5. CONCEPTUAL APPROACH

Practical sciences proceed by building up; theoretical sciences by resolving into components.

(Saint Thomas Aquinas)

Contents

This chapter describes the proposed approach for a systematic tailoring of metrics based on the concept of a metric framework. The metric framework is the central element of the approach since it offers the metric frames from which the different metrics are built.

First, the requirements of this approach are described, including the requirements such an approach has to support. Next, the approach is explained. This approach is called **Abstract Interpretation**. The idea comes from the fact that a metric framework can be seen as an abstraction from the concrete metrics. The abstract interpretation is a bi-dimensional approach. One dimension (or domain) constitutes the representation of the metric framework in an *abstract domain*. Project specific metrics belong to the second domain: the *concrete domain*. To conclude the chapter, the tailoring process and an example on its use are covered.

5.1. Requirements

In Section 1.1, the objectives of the work were summarized. Based on these objectives, the functional and non functional requirements for the solution which should be developed are covered here. First, the approach used for the tailoring of organizational wide defined metrics should be able to address variability management by covering the next issues:

1. The definition of variation points in the metrics.

2. The definition of variants, in case of any, that can be potentially bound to these variation points.

3. The definition of the constraints and dependencies among variation points and variants.

4. A mechanism to specify variants.

5.1.1. Non-Functional Requirements

Based on the requirements previously described, some further requirements on the approach are derived.

Intuitive. As any type of approach, it should be based on concepts that have a direct correspondence in the domain of metrics. The way these concepts are used and combined, should be logically consistent with the view of a metrics expert.

Reusable. The approach should prove to be reusable during tailoring of metrics. This requires a delicate balance between falling in a too general or too specific approach. It also means that the dimension of the approach has to be chosen so that it can be taken as a guide; a guide that is applicable within any organization wishing to concentrate their metrics and make them available to its projects for tailoring.

Flexible. The approach should be allow an easy adaptation to deal with variability modeled with it.

5.2. Abstract Interpretation

The general purpose of the abstract interpretation is to provide with a formal representation of the metrics in an organization. The abstract interpretation is an approach that establishes a correlation between the collection of metrics from an organization and a metric framework. It also shows a dependency from the project specific metrics to the metric framework in a sense that these metrics are derived from it.

Figure 5.1 shows a macroscopic view of the abstract interpretation, conformed by two domains and is based on the differentiation between the organizational and project levels.

Abstract Domain (AD). This domain covers aspects regarding the structure of the frames from the metric framework, how choices are modeled, which kind of abstraction is used to manage possible dependencies and relations between those frames. AD represents the organization level and it is hierarchically organized in two layers.

Metric Meta-Model (MM). This layer (depicted in the upper part of Figure 5.1) provides the representation of a meta-model that constraints the construction of applicable and useful metrics. This layer also provides a

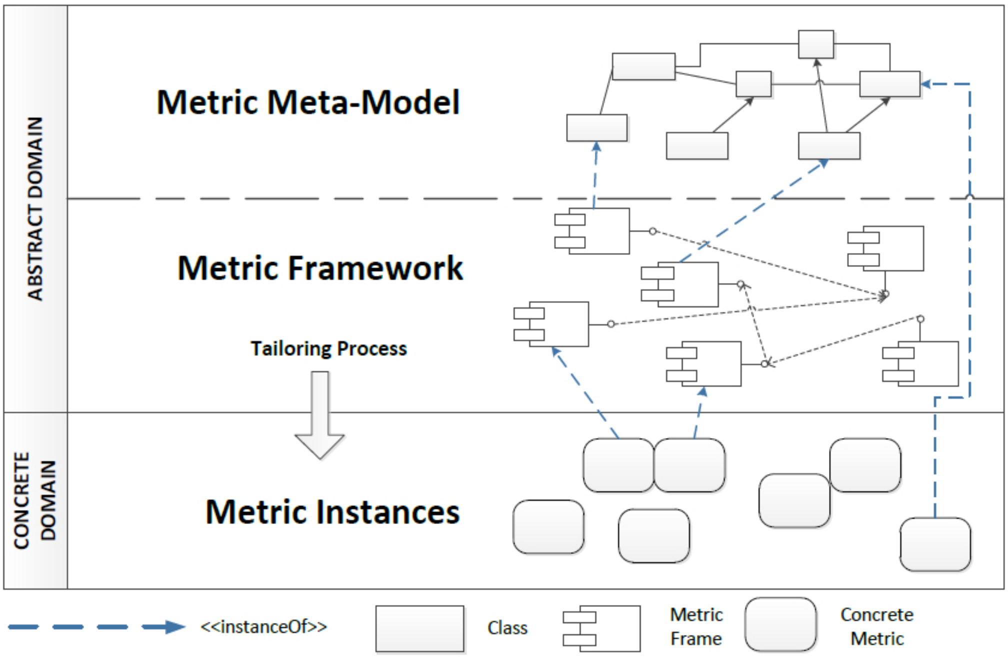

Figure 5.1.: Overview of the abstract interpretation

kind of grammar and lexicon (to comply with its referenced metric meta-model). For the purpose of this approach, the current layer is said to be complete with respect to the definition of metrics.

Metric Framework (MF). This layer (depicted in the middle part of Figure 5.1) consists of elements denominated metric frames, which represent a collection of entities from the metric meta-level layer. MF is referred as the semantic representation.

Concrete Domain (CD). This domain constitutes all the collection of metrics from every project within an organization. CD contains the entities of measurement and it is composed of one layer.

Metric Instance (MI). This layer (depicted in the lower part of Figure 5.1) contains local collections of instantiated metrics. It is in this layer where the metrics define a specific purpose according to the projects information needs.

Besides the composition from the domains, another aspect to consider is the communication that exists between them. This communication (depicted as an arrow in Figure 5.1) is indeed the tailoring process and is unidirectional, going from the AD to the CD via a tailoring process. The tailoring process only occurs between the metric framework and the metric instance. It is also possible to establish a direct communication between the metric meta-model and the metric instance. This communication is illustrated in Figure 5.1 with a blue dotted line.

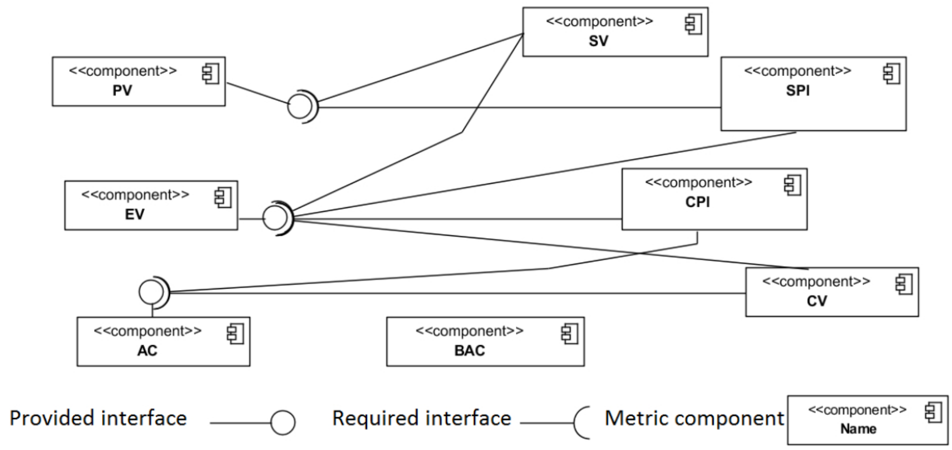

Figure 5.2.: UML component diagram used to represent a metric framework

5.2.1. Metric Meta-Model

The formal representation of metrics is based on the metric meta-model from the MeDIC Tool (see Section 2.2.3). The metric meta-model consists of elements covering the terms and concepts proposed by the measurement information model. Moreover, the metric meta-model has an extension. This extension is used to represent aspects related to reporting, e.g. the meta-model element *report* is used to model a project cockpit. The project cockpit includes measurement constructs as means of visualization, which are modeled with the meta-model element *report item*.

5.2.2. Metric Framework

A metric framework provides the necessary support for representation of the knowledge related to metrics in a software organization. A metric framework comprises a collection of pre-fabricated metric frames that are intended to be reused. The metric frames follow an implicit classification scheme. This scheme is based on two points: first, the type of project where the metrics are used. For example, an organization has two types of Software Engineering projects: small and large. Second, the type of metric frames are base and derived (just as the type of possible metrics). Thus, the inter-relationships that may exist between metric frames can only occur within projectsof the same type.

When thinking of metric frameworks, it is sometimes useful to apply analogies of other domains where the concept of a metric frame is better understood. For example, in the context of software systems the metric framework can be thought of as an interconnected collection of components. Thus, a metric component encapsulates its content and defines its behavior in terms of provided and required interfaces.

An example of such analogy is depicted in Figure 5.2. The UML component diagram demonstrates the existence of some metric components and their inter-relationships. Metric components are represented as a rectangle with the stereotype «`component`» and a component icon in the upper-right corner. Assembly connectors "link" the provided interfaces supplied by metric components *Planned Value* (PV), *Earned Value* (EV) and *Actual Cost* (AC) to required interfaces specified by metric components *Schedule Variance* (SV), *Schedule Performance Index* (SPI), *Cost Performance Index* (CPI) and *Cost Variance* (CV). Additionally, it exists one more metric component (*Budget at Completion* (BAC)), which on this example neither provides an interface nor requires one. Appendix A provides information about these metrics.

As a remark, it is assumed that all the metric components that are inter-related belong to the same type of project during their selection. The types of Software Engineering projects are either small or large. Nevertheless, unlike components, a metric frame does not have such a composition. For the sake of clarity, the idea of metric frame prevails.

5.2.3. Metric Instances

Metric instances represent the entities of measurement used to support the project goals and needs. These instances are constructed by using the metric frames during the tailoring process or directly from the metric meta-model. The metric instances that result from the tailoring (of metric frames) and/or direct construction (from the metric meta-model) are the same. The reasons for using one or another mechanism rely on the different needs of the project. These reasons are introduced during the tailoring process.

5.3. Metric Frame

Every metric frame belongs to the metric framework, which is located on the abstract domain. Thus, a metric frame is an abstraction that provides with a structure conformed with two parts: a common and a variant part. In the metric frame, the concept of choice refers to *variant*[1]. The variants are used to refine the variability.

Figure 5.3 shows the structure of a metric frame. The upper part represents the core that remains unmodified and it consists of four categories: metric, information needs, interpretation aids and reporting (see Section 2.2.1). The underlying part forms the variant part. Thus, a formal representation of variability is required in order to:

- Understand what is varying, which variation points include, what are the variants corresponding to each variation point.
- Analyze the causes behind the variability. For instance, stakeholder needs or realization of other variability dependencies.

[1] See Section 2.4.1.

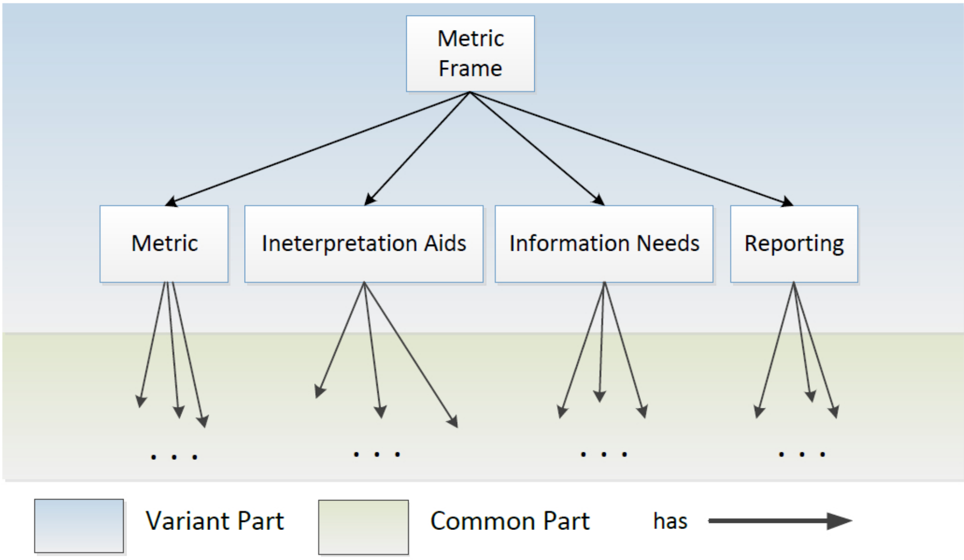

Figure 5.3.: Metric frame structure

- Understand how metric frames vary. By modeling the metric frames, it is possible to see whether there are dependencies between variation points and/or variants.

5.3.1. Variability Model

The metric framework requires a representation from its variation points and their respective variants. Variability modeling can be achieved in different ways (see Section 2.4.2). This section incrementally introduces the elements required to build the variability meta-model depicted in Figure 5.4.

The core elements from the variability model are the variation points, variants and their relations. A **Variation point** defines the conditions where the variation from the metric frame can be adjusted. The variation point is classified in two types, which are either external or internal.

External variation point. An external variation point implies that is visible to *users*[2], which means the users can choose the variants they need. The external variation point has associated variants that are visible to the user and *metrics expert*[2]. An example of this type is the variation point "Entity of Measurement", used to categorize the projects into large or small types.

Internal variation point. This is the second type of variation point, where all the decisions that concern defining and resolving internal variability are within the

[2] A distinction between stakeholders is required. A *user* can be a project manager or another person within a project. A *metrics expert* is the analyst that understands the metrics within an organization.

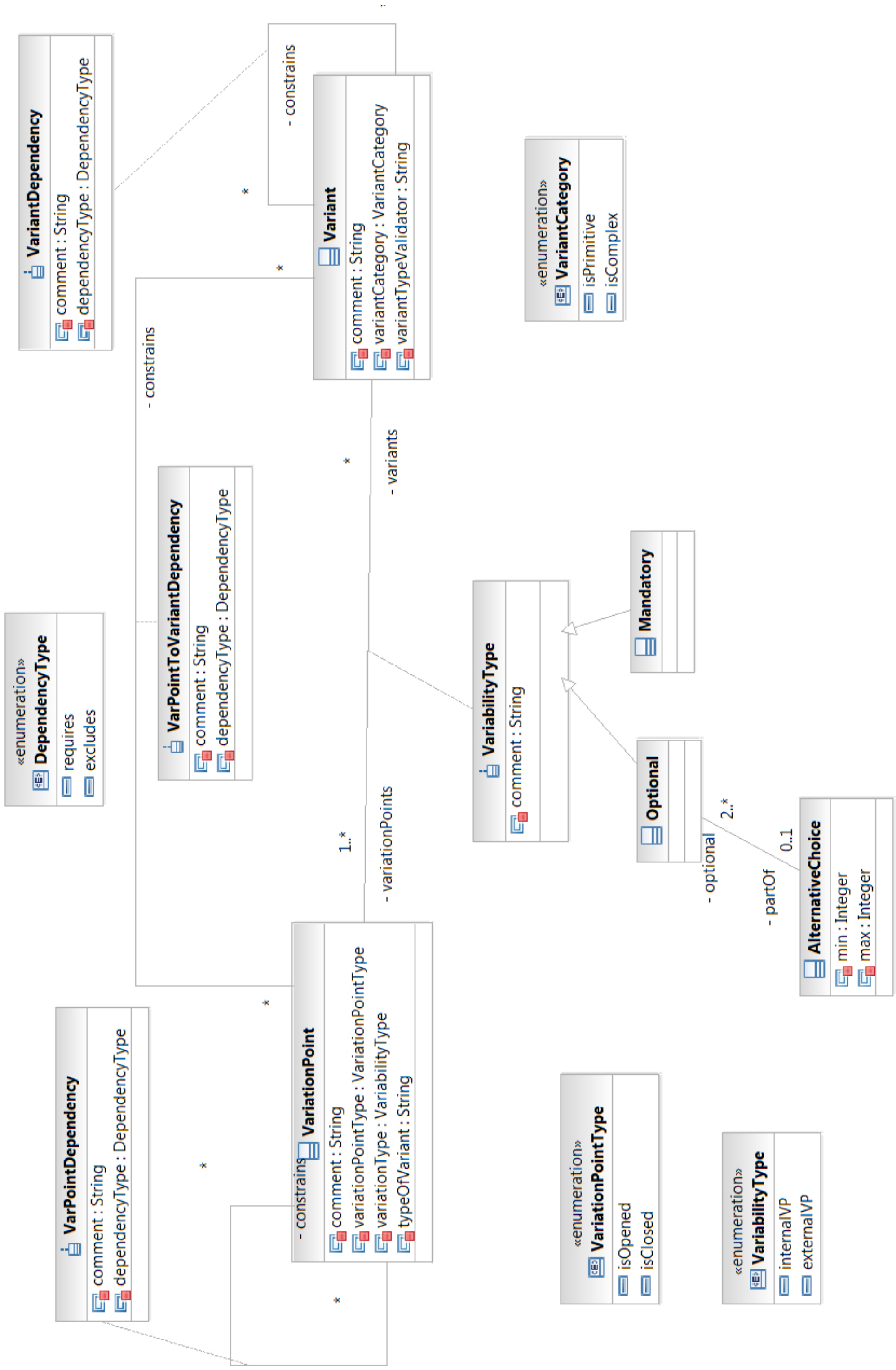

Figure 5.4.: Variability meta-model

responsibility of the metrics expert. Therefore, this type of variation point is only visible to the metrics expert. For example, a variation point "Scale Type" (see Section 2.2.1) is internal and it concerns only to the metrics expert.

Further on, every variation point is either opened or closed. This attribute states the possibility or restriction on providing or selecting a variant.

Variation point is opened. It means that the variants are selected from the available set or are provided by the user, e.g. a variation point "Storage Information", used to specify the location from the metrics within a project, cannot be fixed for every project.

Variation point is closed. The selection of a variant is only permitted from the provided set, e.g. a variation point "Measure Interval", which is used to specify the frequency of the measureemnt, has predefined variants like daily, weekly, monthly. It is nos possible to add more variants but only select one of the provided.

The last attribute for a variation point is a comment, depicted in Figure 5.4. A comment allows to textually describe the rational for introducing the element.

Variant is the representation of variability, which allows the tailoring of metrics and it is delayed until the end, when specific information needs must be met. A variant has an attribute used to specify the type of entity from the metric meta-model. Additionally, the variants follow in two categories: primitive or complex.

Primitive. A variant of the category primitve is represented by a number or character, e.g. the function used to calculate CPI. The type used to represent the function in the metric meta-model is String.

Complex. A variant of this category is a variant that is not primitive, e.g. variants *daily*, *weekly* and *monthly*, from the variation point "Measure Interval", fall on this category. These variants are of type *Interval Type* in the metric meta-model.

The next aspect to review is the type of **relations** that exist between the variation points and the variants. The relations defined on this variability model were inspired by FODA (see Section 2.4.2). A relation type is the association between the variation point and the variant. The association states that a variation point offers a certain variant. The multiplicities of the associations imply the following:

- A variation point can be associated with zero or more variants.
- All the variants must be associated with at least one variation point.
- A variation point can offer more than one variant.
- A variant can be associated with more than one variation point.

The relation type has three types of variability, which are optional, mandatory or alternative choice. This variability on the relations describes what attributes of the metric frames must be included as part of the metric and which could be left out.

Optional variability. It states that the variant that is part of a variation point can be selected during tailoring but does not need to be tailored, e.g. the variation point "Measure Class" has variants "process metric" and "product metric"; the variation point is used to specify whether the metric is used to measure a process, a product or both of them. The variaton point provides information about the context where the metric is applied. However, it is not mandatory.

Mandatory variability. It states that a variant must be selected if and only if the associated variation point is part of the metric, e.g. the variation point "Storage Information" is required in order to know the source where the information is located.

Alternative choice variability. It groups a set of variants that are related to the same variation point and a range defines how many variants it is allowed to select. The range of permissible variants is specified with a minimum and a maximum values, e.g. the variation point "Information Need" is used to group the different needs into categories like cost, risk, time, quality and content. The selection of variants, in this example, requires at least one variant to be selected and a maximum of two variants. The multiplicities from the alternative choice enforce the following conditions:

- Alternative choice groups at least two optional variants.
- Each optional variant may be part of at most one alternative choice.

For the case where no range is provided, exactly one of the variants has to be selected.

Another aspect that belongs to the variability model is the type of **dependencies** that exist between the variation points and the variants. The dependencies are used to define constraints on the metric frames and are classified in two types.

Requires. This dependecy states that the elements associated with this type must be part of a metric frame either when one of them is choosed.

Excludes. This dependecy states that the elements associated with this type cannot belong to a metric frame simultaneously.

These dependencies can happen between variation points, variants and variants to variation points.

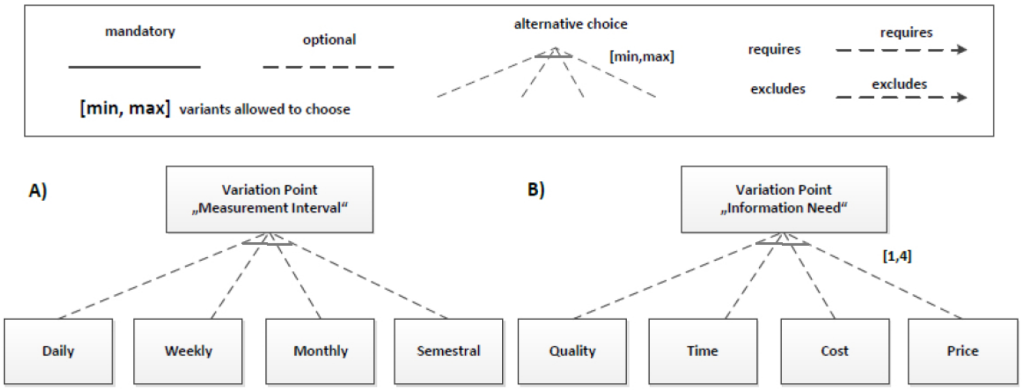

Figure 5.5.: Graphical notation for metric frames. A) Variation Point "Measure Interval" and B) Variation Point "Information Need"

5.3.2. Graphical Representation

To be able to graphically represent the variability model introduced in the previous section, an association from the types of variability and the dependencies with a graphical notation is established as depicted in the upper part from Figure 5.5. The graphical notation for the alternative choice includes the range of minimum and maximum values enclosed by square brackets ([min,max]). The asterisk character enclosed by square brackets matches zero or more variants. As stated before, the default value is [1,1] and for such cases, the range is dismissed from the graphical notation.

Figure 5.5 also includes two examples of variation points: A) The variation point "Measurement Interval" includes variants daily, weekly, monthly and semestral as alternative choice. Since the range has been omitted, only one variant can be selected. B) The variation point "Information Need" includes variants quality, cost, time, price and content as alternative choice and it states that at least one variant should be selected. It is also possible to select the four variants.

5.4. Tailoring Process

The process followed by the stakeholders during creation of metrics can be classified in to at least two scenarios. The outcome of both scenarios are metrics that suit the needs of a particular project. The possible scenarios are described below:

The metrics for a project are created purely based on the metric meta-model. The rationale behind this scenario is that there are metrics only used for a short period of time and/or limited number of projects. For example, a project leader requires creating a metric *review*, but his project uses *pair programming*[3]. The

[3]Software development technique where two programmers work together in the same workstation. One programmer is typing the code, while the other is reviewing every line of code that is written. The

project leader defines the metric review as the lines of code (LOC) that are developed during pair programming. This metric is used during a period of three or four months and then is archived.

The metrics for a project are tailored from the metric framework. The metrics that are extensively used should be included in a metric framework from the organization. In regards with the usage, there are two paths to follow:

Use the metric frame "as-is". The case where the metric frame is selected and applied without any adjustment. For example, the use of metric frames CPI and SPI direct from project management software programs.

Tailor the metric frame. The metric frame is adjusted before its use. For example, the metric frame EV is adjusted by selecting the choice "weekly" as its "measure interval".

The tailoring process defined here, covers only the second scenario where the framework is used and some adjustment is applied to the metric frames. The purpose of create a metric framework is to use it as a best practice library within an organization. One property from the metric framework is to standardize the structure of the metrics in the so-called metric frames, which represent pre-fabricated metrics. These metric frames are ready-to-use metrics and they can be tailored to satisfy particular circumstances or factors that surround a project. The tailoring process requires some input in order to follow some activities that return an outcome. The elements from the tailoring process are described below:

Input. A list of metric frames from the metricframework reflecting the information needs from a project.

Process. The process is decomposed into the following activities:

1. Select a metric frame that is required within a project. Additionally, the activity provides with information about the type of variation that is associated to the metric frame.

2. Make tailoring decisions in accordance with the information needs that were provided as an input to the metric frame from activity 1. This activity involves the selection and/or creation of variants from the different variation points contained in the metric frame.

3. In the case of inter-relationships between the metric frame from activity 1 and other metric frames, repeat the activity 2 for the metric frames involved. If the metrics frames were previously tailored, do not perform this activity.

4. Repeat the activities 1 through 3 until all the metrics from the list that was provided as an input are covered.

roles are switched frequently.

The activities included on this process should be accomplished by a stakeholder that is involved in the project and is aware of the circumstances that influence the tailoring.

Outcome. As a result of the successful tailoring process, a representation of the selected metric and its variability model reflecting the needs of a project is created. The representation is constituted by the variability part from the metric frame and the choices that were selected. That is, a list of variations points and the variants that were assigned to them during the tailoring process. The structure of this representation is described by the grammar listed below, where the variation points fragment represents the result of the tailoring process.

$$
\begin{array}{rcl}
MetricFrame & ::= & \textit{MetricFrameName}[< VariationPointList >] \\
< VariationPointList > & ::= & < VariationPoint >< VariationPointList > \\
 & & | < VariationPoint > |\texttt{null} \\
< VariationPoint > & ::= & (\textit{VariationPointName}, (< VariantList >)) \\
< VariantList > & ::= & \textit{VariantName}, < VariantList > |VariantName \\
 & & \textit{"VariantName"}|\texttt{null}
\end{array}
$$

5.4.1. Example

In order to follow the tailoring process covered in the previous section, it is useful to examine a simple example. The example introduced here, follows the format of the tailoring process.

Input. A large Software Engineering project wants to answer the question about whether the proposed budget is sufficient. It also wants to evaluate whether a deadline can be met. This information need is associated with time and cost and is satisfied by the metric frames CPI and SPI.

Process.

1. The metric frame CPI is selected. Some information about the type of variatio points that is associated to the metric frame CPI is included below:
   Type of Project: It classifies the type of project that uses the metric.
   Measure Interval: It indicates the frequency for the metric to be calculated.

2. **Type of project** has two choices, which are small and large. The second variant is selected.
   Measure Interval has four choices, which are daily, weekly, monthly and yearly; monthly is selected.

3. The metric frame CPI has inter-relationships with metric frames EV and AC. The activities 1 and 2 are repeated for these metric frames.

4. Activities 1 and 2 are repeated for the metric frame SPI. Activity 3 is repeated for metric frame PV.

Output. A representation of the metric frames CPI, SPI, EV, AV and PV that includes their three variation points and the variants assigned to them during the tailoring. The abstract syntax of the metric frames is listed below:

$$\begin{array}{rl} CPI & [(TypeofProject, (\text{"}large\text{"}))(MeasureInterval, (\text{"}monthly\text{"}))] \\ SPI & [(TypeofProject, (\text{"}large\text{"}))(MeasureInterval, (\text{"}monthly\text{"}))] \\ EV & [(TypeofProject, (\text{"}large\text{"}))(MeasureInterval, (\text{"}monthly\text{"}))] \\ AC & [(TypeofProject, (\text{"}large\text{"}))(MeasureInterval, (\text{"}monthly\text{"}))] \\ PV & [(TypeofProject, (\text{"}large\text{"}))(MeasureInterval, (\text{"}monthly\text{"}))] \end{array}$$