM, version 0.98

The combination of these models into a single improvement framework was intended for use by organizations in their pursuit of enterprise-wide process improvement. These models were selected because of their widespread adoption in the software and systems engineering communities and because of their different approaches to improve processes in an organization.

CMMI is the successor of these models (SW-CMM version 2 development has been stopped). This framework has been designed to support the future integration of other disciplines. Furthermore, CMMI was developed to be compatible with the ISO/IEC 15504.

There are multiple CMMI models available, as generated from the CMMI framework. For these models there are two representations available: *Continuous* and *Staged*. Both representation presents their own characteristics and the organization must chose the representation fitting with its objectives.

Continuous representation:

- Allow you to select the order of improvement that best meets the organization's business objectives and mitigates the organization's area of risk
- Enable comparisons across and among organizations on a process area by process area basis or by comparing results through the use of equivalent staging
- Provide an easy migration from EIA/IS 731 to CMMI
- Afford an easy comparison of process improvement to ISO/IEC 15504

Staged representation:

- Provide a proven sequence of improvements, beginning with basic management practices and progressing through a predefined and proven path of successive levels, each serving as a foundation for the next

---

[5] This description comes from [AHER01], [CHRI03], [CMMI02A], [CMMI02B].

- Permit comparisons across and among organizations by the use of maturity levels
- Provide an easy migration from SW-CMM to CMMI
- Provide single rating that summarize appraisals results and allows comparisons among organizations

Currently there are four bodies of knowledge (also referenced as "disciplines") available to you when selecting a CMMI model:

- System engineering
- Software engineering
- Integrated Product and Process Development
- Supplier sourcing

In this report we will take into account the two representations of CMMI dedicated to the Software Engineer discipline.

2.5.2 Continuous Representation

In a continuous representation the practices of a process area are organized in a way that supports individual process area growth and improvement[6]. Most of the practices associated with process improvement are generic; they are external to the individual process areas and apply to all process areas. The generic practices are grouped into capability levels, each of which has a definition that is equivalent to the definition of the maturity levels in a staged model. Process areas are improved and institutionalized by implementing the generic practices in those process areas. In a continuous model such as EIA/IS 731, goals are not specifically stated, which puts even more emphasis on practices. The collective capability levels of all process areas determine organizational improvement, and an organization can tailor a continuous model and target only certain process areas for improvement.

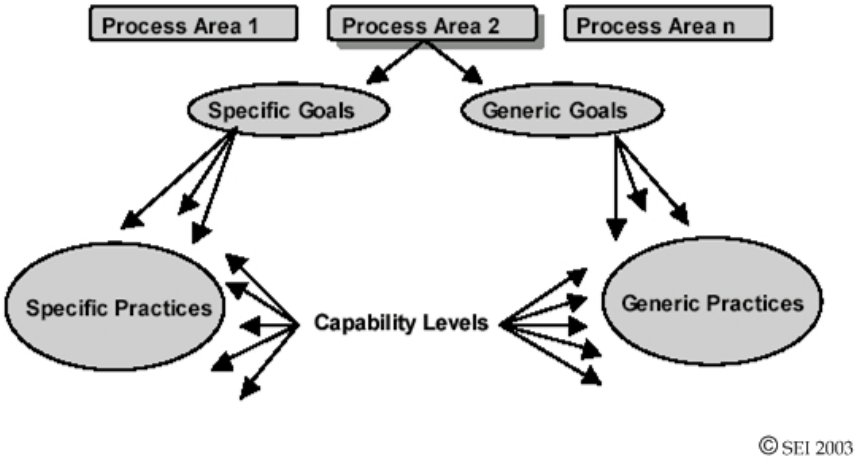

Figure 4 CMMI for Software Engineering. Continuous Representation

As shown in Figure 4, the specific goals organize specific practices and the generic goals organize generic practices. Each specific and generic practice corresponds to a capability level. Specific goals and specific practices apply to individual process areas.

Generic goals and generic practices apply to multiple process areas. The generic goals and generic practices define a sequence of capability levels that represent improvements in the implementation and effectiveness of all the processes you choose to improve.

---

[6] This description comes from [AHER01].

2.5.3 Staged Representation

The staged representation provides a predefined road map for organizational improvement based on proven grouping and ordering of processes and associated organizational relationships. The term "staged" comes from the different stages which are the five maturity levels. Each maturity level has a set of process areas that indicate where an organization should focus to improve its organizational process.

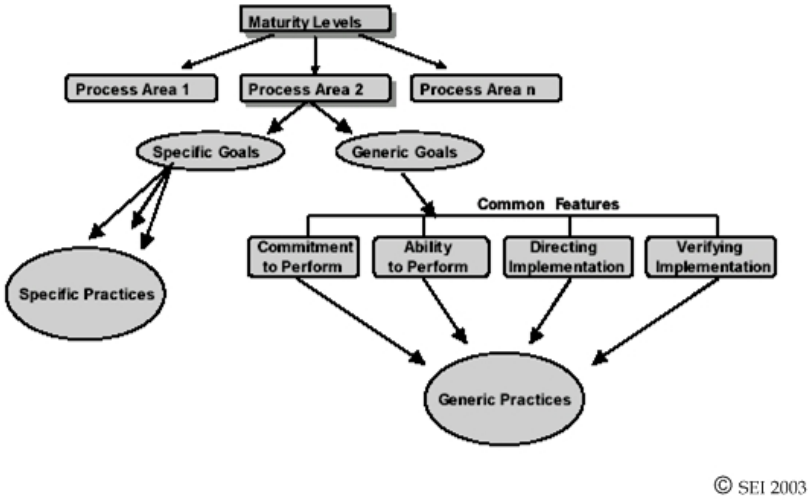

Figure 5 CMMI for Software Engineering. Staged Representation

CMMI models are designed to describe discrete levels of process improvement. In the staged representation, maturity levels provide a recommended order for approaching process improvement in stages. As illustrated in Figure 5, maturity levels organize the process areas. Within the process areas are generic and specific goals as well as generic and specific practices. Common features organize generic practices.

This representation focuses on best practices your organization can use to improve processes in the process areas that are within the maturity level it chooses to achieve. Before you begin using a CMMI model for improving processes, you must map your processes to CMMI process areas. This mapping enables you to control process improvement in your organization by helping you track your organization's level of conformance to the CMMI model you are using. It is not intended that every CMMI process area maps one to one with your organization's processes.

3 UML Class diagram modeling

3.1 Class Diagram

According to the version 1.5 of the OMG-Unified Modeling Language specifications[7] a *Class Diagram* is a graph of Classifier elements connected by their various static relationships. A class diagram is a collection of static declarative model elements, such as classes, interfaces, and their relationships, connected as a graph to each other and to their contents.

3.2 Components of a class diagram

3.2.1 Class

A class is the descriptor for a set of objects with similar structure, behaviour, and relationships. A class represents a concept within the system being modelled. Classes have data structure and behaviour and relationships to other elements.

3.2.2 Binary Association

A binary association is an association among exactly two classifiers[8] (including the possibility of an association from a classifier to itself).

The association name designates the (optional) name of the association. It is shown as a name string near the path (but not near enough to an end to be confused with a rolename). The name string may have an optional small black solid triangle in it. The point of the triangle indicates the direction in which to read the name. The name-direction arrow has no semantics significance, it is purely descriptive. The classifiers in the association are ordered as indicated by the name-direction arrow.

3.2.3 Multiplicity

A multiplicity item specifies the range of allowable cardinalities that a set may assume. Multiplicity specifications may be given for roles within associations, parts within composites, repetitions, and other purposes. Essentially a multiplicity specification is a subset of the open set of nonnegative integers.

3.2.4 Generalization

Generalization is the taxonomic relationship between a more general element (the parent) and a more specific element (the child) that is fully consistent with the first element and that adds additional information.

Generalization is shown as a solid-line path from the child (the more specific element, such as a subclass) to the parent (the more general element, such as a superclass), with a large hollow triangle at the end of the path where it meets the more general element.

3.2.5 Aggregation

A hollow diamond is attached to the end of the path to indicate aggregation. The diamond may not be attached to both ends of a line, but it need not be present at all. The diamond is attached to the class that is the aggregate. The aggregation is optional, but not suppressible.

---

[7] This description comes from [UML03].

[8] Classifier is the metamodel superclass of Class, DataType, and Interface. All of these have similar syntax and are therefore all notated using the rectangle symbol with keywords used as necessary. [UML03]

4 Proposed modeling for the five models

4.1 General notes on the proposed modelling

In the following section the class diagram of the different process improvement models are proposed. For each model there are three sections. *Class Diagram*, *Definitions of classes* and *Modelling notes*.

Class Diagram contains only the graphic modelling.

Definitions of classes contains, for each class present in the diagram, the definition proposed in the original model with its detailed bibliographical reference. In some case (e.g. Bootstrap, we did not found explicit definitions. So we collected all available elements to build a definition)

Modelling notes contains complementary comments about the proposed modelling. In some case modeling choices have been done and are justified in this section.

Associations between classes are entirely based on the definitions and are understandable just by reading the two definitions. But in some cases, this association is deducted and several modeling choices are possible.

By default, when the multiplicity equals to 1, it is hidden as recommended by [UML03].

4.2 SW-CMM 1.1

4.2.1 Class Diagram of SW-CMM 1.1

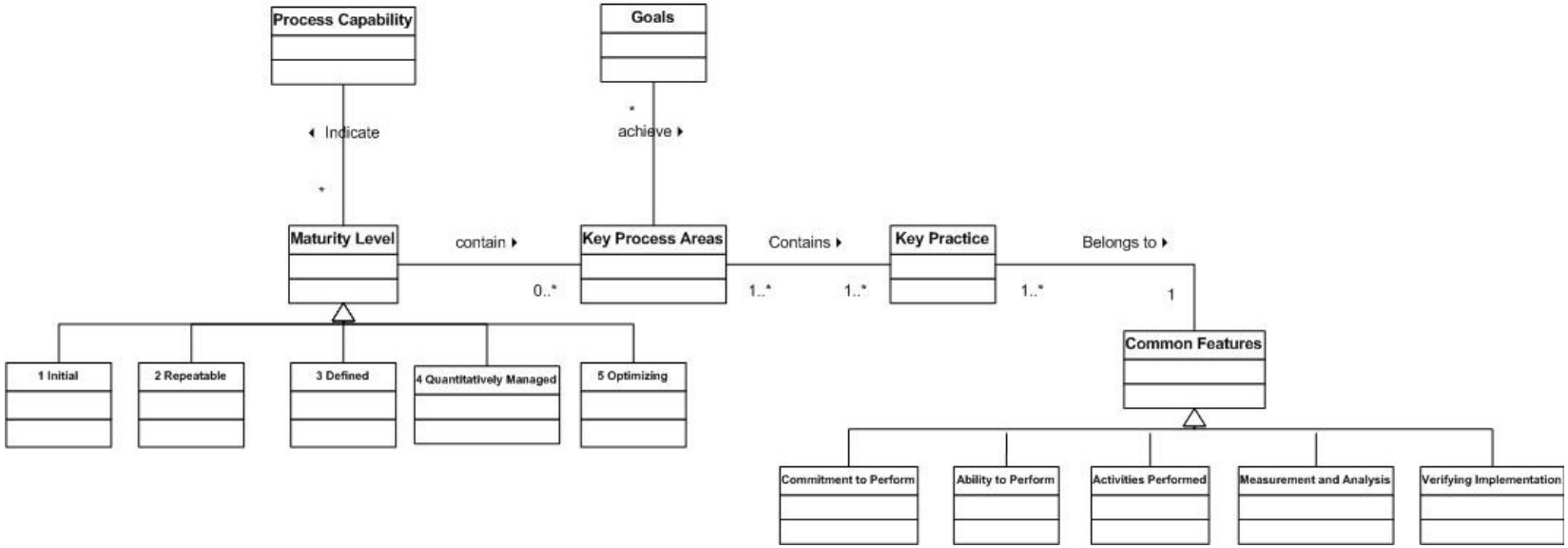

Figure 6 Class Diagram of SW- CMM 1.1

4.2.2 Definitions of classes

PROCESS CAPABILITY: "describes the range of expected results that can be achieved by following a software process. The software process capability of an organization provides one means of predicting the most likely outcomes to be expected from the next software project the organization undertakes." [PAUL93A] (p. 3)

MATURITY LEVEL: "is a well-defined evolutionary plateau toward achieving a mature software process. (...) Each level comprises a set of process goals that, when satisfied, stabilize an important component of the software process. Achieving each level of the maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organization. (...) Organizing the CMM into the five levels (...)."[PAUL93A] (p.7)

1 INITIAL: "The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort." [PAUL93A] (p. 8) There is no Key Process Area at this basic level." [PAUL93A] (p. 9)

2 REPEATABLE: "Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications." [PAUL93A] (p. 9)

3 DEFINED: "The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software." [PAUL93A] (p. 9)

4 MANAGED: "Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled." [PAUL93A] (p. 9)

5 OPTIMIZING: "Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies." [PAUL93A] (p. 9) These levels contains at least 2 KPA.

KEY PROCESS AREA: "identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability. The key process areas have been defined to reside at a single maturity level (...) Nevertheless, all the goals of a key process area must be achieved for the organization to satisfy that key process area. When the goals of a key process area are accomplished on a continuing basis across projects, the organization can be said to have institutionalized the process capability characterized by the key process area." [PAUL93A] (p. 30)

COMMON FEATURES: "are attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting." [PAUL93A] (p. 37) There are five common features.

COMMITMENT TO PERFORM: "describes the actions the organization must take to ensure that the process is established and will endure. Commitment to Perform typically involves establishing organizational policies and senior management sponsorship." [PAUL93A] (p. 38)

ABILITY TO PERFORM: "describes the preconditions that must exist in the project or organization to implement the software process competently. Ability to Perform typically involves resources, organizational structures, and training." [PAUL93A] (p. 38)

ACTIVITIES PERFORMED: "describes the roles and procedures necessary to implement a key process area. Activities Performed typically involve establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary." [PAUL93A] (p.38)

MEASUREMENT AND ANALYSIS: "describes the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed." [PAUL93A] (p.38)

VERIFYING IMPLEMENTATION: "describes the steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance." [PAUL93A] (p. 38)

GOALS: "summarize the key practices of a key process area and can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area." [PAUL93A] (p. 32)

KEY PRACTICES: "describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area. Each key practice consists of a single sentence, often followed by a more detailed description, which may include examples and elaboration. These key practices, also referred to as the top-level key practices, state the fundamental policies, procedures, and activities for the key process area." [PAUL93A] (p. 39)

4.2.3 Modelling notes

A *Maturity Level* is, in SW-CMM, the parent class of five level (1 to 5) so it is presented as a generalization. By modeling all the child, whose the first *1 Initial*, the multiplicity of the association between *Maturity Level* and *Key Process Area* is 0..* because the *Initial* level does not contains any *Key Process Area*.

The multiplicity of the association between *Key Process Area* and *Common Features* is 5 because any *Key Process Area* has exactly 5 *Common Features*.

4.3 Trillium 3.0

4.3.1 Class Diagram of Trillium 3.0

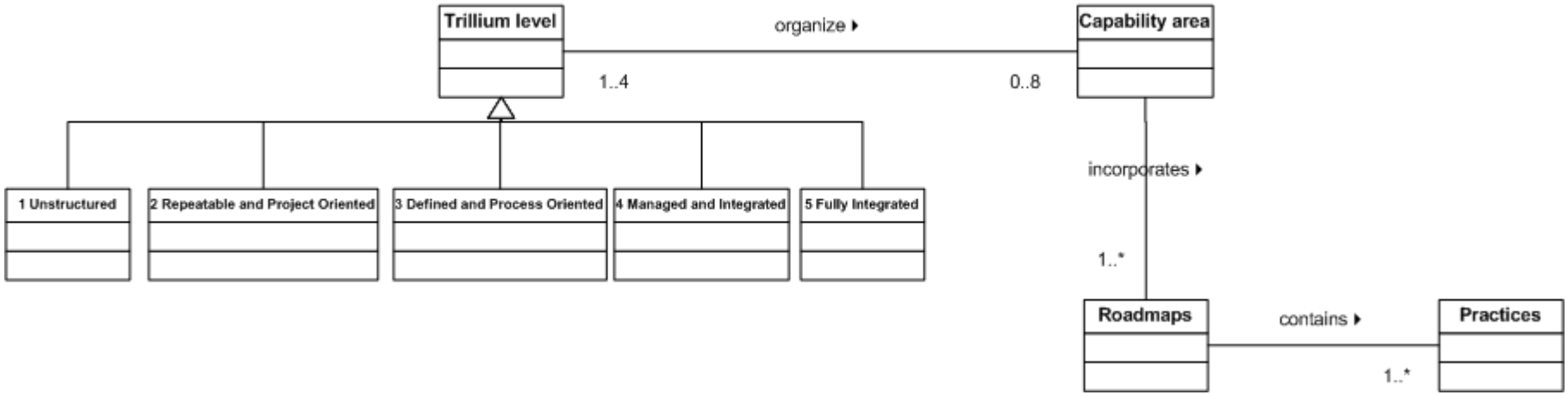

Figure 7 Class Diagram of Trillium 3.0

4.3.2 Definitions of classes

TRILLIUM LEVEL: "The Trillium scale spans levels 1 through 5. The levels can be characterized in the following way (…)". [TRIL94] (p.22) "To achieve a Trillium level, an organization must satisfy a minimum of 90% of the criteria in each of the 8 Capability Areas at the level. Levels 3, 4 and 5 require the achievement of all lower Trillium levels (i.e., levels cannot be skipped)." [TRIL94] (p. 24)

1 UNSTRUCTURED: "The development process is adhoc. Projects frequently cannot meet quality or schedule targets. Success, while possible, is based on individuals rather than on organizational infrastructure. (Risk - High)" [TRIL94] (p.22)

2 REPEATABLE AND PROJECT ORIENTED: "Individual project success is achieved through strong project management planning and control, with emphasis on requirements management, estimation techniques, and configuration management. (Risk - Medium)" [TRIL94] (p.22)

3 DEFINED AND PROCESS ORIENTED: "Processes are defined and utilized at the organizational level, although project customization is still permitted. Processes are controlled and improved. ISO 9001 requirements such as training and internal process auditing are incorporated. (Risk - Low)" [TRIL94] (p.22)

4 MANAGED AND INTEGRATED: "Process instrumentation and analysis is used as a key mechanism for process improvement. Process change management and defect prevention programs are integrated into processes. CASE tools are integrated into processes. (Risk - Lower)" [TRIL94] (p.23)

5 FULLY INTEGRATED: "Formal methodologies are extensively used. Organizational repositories for development history and process are utilized and effective. (Risk - Lowest)" [TRIL94] (p.23)

CAPABILITY AREA: "There are 8 Capability Areas within the Trillium model. Each Capability Area contains practices at multiple Trillium levels. For example, Management spans levels 2 to 4 while Quality System spans levels 2 to 5." [TRIL94] (p.24)

ROADMAPS: "Each Capability Area incorporates one or more roadmaps. A roadmap is a set of related practices that focus on an organizational area or need, or a specific element within the product development process. Each roadmap represents a significant capability for a software development organization. Within a given roadmap, the level of the practices is based on their respective degree of maturity. The most fundamental practices are at a lower level whereas the most advanced ones are located at the higher level. An organization matures through the roadmap levels. Lower level practices must be implemented and sustained for higher level practices to achieve maximum effectiveness." [TRIL94] (p.25)

PRACTICES: There is no proposed definition for *Practices*. But the different sources used by Trillium are listed in [TRIL94] (p.28-29).

4.3.3 Modelling notes

As SW-CMM, Trillium contains five *Trillium (*maturity) Levels which are identically organized. So our modeling for Trillium is exactly the same, by using a generalization. By modeling the first level *Unstructured*, the multiplicity of the association between *Trillium Level* and *Capability Area* is 0..8.

4.4 ISO/IEC 15504

4.4.1 Class Diagram of ISO/IEC 15504-Part 2 Reference Model

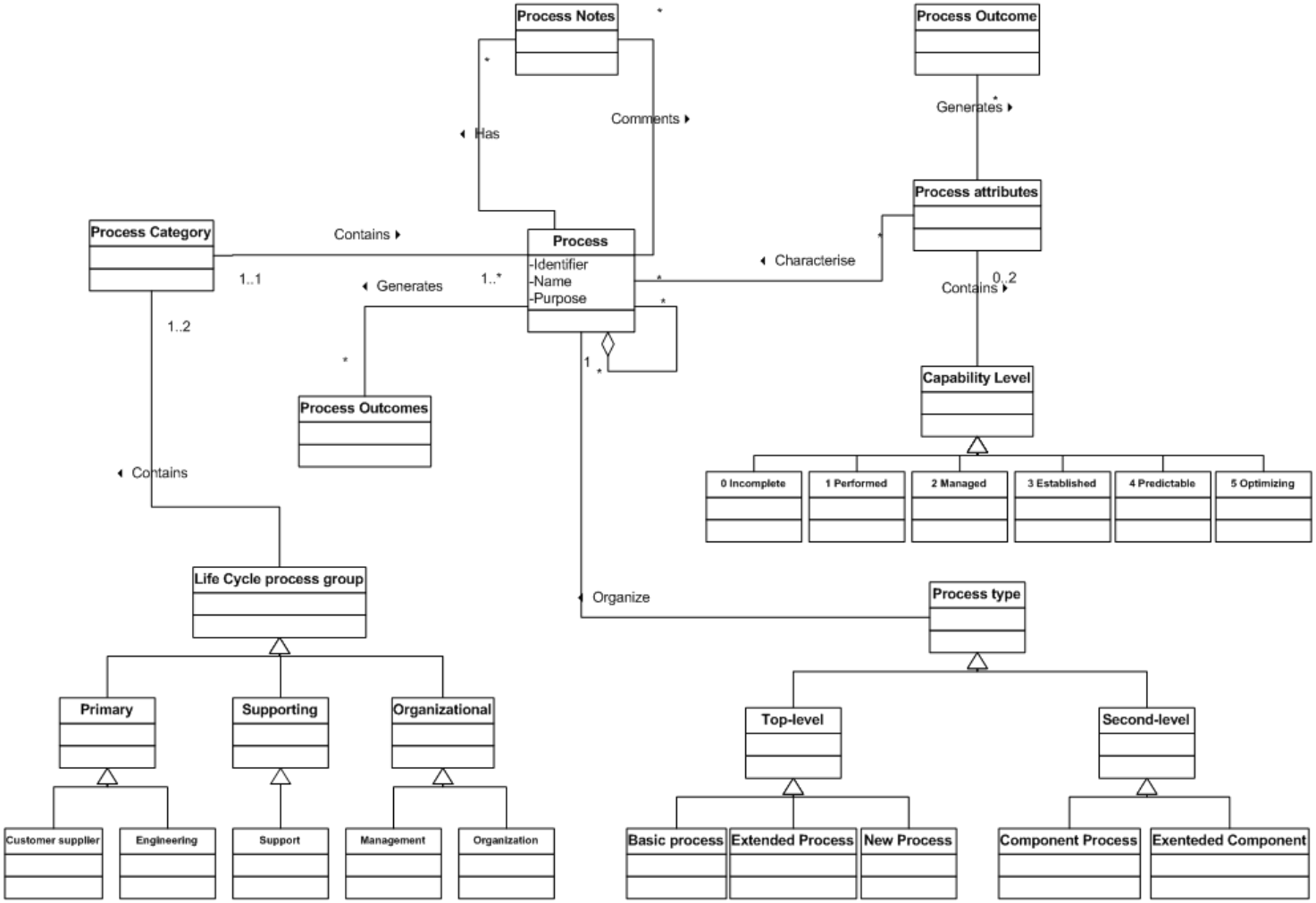

Figure 8 Class Diagram of ISO/IEC15504- Part 2 Reference Model

4.4.2 Definitions of classes (Part 2 – Reference Model)

CAPABILITY LEVEL: "is characterized by a set of attribute(s) that work together to provide a major enhancement in the capability to perform a process. Each level provides a major enhancement of capability in the performance of a process. The levels constitute a rational way of progressing through improvement of the capability of any process." [ISO98B] (p.4)

0 INCOMPLETE: "There is general failure to attain the purpose of the process. There are little or no easily identifiable work products or outputs of the process." [ISO98B] (p.4)

1 PERFORMED: "The purpose of the process is generally achieved. The achievement may not be rigorously planned and tracked. Individuals within the organization recognize that an action should be performed, and there is general agreement that this action is performed as and when required. There are identifiable work products for the process, and these testify to the achievement of the purpose." [ISO98B] (p.4)

2 MANAGED: "The process delivers work products according to specified procedures and is planned and tracked. Work products conform to specified standards and requirements. The primary distinction from the Performed Level is that the performance of the process now delivers work products that fulfil expressed quality requirements within defined timescales and resource needs." [ISO98B] (p.4)

3 ESTABLISHED: "The process is performed and managed using a defined process based upon good software engineering principles. Individual implementations of the process use approved, tailored versions of standard, documented processes to achieve the process outcomes. The resources necessary to establish the process definition are also in place. The primary distinction from the Managed Level is that the process of the Established Level is using a defined process that is capable of achieving its process outcomes." [ISO98B] (p.4)

4 PREDICTABLE: "The defined process is performed consistently in practice within defined control limits, to achieve its defined process goals. Detailed measures of performance are collected and analyzed. This leads to a quantitative understanding of process capability and an improved ability to predict and manage performance. Performance is quantitatively managed. The quality of work products is quantitatively known. The primary distinction from the Established Level is that the defined process is now performed consistently within defined limits to achieve its process outcomes." [ISO98B] (p.4)

5 OPTIMIZING: "Performance of the process is optimized to meet current and future business needs, and the process achieves repeatability in meeting its defined business goals. Quantitative process effectiveness and efficiency goals (targets) for performance are established, based on the business goals of the organization. Continuous process monitoring against these goals is enabled by obtaining quantitative feedback and improvement is achieved by analysis of the results. Optimizing a process involves piloting innovative ideas and technologies and changing non-effective processes to meet defined goals or objectives. The primary distinction from the Predictable Level is that the defined and standard processes now dynamically change and adapt to effectively meet current and future business goals." [ISO98B] (p.4)

LIFE CYCLE PROCESS GROUP: there are three 3 life cycle level process groups containing five process categories. The groups are: Primary life cycle processes, Supporting life cycle processes and Organizational life cycle processes. [ISO98B] (p.6)

PRIMARY LIFE CYCLE PROCESSES: "consist of the process categories Customer-Supplier and Engineering and are described as follows:" [ISO98B] (p.6)

CUSTOMER-SUPPLIER PROCESS CATEGORY: "consists of processes that directly impact the customer, support development and transition of the software to the customer, and provide for the correct operation and use of the software product and/or service." [ISO98B] (p.7)

ENGINEERING PROCESS CATEGORY: "consists of processes that directly specify, implement, or maintain the software product, its relation to the system and its customer documentation." [ISO98B] (p.10)

SUPPORTING LIFE CYCLE PROCESSES: "consist of the process category Support and this category is described as follows:" [ISO98B] (p.6)

SUPPORT PROCESS CATEGORY: "consists of processes which may be employed by any of the other processes (including other supporting processes) at various points in the software life cycle." [ISO98B] (p.14)

ORGANIZATION LIFE CYCLE PROCESSES: "consist of the process categories Management and Organization and are described as follows:" [ISO98B] (p.6)

MANAGEMENT PROCESS CATEGORY: "consists of processes which contain practices of a generic nature which may be used by anyone who manages any type of project or process within a software life cycle." [ISO98B] (p.17)

ORGANIZATION PROCESS CATEGORY: "consists of processes that establish the business goals of the organization and develop process, product, and resource assets which, when used by the projects in the organization, will help the organization achieve its business goals." [ISO98B] (p.19)

PROCESS CATEGORY: "A set of processes addressing the same general area of activity. The process categories address five general areas of activity: customer-supplier, engineering, support, management, and organization." [ISO98D] (p.4)

PROCESS: "A set of interrelated activities, which transform inputs into outputs." [ISO98D] (p.4)

PROCESS OUTCOME: "an observable result of the successful implementation of a process." [ISO98D] (p.5)

PROCESS PURPOSE: "The high level measurable objectives of performing the process and the likely outcomes of effective implementation of the process." [ISO98D] (p.5)

PROCESS ATTRIBUTE: "a measurable characteristic of process capability applicable to any process." [ISO98D] (p.4)

PROCESS CAPABILITY LEVEL: "a point on the six-point ordinal scale (of process capability) that represents the increasing capability of the performed process; each level builds on the capability of the level below." [ISO98D] (p.4)

PROCESS TYPE: "there are five types of process. 3 top-level (basic, extended and new) and 2 second-level (component and extended component)." [ISO98B] (p.6)

TOP-LEVEL: "(basic, extended and new)" [ISO98B] (p.6)

SECOND-LEVEL: "(component and extended component)." [ISO98B] (p.6)

BASIC PROCESSES: "identical in intent to the processes in ISO/IEC 12207;" [ISO98B] (p.6)

EXTENDED PROCESSES: "that are expansions of ISO/IEC 12207 processes;" [ISO98B] (p.6)

NEW PROCESSES: "that are outside the scope of ISO/IEC 12207;" [ISO98B] (p.6)

COMPONENT PROCESSES: "(a group of one or more ISO/IEC 12207's activities from the same process);" [ISO98B] (p.6)

EXTENDED COMPONENT: "Processes that are one or more of ISO/IEC 12207's activities from the same process, with additional material. These would normally be Component Processes of Extended Processes." [ISO98B] (p.6)

PROCESS NOTES: "an optional list of informative notes regarding the process and its relation to other processes." [ISO98B] (p.6)

4.4.3 Modelling notes (Part 2 – Reference Model)

A *Process* can be the parent process of other processes. This relation is modelled by the aggregation.

A *Capability Level* is the parent class of six levels (0 to 5) so it is presented as a generalization. By modeling all the child, whose the first *0 Incomplete*, the multiplicity of the association between *Capability Level* and *Process Attributes* is 0..2.

Process has three attributes which are has several (*) *Process Notes* and a *Process Notes* comments only one *Process*.

4.4.4 Class Diagram of ISO/IEC 15504-Part 5 Assessment Model

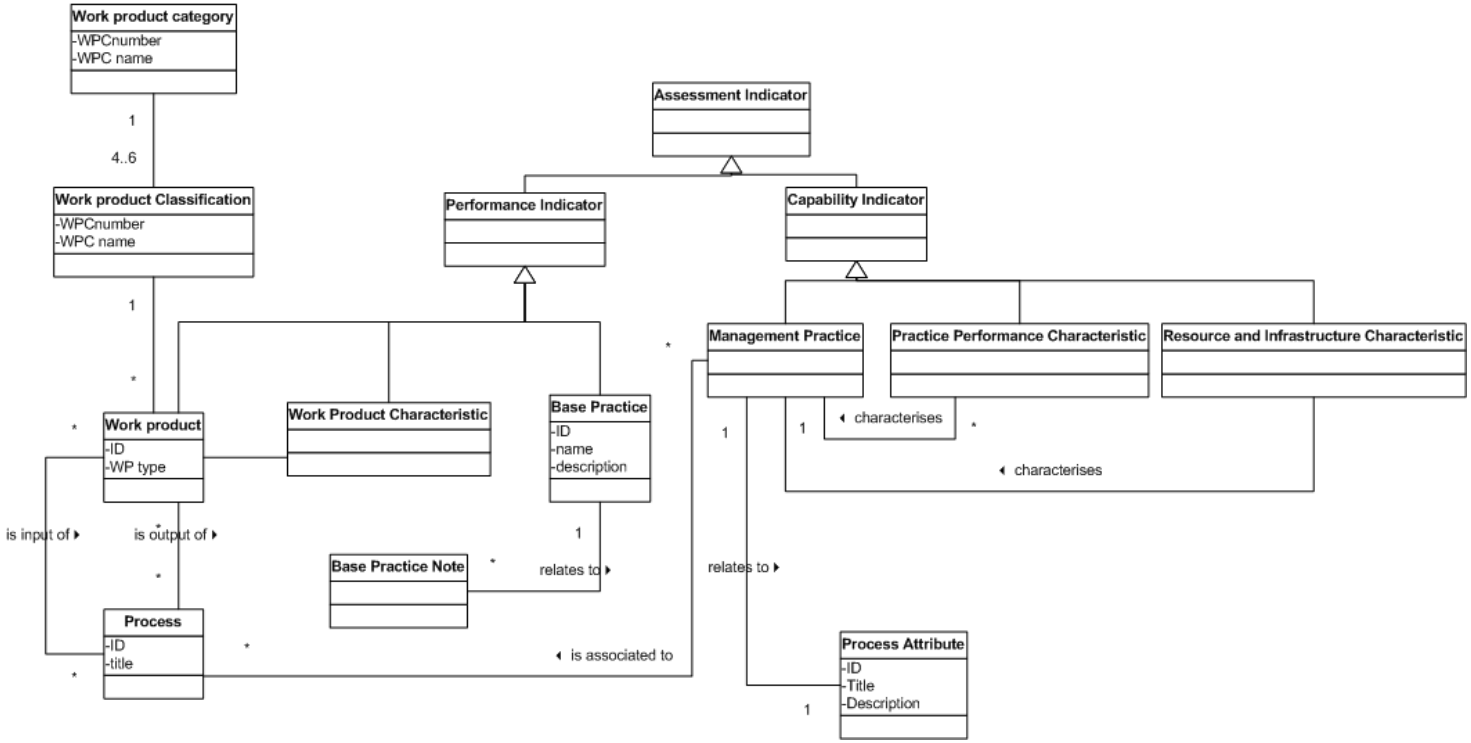

Figure 9 Class Diagram of ISO/IEC15504- Part 5 Assessment Model[9]

[9] This modeling is mainly inspired by [LEPA02]

4.4.5 Definitions of classes (Part 5 – Assessment Model)

WORK PRODUCT CATEGORY: "3 categories : Organisation, Project and Records" [ISO98C] (p.92)

WORK PRODUCT CLASSIFICATION: "Provides a classification of the work products into 3 categories : Organisation, Project and Records" [ISO98C] (p.92)

WORK PRODUCT CHARACTERISTIC: "Provides examples of the potential characteristics associated with the work product types." [ISO98C] (p.92)

WORK PRODUCT: "an artefact associated with the execution of a process. NOTE A work product might be used, produced or changed by a process." [ISO98D] (p.6)

PROCESS: "A set of interrelated activities, which transform inputs into outputs." [ISO98D] (p.4)

PROCESS ATTRIBUTE: "a measurable characteristic of process capability applicable to any process." [ISO98D] (p.4)

ASSESSMENT INDICATOR: "an objective attribute or characteristic of a practice or work product that supports the judgment of the performance of, or capability of, an implemented process" [ISO98D] (p.1)

PERFORMANCE INDICATOR: Base Practices (Clause 5) Work Products and their Characteristics (Annex A & Annex C) [ISO98C] (p.2)

CAPABILITY INDICATOR: Management practices (Clause 6) Practice performance characteristics (Annex B) Resource and Infrastructure Characteristics (Annex B) [ISO98C] (p.2)

MANAGEMENT PRACTICE: "a management activity or task that addresses the implementation or institutionalisation of a specific process attribute" [ISO98C] (p.2)

PRACTICE PERFORMANCE CHARACTERISTIC: "represent the type of evidence that would substantiate judgements of the extent to which the management practice is performed." [ISO98C] (p.6)

RESOURCE AND INFRASTRUCTURE CHARACTERISTIC: "represent the type of evidence that would substantiate judgements of the extent to which the management practice is performed." [ISO98C] (p.6)

BASE PRACTICE: "a software engineering or management activity that, when consistently performed, contributes to achieve the purpose of a particular process" [ISO98C] (p.2)

4.5 Bootstrap 3.0

4.5.1 Class Diagram of Bootstrap 3.0

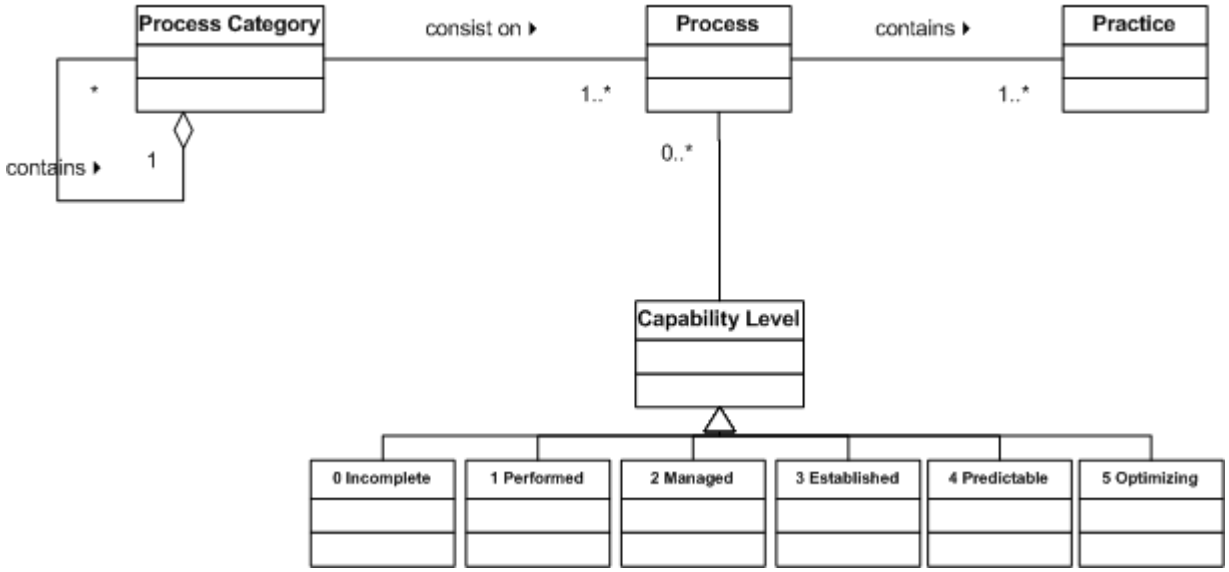

Figure 10 Class Diagram of Bootstrap 3.0

4.5.2 Definitions of classes

PROCESS CATEGORY: contains process or a other process category. [KUVA99] (p.12)

PROCESS: no definition available

PRACTICE: no definition available

CAPABILITY LEVEL: "is the ability of each process to achieve its goals in the context of the assessed organisation." [KUVA99] (p.13)

0 INCOMPLETE: "Process the process fails to achieve its purpose as it is incompletely implemented;" [KUVA99] (p.13)

1 PERFORMED: "Process a set of practices is performed that allow the process to achieve its purpose;" [KUVA99] (p.13)

2 MANAGED: "Process the process delivers work products of acceptable quality within defined time-scale and resources;" [KUVA99] (p.13)

3 ESTABLISHED: "Process the process is performed based on a recognised organisational process definition;" [KUVA99] (p.13)

4 PREDICTABLE: "Process the established process is performed using defined quantitative control limits;" [KUVA99] (p.13)

5 OPTIMISING: "Process changes to the definition, management and performance of the process are identified and performed in a controlled way to continuously improve process performance." [KUVA99] (p.13)

4.5.3 Modelling notes

A *Capability Level* is the parent class of six levels (0 to 5) so it is presented as a generalization. By modeling all the child, whose the first *0 Incomplete*, the multiplicity of the association between *Capability Level* and *Process* is 0..* because the *Incomplete* level does not rate any *Process*.

A *Process category* can contain an other *Process category*.

4.6 CMMI for software engineering 1.1 - Continuous Representation

4.6.1 Class Diagram of CMMI 1.1 - Continuous Representation

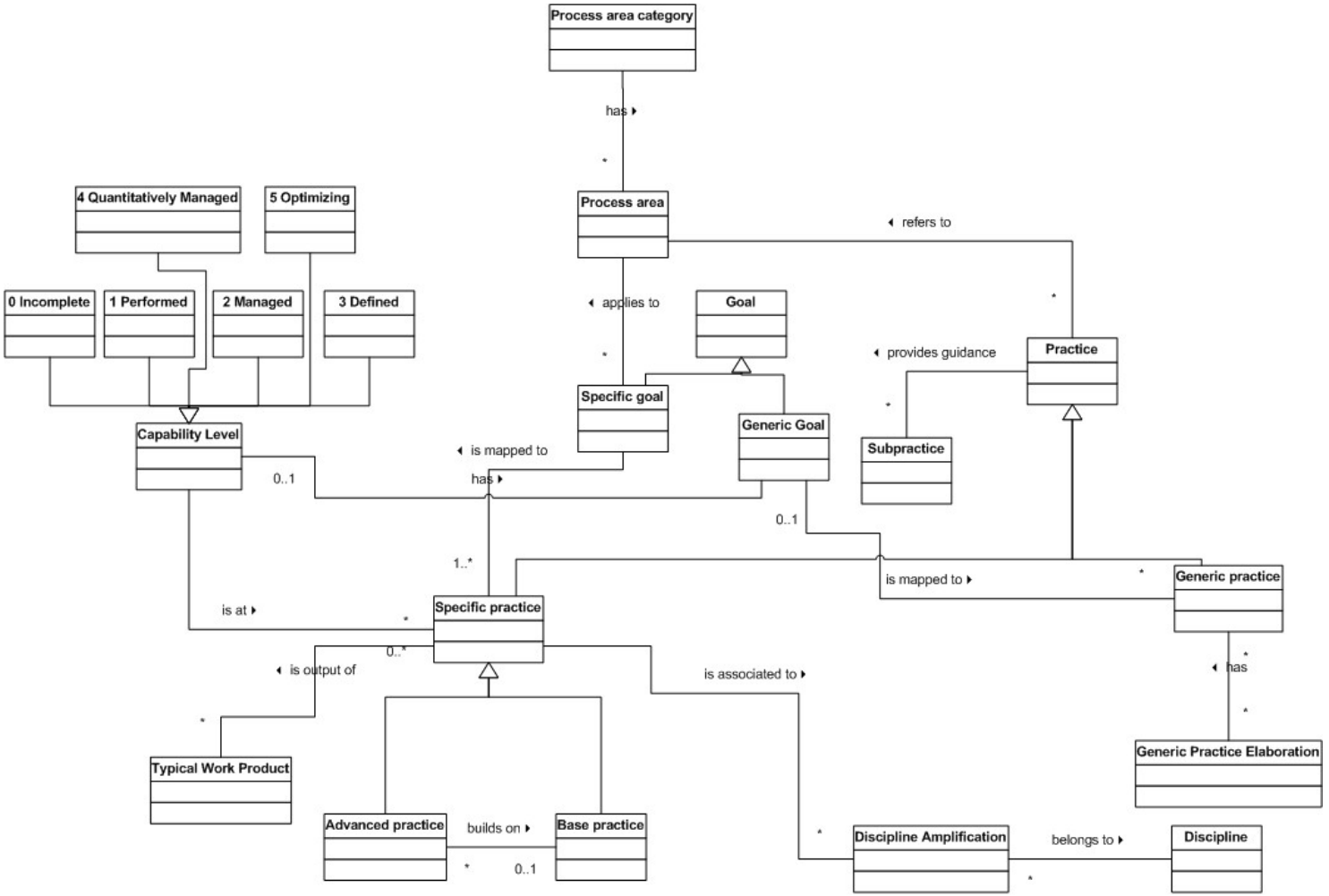

Figure 11 Class Diagram of CMMI 1.1 - Continuous Representation

4.6.2 Definitions of classes

CAPABILITY LEVEL: "consists of related specific and generic practices for a process area that can improve the organization's processes associated with that process area. As you satisfy the generic and specific goals for a process area at a particular capability level, and you achieve that capability level, you reap the benefits of process improvement. [FM103.HDA101.HDB102.T101]

There are six capability levels, designated by the numbers 0 through 5: [FM103.HDA101.HDB102.T103] 0. Incomplete; 1. Performed; 2. Managed; 3. Defined; 4. Quantitatively Managed; 5. Optimizing" [CMMI02A] (p. 13)

0 INCOMPLETE: "An incomplete process is a process that is either not performed or partially performed. One or more of the specific goals of the process area are not satisfied. [CL101]" [CMMI02A] (p. 35)

1 PERFORMED: "A capability level 1 process is characterized as a "performed process." [CL102] A performed process is a process that satisfies the specific goals of the process area. It supports and enables the work needed to produce identified output work products using identified input work products. [CL102.N104] A critical distinction between an incomplete process and a performed process is that a performed process satisfies all of the specific goals of the process area. [CL102.N103]" [CMMI02A] (p. 35)

2 MANAGED: "A capability level 2 process is characterized as a "managed process."[CL103] A managed process is a performed (capability level 1) process that is also planned and executed in accordance with policy, employs skilled people having adequate resources to produce controlled outputs, involves relevant stakeholders; is monitored, controlled, and reviewed; and is evaluated for adherence to its process description. The process may be instantiated by an individual project, group, or organizational function." [CMMI02A] (p. 36-37)

3 DEFINED: "A capability level 3 process is characterized as a "defined process." [CL104] A defined process is a managed (capability level 2) process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes work products, measures, and other process-improvement information to the organizational process assets. [CL104.N106]" [CMMI02A] (p. 46)

4 QUANTITATIVELY MANAGED: A capability level 4 process is characterized as a "quantitatively managed process." [CL105] A quantitatively managed process is a defined (capability level 3) process that is controlled using statistical and other quantitative techniques. Quantitative objectives for quality and process performance are established and used as criteria in managing the process. The quality and process performance are understood in statistical terms and are managed throughout the life of the process. [CL105.N111]" [CMMI02A] (p. 49)

5 OPTIMIZING: "A capability level 5 process is characterized as an "optimizing process." An optimizing process is a quantitatively managed (capability level 4) process that is changed and adapted to meet relevant current and projected business objectives. An optimizing process focuses on continually improving the process performance through both incremental and innovative technological improvements. Process improvements that would address root causes of process variation and measurably improve the organization's processes are identified, evaluated, and deployed as appropriate. These improvements are selected based on a quantitative understanding of their expected contribution to achieving the organization's process-improvement objectives versus the cost and impact to the organization. The process performance of the organization's processes is continually improved.[CL106.N107]" [CMMI02A] (p. 53)

PROCESS AREA: "is a cluster of related practices in an area that, when performed collectively, satisfy a set of goals considered important for making significant improvement in that area. All CMMI process areas are common to both continuous and staged representations. In the continuous representation, process areas are organized by process area categories. [FM103.HDA102.HDB101.T116]" [CMMI02A] (p. 14)

SPECIFIC GOALS: "apply to a process area and address the unique characteristics that describe what must be implemented to satisfy the process area. Specific goals are required model components and are used in appraisals to help determine whether a process area is satisfied. There can be specific practices at different capability levels mapped to the same goal. However, every goal has at least one capability level 1 practice mapped to it. [FM103.HDA102.HDB103.T102]" [CMMI02A] (p. 15)

GENERIC GOALS: "Each capability level (1-5) has only one generic goal that describes the institutionalization that the organization must achieve at that capability level. Thus, there are five generic goals; each appears in every process area. Achievement of a generic goal in a process area signifies improved control in planning and implementing the processes associated with that process area thus indicating whether these processes are likely to be effective, repeatable, and lasting. Generic goals are required model components and are used in appraisals to determine whether a process area is satisfied. (Only the generic goal title and statement appear in the process areas.) [FM103.HDA102.HDB105.T103]" [CMMI02A] (p. 16)

SPECIFIC PRACTICES: "is an activity that is considered important in achieving the associated specific goal. The specific practices describe the activities expected to result in achievement of the specific goals of a process area. Every specific practice is associated with a capability level. Specific practices are expected model components. [FM103.HDA102.HDB104.T102]" [CMMI02A] (p. 15)

GENERIC PRACTICES: "Generic practices provide institutionalization to ensure that the processes associated with the process area will be effective, repeatable, and lasting. Generic practices are categorized by capability level and are expected components in CMMI models. In the continuous representation, each generic practice maps to one generic goal. (Only the generic practice title, statement, and elaborations appear in the process areas.) [FM103.HDA102.HDB107.T103]" [CMMI02A] (p. 17)

PRACTICES: there are two kinds of practices: Specific and Generic.

BASE PRACTICES: "In the continuous representation, all the specific practices with a capability level of 1 are called "base practices." [FM103.HDA102.HDB111.T101]" [CMMI02A] (p.15)

ADVANCED PRACTICES: "In the continuous representation, all the specific practices with a capability level of 2 or higher are called "advanced practices." [FM103.HDA102.HDB112.T101]

For example, within the Requirements Management process area, "Develop an understanding with the requirements providers on the meaning of the requirements" is a capability level 1 specific practice, whereas "Obtain commitment to the requirements from the project participants" is a capability level 2 specific practices. [FM103.HDA102.HDB112.T102] Sometimes an advanced practice builds on a base practice. When this happens, the advanced practice is included in the staged representation as a specific practice, but the base practice is not. Rather, the base practice is presented as informative material after the specific practice. This informative material explains how the base and advanced practices appear in the continuous representation. Other times, an advanced practice does not build on a base practice. When this happens, the advanced practice is automatically included in the staged representation as a specific practice. [FM103.HDA102.HDB112.T103]" [CMMI02A] (p. 15)

GENERIC PRACTICE ELABORATIONS: "are informative model components that appear in each process area to provide guidance on how the generic practices should uniquely be applied to the process area. For example, when the generic practice "Train the people performing or supporting the planned process as needed" is incorporated into the Configuration Management process area, the specific kinds of training for doing configuration management are described. [FM103.HDA102.HDB116.T101] " [CMMI02A] (p. 17)

TYPICAL WORK PRODUCTS: "are an informative model component that provides example outputs from a specific or generic practice. These examples are called "typical work products" because there are often other work products that are just as effective, but are not listed. [FM103.HDA102.HDB113.T101] " [CMMI02A] (p. 16)

SUBPRACTICES: "are detailed descriptions that provide guidance for interpreting specific or generic practices. Subpractices may be worded as if prescriptive, but are actually an informative component in CMMI models meant only to provide ideas that may be useful for process improvement. [FM103.HDA102.HDB114.T101]" [CMMI02A] (p. 16)

DISCIPLINE AMPLIFICATIONS: "are informative model components that contain information relevant to a particular discipline and are associated with specific practices. For example, if in the CMMI-SE/SW model, you want to find a discipline amplification for software engineering, you would look in the model for items labeled "For Software Engineering." The same is true for other disciplines. [FM103.HDA102.HDB115.T101] " [CMMI02A] (p. 18)

DISCIPLINE: "The word "discipline," when used in the CMMI Product Suite, refers to the bodies of knowledge available to you when selecting a CMMI model (e.g., systems engineering). The CMMI Product Team envisions that other bodies of knowledge will be integrated into the CMMI Framework.[FM114.HDA102.HDB113.T101]" [CMMI02A] (p. 24)

REFERENCES: "are informative model components that direct the user to additional or more detailed information in related process areas. Typical phrases expressing these pointers are "Refer to the Decision Analysis and Resolution process area for determining the best integration strategy" or "Refer to the Project Planning process area for more information about global project planning." All references are clearly marked in italics. [FM103.HDA102.HDB117.T101]" [CMMI02A] (p. 18)

PROCESS AREA CATEGORY: "there are four categories: [FM102.HDA101.T102] Process Management, Project Management, Engineering, Support." [CMMI02A] (p. 57)

4.6.3 Modelling notes

A *Capability Level* is the parent class of six levels (0 to 5) so it is presented as a generalization. By modeling all the child, whose the first *0 Incomplete*, the multiplicity of different association are affected. First, the multiplicity of the association between *Capability Level* and *Specific Practice* is 0..*. Second, the multiplicity of the association between *Capability Level* and *Generic Goal* is 0..1.

Although the *Goal* class is not defined in the model, we assume that *Specific Goal* and *Generic Goal* are specifications of the parent class *Goal*.

Practice is the generalization of *Specific practice* and *Generic practice*.

4.7 CMMI for software engineering 1.1 - Staged Representation

4.7.1 Class Diagram of CMMI 1.1 - Staged Representation

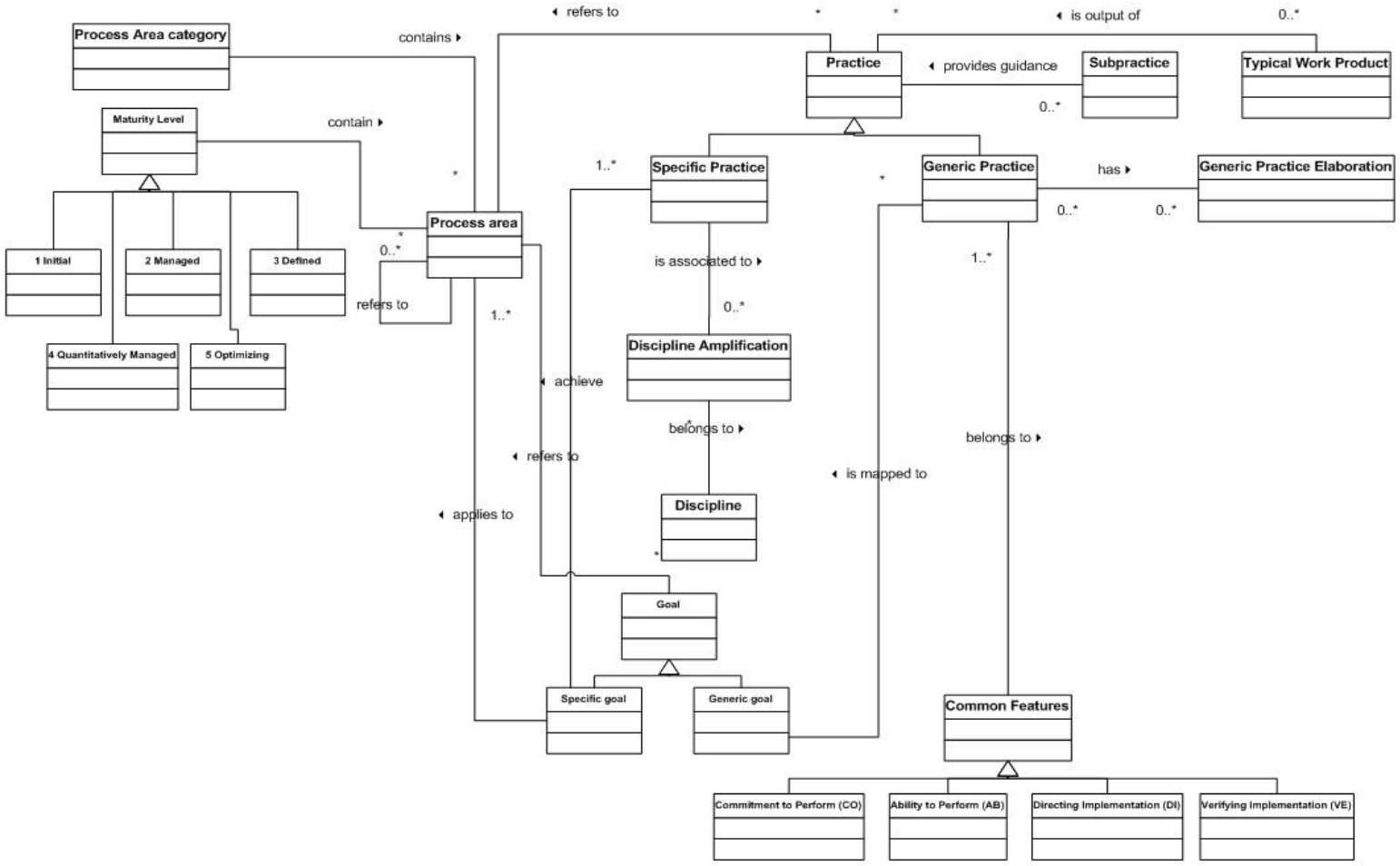

Figure 12 Class Diagram of CMMI 1.1 - Staged Representation

4.7.2 Definitions of classes

MATURITY LEVEL: "is a defined evolutionary plateau of process improvement. Each maturity level stabilizes an important part of the organization's processes.(…) In CMMI models with a staged representation, there are five maturity levels, each a layer in the foundation for ongoing process improvement, designated by the numbers 1 through 5" [CMMI02B] (p. 10)

1 INITIAL: "At maturity level 1, processes are usually ad hoc and chaotic. The organization usually does not provide a stable environment. Success in these organizations depends on the competence and heroics of the people in the organization and not on the use of proven processes. In spite of this ad hoc, chaotic environment, maturity level 1 organizations often produce products and services that work; however, they frequently exceed the budget and schedule of their projects.[FM103.HDA101.HDB104.T101]" [CMMI02B] (p. 11)

2 MANAGED: "At maturity level 2, an organization has achieved all the specific and generic goals of the maturity level 2 process areas. In other words, the projects of the organization have ensured that requirements are managed and that processes are planned, performed, measured, and controlled. [FM103.HDA101.HDB105.T101] " [CMMI02B] (p. 11)

3 DEFINED: "At maturity level 3, an organization has achieved all the specific and generic goals of the process areas assigned to maturity levels 2 and 3. At maturity level 3, processes are well characterized and understood, and are described in standards, procedures, tools, and methods.[FM103.HDA101.HDB106.T101]" [CMMI02B] (p. 12)

4 QUANTITATIVELY MANAGED: "At maturity level 4, an organization has achieved all the specific goals of the process areas assigned to maturity levels 2, 3, and 4 and the generic goals assigned to maturity levels 2 and 3. Subprocesses are selected that significantly contribute to overall process performance. These selected subprocesses are controlled using statistical and other quantitative techniques. [FM103.HDA101.HDB107.T101]" [CMMI02B] (p. 13)

5 OPTIMIZING: "At maturity level 5, an organization has achieved all the specific goals of the process areas assigned to maturity levels 2, 3, 4, and 5 and the generic goals assigned to maturity levels 2 and 3. Processes are continually improved based on a quantitative understanding of the common causes of variation3 inherent in processes. [FM103.HDA101.HDB108.T101] " [CMMI02B] (p. 13)

PROCESS AREA: "is a cluster of related practices in an area that, when performed collectively, satisfy a set of goals considered important for making significant improvement in that area. All CMMI process areas are common to both continuous and staged representations. In the staged representation, process areas are organized by maturity levels.[FM103.HDA102.HDB101.T101]" [CMMI02B] (p. 17)

SPECIFIC GOALS: "apply to a process area and address the unique characteristics that describe what must be implemented to satisfy the process area. Specific goals are required model components and are used in appraisals to help determine whether a process area is satisfied. [FM103.HDA102.HDB103.T101]" [CMMI02B] (p. 17)

GENERIC GOALS: "are called "generic" because the same goal statement appears in multiple process areas. In the staged representation, each process area has only one generic goal. Achievement of a generic goal in a process area signifies improved control in planning and implementing the processes associated with that process area, thus indicating whether these processes are likely to be effective, repeatable, and lasting. Generic goals are required model components and are used in appraisals to determine whether a process area is satisfied. (Only the generic goal title and statement appear in the process areas.) [FM103.HDA102.HDB105.T101]" [CMMI02B] (p. 18)

PRACTICES: there are two kind of practices: Specific and Generic.

SPECIFIC PRACTICES: "is an activity that is considered important in achieving the associated specific goal. The specific practices describe the activities expected to result in achievement of the specific

goals of a process area. Specific practices are expected model components.[FM103.HDA102.HDB104.T101]" [CMMI02B] (p. 17)

GENERIC PRACTICES: "provide institutionalization to ensure that the processes associated with the process area will be effective, repeatable, and lasting. Generic practices are categorized by generic goals and common features and are expected components in CMMI models. (Only the generic practice title, statement, and elaborations appear in the process areas.) [FM103.HDA102.HDB107.T101] " [CMMI02B] (p. 19)

GENERIC PRACTICE ELABORATIONS: "are informative model components that appear in each process area to provide guidance on how the generic practices should uniquely be applied to the process area. For example, when the generic practice "Train the people performing or supporting the planned process as needed" is incorporated into the Configuration Management process area, the specific kinds of training for doing configuration management are described. [FM103.HDA102.HDB116.T101] " [CMMI02B] (p. 19)

COMMON FEATURES: "Four common features organize the generic practices of each process area. Common features are model components that are not rated in any way. They are only groupings that provide a way to present the generic practices. (...) [FM103.HDA102.HDB106.T101] " [CMMI02B] (p. 17)

COMMITMENT TO PERFORM (CO): "groups the generic practices related to creating policies and securing sponsorship. [FM122.HDA103.T104]" [CMMI02B] (p. 36)

ABILITY TO PERFORM (AB): "groups the generic practices related to ensuring that the project and/or organization has the resources it needs.[FM122.HDA103.T105]" [CMMI02B] (p. 36)

DIRECTING IMPLEMENTATION (DI): "groups the generic practices related to managing the performance of the process, managing the integrity of its work products, and involving relevant stakeholders. [FM122.HDA103.T106]" [CMMI02B] (p. 36)

VERIFYING IMPLEMENTATION (VE): "groups the generic practices related to review by higher level management and objective evaluation of conformance to process descriptions, procedures, and standards. [FM122.HDA103.T107]" [CMMI02B] (p. 36)

TYPICAL WORK PRODUCTS: "are an informative model component that provides example outputs from a specific or generic practice. These examples are called "typical work products" because there are often other work products that are just as effective, but are not listed. [FM103.HDA102.HDB113.T101] " [CMMI02B] (p. 18)

SUBPRACTICES: "are detailed descriptions that provide guidance for interpreting specific or generic practices. Subpractices may be worded as if prescriptive, but are actually an informative component in CMMI models meant only to provide ideas that may be useful for process improvement. [FM103.HDA102.HDB114.T101] " [CMMI02B] (p. 18)

DISCIPLINE AMPLIFICATIONS: "are informative model components that contain information relevant to a particular discipline and are associated with specific practices. For example, if in the CMMI-SE/SW model, you want to find a discipline amplification for software engineering, you would look in the model for items labeled "For Software Engineering." The same is true for other disciplines. [FM103.HDA102.HDB115.T101] " [CMMI02B] (p. 18)

DISCIPLINE: "The word "discipline," when used in the CMMI Product Suite, refers to the bodies of knowledge available to you when selecting a CMMI model (e.g., systems engineering). The CMMI Product Team envisions that other bodies of knowledge will be integrated into the CMMI Framework.[FM114.HDA102.HDB113.T101]" [CMMI02B] (p. 24)

REFERENCES: "are informative model components that direct the user to additional or more detailed information in related process areas. Typical phrases expressing these pointers are "Refer to the Organizational Training process area for more information about identifying training needs and providing the necessary training" or "Refer to the Decision Analysis and Resolution process area for more information about evaluating and selecting among alternatives." All references are clearly marked in italics. [FM103.HDA102.HDB117.T102] " [CMMI02B] (p. 19)

PROCESS AREA CATEGORY: there are four categories: [FM102.HDA101.T102] Process Management, Project Management, Engineering, Support. [CMMI02B] (p. 47)

4.7.3 Modelling notes

A *Maturity Level* is the parent class of five level (1 to 5) so it is presented as a generalization. By modeling all the child, whose the first *1 Initial*, the multiplicity of the association between *Maturity Level* and *Process Area* is 0..* because the *Initial* level does not contains any *Process Area*.

Although the *Goal* class is not defined in the model, we assume that *Specific Goal* and *Generic Goal* are specifications of the parent class *Goal*.

Practice is the generalization of *Specific practice* and *Generic practice*.

5 Conclusion

The modeling of models coming from different sources with the same technique provides several advantages and difficulties. Among advantages the comprehension and comparison of models are facilitated. The class diagram provides a quick overview of the model structure that allows a better comprehension of the model concepts. The main difficulty lies in the class definition when the model does not provide clear definition of its components.

Our future works:

- modeling of other process models
- detailed comparison of the modeling
- to derive a meta model from these particular models

6 References

[AHER01]

Ahern, D., Clouse, A., Turner, R., **CMMI℠ Distilled. A Practical Introduction to Integrated Process Improvement**, Addison-Wesley, SEI Series in Software Engineering, London, 2001.

[CHRI03]

Chrissis, M.B., Konrad, M., Shrum, S., **CMMI© Guidelines for Process Integration and Product Improvement**, Addison-Wesley, SEI Series in Software Engineering, Boston, 2003.

[CMMI02a]

"*Capability Maturity Model® Integration (CMMISM), Version 1.1, CMMISM for Software Engineering (CMMI-SW, V1.1) Continuous Representation*", CMU/SEI-2002-TR-028, Software Engineering Institute, Pittsburgh, 2002.

[CMMI02b]

"*Capability Maturity Model® Integration (CMMISM), Version 1.1, CMMISM for Software Engineering (CMMI-SW, V1.1) Staged Representation*", CMU/SEI-2002-TR-029, Software Engineering Institute, Pittsburgh, 2002.

[HAAS94]

Haase, V., Messnarz, R., Koch, G., Kugler, H., Decrinis, P., "Bootstrap: Fine-Tuning Process Assessment," in *IEEE Software*, Vol.11, n°4, 1994, p.25-35.

[HUMP89]

Humphrey, W., **Managing the Software Process**, Addison-Wesley, Reading, MA, 1989.

[ISO98a]

ISO/IEC, "ISO/IEC TR 15504 : Information Technology – Software Process Assessment (parts 1-9)", International Organization for Standardization and the International Electrotechnical Commission, 1998.

[ISO98b]

ISO/IEC, "ISO/IEC 15504-2: 1998 Information Technology- Software Process Assessment – Part2: A reference model for processes and process capability", International Organization for Standardization and the International Electrotechnical Commission, 1998.

[ISO98c]

ISO/IEC, "ISO/IEC 15504-5: 1998 Information Technology- Software Process Assessment – Part5: Software process assessment", International Organization for Standardization and the International Electrotechnical Commission, 1998.

[ISO98d]

ISO/IEC, "ISO/IEC 15504-9: 1998 Information Technology- Software Process Assessment – Part9: Vocabulary", International Organization for Standardization and the International Electrotechnical Commission, 1998.

[KUVA99]

Kuvaja, P., "BOOTSTRAP 3.0—A SPICE1 Conformant Software Process Assessment Methodology", in *Software Quality Journal*, Vol. 8, n°1, 1999, pp. 7-19.

[LEPA02]

Lepasaar, M., Mäkinen, T., Varkoi, T., Structural comparison of SPICE and continuous CMMI. In Proceedings of SPICE 2002, (2002), Venice, Italy, 223-234.

[PAUL93a]

Paulk, M., Curtis, B., Chrissis, M., Weber, Ch., "*Capability Maturity Model for Software. Version 1.1*," Software Engineering Institute, Carnige Mellon University, Pittsburgh, PA, CMU/SEI-93-TR-024, 1993.

[PAUL93b]

Paulk, M., Curtis, B., Chrissis, M.B., and Weber, C., "Capability Maturity Model, Version 1.1," in *IEEE Software*, Vol. 10, No. 4, 1993, pp. 18-27.

[PAUL99a]

Paulk, M. et al, **The Capability Maturity Model. Guidelines for Improving the Software Process**, Carnige Mellon University, Software Engineering Institute, Addison-Wesley, 1999.

[STIE97]

Stienen, H., Engelmann, F., Lebsanft, E., "BOOTSTRAP: Five Years of Assessment Experience," in *Proceedings of the 8th International Workshop on software technology and Engineering Practice (STEP '97)*, p.371-379.

[TRIL94]

"TRILLIUM. Model for Telecom Product Development & Support Process Capability", Release 3.0, Bell Canada, Canada, 1994.

[UML03]

Unified Modeling Language: http://www.omg.org/technology/documents/formal/uml.htm 12.02.2003

[ZAHR97]

Zahran, S., **Software Process Improvement. Practical Guidelines for Business Success**, Addison-Wesley, 1997.