3.3. Quality Model Elements

The qmodel_classes Element contains the Elements with the information about all aspects of the Quality Model. Each of these aspects will be detailed in separate sections.
Visualització de models de qualitat mitjançant una interfície web

relation_scale_list, requirement_list,
requirement_pattern_roots,
requirement_pattern_list>

The sources_list Element contains the information about the sources in the Quality Model.

Example:

<qmodel_classes>
  <sources_list>
    <source id="1">ISO/IEC 9126-1:2001</source>
  </sources_list>
  <quality_entity_list> ... </quality_entity_list>
  <evaluation_list> ... </evaluation_list>
  <metrics_list> ... </metrics_list>
  <relation_list> ... </relation_list>
  <relation_kind_list> ... </relation_kind_list>
  <relation_scale_list> ... </relation_scale_list>
  <requirement_list> ... </requirement_list>
  <requirement_pattern_roots> ... </requirement_pattern_roots>
  <requirement_pattern_list> ... </requirement_pattern_list>
</qmodel_classes>

3.4. Quality Entity List Element

The quality_entity_list Element contains the Elements that represent all the Quality Entities existing in the Quality Model.

<!ELEMENT quality_entity_list (quality_entity*)>
The quality_entity Element has three attributes which specify Id, Entity kind and decomposition level. Please note that child Elements marked "?" are optional.

```
<!ELEMENT quality_entity (quality_entity_id,
   quality_entity_attributes,
   user, sources?, comments?,
   entity_successors?, entity_evaluations?,
   entity_requirements?, entity_relations?,
   entity_composition?)>

<!ATTLIST quality_entity id ID #REQUIRED>
<!ATTLIST quality_entity kind
   (characteristic|subcharacteristic|attribute
    |basic_attribute|derived_attribute) #REQUIRED>
<!ATTLIST quality_entity level (generic|specific)
   #REQUIRED>
```

The quality_entity_id Element contains the name of the Quality Entity.

```
<!ELEMENT quality_entity_id (#PCDATA)>
```

The quality_entity_attributes Element contains the name and explanation as the common Elements previously defined (user, sources and comments are also used). Please note the name is replicated, and always up to date with the name in the quality_entity_id Element.

```
<!ELEMENT quality_entity_attributes (name,explanation)>
```

The entity_successors, entity_evaluations, entity_relations and entity_requirements Elements contain lists of Elements, which have an Id attribute referencing another Element of the Model. They also have the name of the referenced Element in the #PCDATA content, and should be ignored to avoid inconsistency.

```
<!ELEMENT entity_successors (successor_id*)>
   <!ELEMENT successor_id (#PCDATA)>
   <!ATTLIST successor_id id IDREF #REQUIRED>
```
Visualització de models de qualitat mitjançant una interfície web

The entity_evaluations Element contains the id's of the Metrics Assignments of the Quality Entity.

```xml
<!ELEMENT entity_evaluations (evaluation_ref*)>
<!ELEMENT evaluation_ref (#PCDATA)>
<!ATTLIST evaluation_ref id IDREF #REQUIRED>
<!ELEMENT entity_requirements (requirement_ref*)>
<!ELEMENT requirement_ref (#PCDATA)>
<!ATTLIST requirement_ref id IDREF #REQUIRED>
<!ELEMENT entity_relations (relation_ref*)>
<!ELEMENT relation_ref (#PCDATA)>
<!ATTLIST relation_ref id IDREF #REQUIRED>
```

The entity_composition Element contains name and description Elements for the composition of the Entity.

```xml
<!ELEMENT entity_composition (name,description)>
```

Example:

```xml
<quality_entity id="1" kind="characteristic" level="specific">
  <quality_entity_id>Functionality</quality_entity_id>
  <quality_entity_attributes>
    <name>Functionality</name>
    <explanation>The capability of the software ...</explanation>
  </quality_entity_attributes>
  <user id="1">GESSI</user>
  <sources>
    <source id="1">ISO/IEC 9126-1:2001</source>
  </sources>
  <comments/>
  <entity_successors>
    <successor_id id="7">Suitability</successor_id>
    <successor_id id="8">Accuracy</successor_id>
    <successor_id id="9">Interoperability</successor_id>
    <successor_id id="10">Security</successor_id>
    <successor_id id="11">Functionality Compliance</successor_id>
  </entity_successors>
  <entity_evaluations>
    <evaluation_ref id="14">Metrics Assignment</evaluation_ref>
  </entity_evaluations>
</quality_entity>
```

150
3.5. Evaluation List Element (Metrics Assignments)

Quality Entities can have one or more metrics assignment, each one of which has one identifier and some additional information. The evaluation_list Element contains all this information related to Metrics Assignments of one Quality Entity.

Metric Assignments are the relationship between a Quality Entity (with a given parent Entity) and a Metrics.

"<!ELEMENT evaluation_list (evaluation*)>"+

A Metrics Assignment Element is composed of four childs, which are detailed below, and one Attribute for the Id.

The previous sources Element definition is reused for the last child.

<!ELEMENT evaluation (evaluation_id, evaluation_context, evaluation_attributes, sources?)>
<!ATTLIST evaluation id ID #REQUIRED>

The evaluation_id Element contains the name of the Metrics Assignment. This name is generated automatically, and does not report any information because it can be derived from the rest of the information.

<!ELEMENT evaluation_id (#PCDATA)>

The evaluation_context Element contains the references for the Metrics Assignment. As usual, each child Element contains an Id Attribute referencing the desired Element, and #PCDATA with the Name, which should be ignored to avoid inconsistency.
Visualització de models de qualitat mitjançant una interfície web

<!ELEMENT evaluation_context
(eval_entity, eval_predecessor, eval_metric)>
<!ELEMENT eval_entity (#PCDATA)>
<!ATTLIST eval_entity id IDREF #REQUIRED>
<!ELEMENT eval_predecessor (#PCDATA)>
<!ATTLIST eval_predecessor id IDREF #REQUIRED>
<!ELEMENT eval_metric (#PCDATA)>
<!ATTLIST eval_metric id IDREF #REQUIRED>

The evaluation_attributes Element contains an explanation for the Metrics Assignment, and a measuring protocol in the universal_property Element.

<!ELEMENT evaluation_attributes
(explanation, universal_property)>
<!ELEMENT universal_property (#PCDATA)>

Example:

<evaluation_list>
  <evaluation id="1">
    <evaluation_id> Metrics Assignment </evaluation_id>
    <evaluation_context>
      <eval_entity id="1">Attribute 1</eval_entity>
      <eval_predecessor id="11">Compliance</eval_predecessor>
      <eval_metric id="2">Operative Systems</eval_metric>
    </evaluation_context>
    <evaluation_attributes>
      <explanation/>
      <universal_property/>
    </evaluation_attributes>
    <comments/>
    <sources/>
  </evaluation>
  <evaluation id="2">
    <evaluation_id> Metrics Assignment </evaluation_id>
    <evaluation_context>
      <eval_entity id="2">Attribute 5</eval_entity>
      <eval_predecessor id="16">Suitability</eval_predecessor>
      <eval_metric id="2">Languages</eval_metric>
    </evaluation_context>
    <evaluation_attributes>
      <explanation/>
      <universal_property/>
    </evaluation_attributes>
    <comments/>
    <sources/>
  </evaluation>
</evaluation_list>
3.6. Metrics List Element

The metrics_list Element contains a list of all the Metrics Elements representing the Metrics existing in the Quality Model.

```xml
<!ELEMENT metrics_list (metric*)>
```

The metric Element contains all the information of one Metrics, and has two Attributes: one for the Id of the Metrics and another for the Metrics kind.

```xml
<!ELEMENT metric (metric_id,kind)>
<!ATTLIST metric id ID #REQUIRED>
<!ATTLIST metric kind (qualitative | formula | integer | real | boolean | string | domain | set | tuple | function) #REQUIRED>
```

The metric_id Element contains the name of the Metrics.

```xml
<!ELEMENT metric_id (#PCDATA)>
```

The kind Element contains one of the Metrics kind specific Elements, which have all necessary information related to the kind they represent. This Element is optional although it is not recommended to be missing in order to avoid incompleteness.

```xml
<!ELEMENT kind ( (qualitative | formula | integer | real | boolean | string|domain|set|tuple|function) ?)>
```

Example:

```xml
<metric id="8" kind="domain">
    <metric_id>Operative Systems</metric_id>
    <kind>
        <domain>
            ...
        </domain>
    </kind>
</metric>
```
Visualització de models de qualitat mitjançant una interfície web

Note that the kind Element must be coherent with the kind determined by the attribute in the metric Element

3.7. Simple Metrics Kind Elements

Metrics can be classified into simple and composite. Simple Metrics are single-valued, although almost every Metrics kind needs specific information that needs to be stored into the XML Model. The simple Metrics kinds are: Qualitative, Boolean, Integer, Real, String and Domain.

The qualitative Element is very similar to the boolean Element, and contains basically some of the previously defined Elements. These Metric kinds do not need any additional information.

```xml
<!ELEMENT qualitative (qualitative_attributes, comments, sources)>  
<!ELEMENT qualitative_attributes (name,explanation)>  
<!ELEMENT boolean (boolean_attributes, comments, sources)>  
<!ELEMENT boolean_attributes (name,explanation)>
```

Example:

```xml
<metric id="2" kind="boolean">
  <metric_id>Boolean Metrics</metric_id>
  <kind>
    <boolean>
      <boolean_attributes>
        <name>Boolean Metrics</name>
        <explanation>This Metrics....</explanation>
      </boolean_attributes>
      <comments/>
      <sources>
        <source id="1">ISO/IEC 9126-1:2001</source>
      </sources>
    </boolean>
  </kind>
</metric>
```

154
Visualització de models de qualitat mitjançant una interfície web

The integer and real Elements are equal and extend the qualitative and boolean Elements in order to store one maximum and minimum limit for the Metrics value (when evaluated). These Elements are called n and m respectively, and are optional childs.

```xml
<!ELEMENT n (#PCDATA)>
<!ELEMENT m (#PCDATA)>

<!ELEMENT integer (integer_attributes,comments,sources)>
   <!ELEMENT integer_attributes (name,explanation,n?,m?)>

<!ELEMENT real (real_attributes,comments,sources)>
   <!ELEMENT real_attributes (name,explanation,n?,m?)>
```

Example:

```xml
<metric id="3" kind="integer">
   <metric_id>Integer Metrics</metric_id>
   <kind>
      <integer>
         <integer_attributes>
            <name>Integer Metrics</name>
            <explanation>This Metrics....
         </explanation>
         <n>1</n>
      </integer_attributes>
      <comments/>
      <sources/>
   </kind>
</metric>
```

The string Element follows the previous structure. In this case, the additional information to store is a special value, available to use when evaluating. Note that the Element for the special value is optional.

```xml
<!ELEMENT special_value (#PCDATA)>

<!ELEMENT string (string_attributes,comments,sources)>
   <!ELEMENT string_attributes
         (name,explanation,special_value?)>
```

Example:

```xml
<metric id="4" kind="string">
   <metric_id>String Metrics</metric_id>
   <kind>
      <string>
         <string_attributes>
```

155
Visualització de models de qualitat mitjançant una interfície web

The domain Element also follows the previous structure. In this case, the additional information to store is the set of values in the domain. An attribute is also stored in order to determine whether the domain values are ordered or not. There is also an Element for the domain name, but it is not used actually, so it should be always empty. Note the name Element inside kind changes to metric_name in Elements of Domain Metrics.

```xml
<!ELEMENT metric_name (#PCDATA)>
<!ELEMENT domain_name (#PCDATA)>
<!ELEMENT order (#PCDATA)>
<!ELEMENT domain (domain_attributes, domain_value_list, comments, sources)>
<!ATTLIST domain ordered (true|false) #REQUIRED>
<!ELEMENT domain_attributes (metric_name, explanation, domain_name)>
<!ELEMENT domain_value_list (domain_value*)>
<!ELEMENT domain_value (name, order?)>

<metric id="6" kind="domain">
  <metric_id>Domain Metrics</metric_id>
  <kind>
    <domain ordered="true">
      <domain_attributes>
        <metric_name>Domain Metrics</metric_name>
        <explanation/>
        <domain_name/>
      </domain_attributes>
      <domain_value_list>
        <domain_value>
          <name>First Value</name>
          <order>1</order>
        </domain_value>
        <domain_value>
          <name>Second Value</name>
          <order>2</order>
        </domain_value>
        <domain_value>
          <name>Third Value</name>
          <order>3</order>
        </domain_value>
      </domain_value_list>
    </domain>
  </kind>
</metric>
```
3.8. Composite Metrics Kind Elements

Composite Metrics need to reference other Metrics in their definition. For instance, to define a Set Metrics it is necessary to specify the Metrics of the items in the Set. The composite Metrics kinds are: Set, Tuple, Function and Formula.

The metric_element Element will be used to reference other Metrics. It has one attribute for the Id of the Metrics, and its #PCDATA content is the name of that Metrics. This name is replicated inside the Model (it is in the corresponding metrics Element), and to avoid inconsistency, it should be ignored.

```xml
<!ELEMENT metric_element (#PCDATA)>
<!ATTLIST metric_element id IDREF #REQUIRED>"
```

The set Element follows the structure of simple Metrics kind Elements, and contains the reference to the Metrics of the items of the Set. There is also an Element for the set name, but it is not used actually, so it should be always empty. Note the name Element inside kind changes to metric_name in Elements of Set Metrics.
Visualització de models de qualitat mitjançant una interfície web

```xml
<!ELEMENT set_name (#PCDATA)>

<!ELEMENT set (set_attributes, metric_element, comments, sources)>
  <!ELEMENT set_attributes
    (metric_name, explanation, set_name)>

Example:

```xml
<metric id="8" kind="set">
  <metric_id>Set Metrics</metric_id>
  <kind>
    <set>
      <set_attributes>
        <metric_name>Set Metrics</metric_name>
        <explanation/>
        <set_name/>
      </set_attributes>
      <metric_element id="4">String Metrics</metric_element>
      <comments/>
      <sources/>
    </set>
  </kind>
</metric>
```

The tuple Element follows the structure of simple Metrics kind Elements, and contains a list of tuple items, each one with a name and a referenced metrics. There is also an Element for the tuple name, but it is not used actually, so it should be always empty. Note the name Element inside kind changes to metric_name in Elements of Tuple Metrics.

```xml
<!ELEMENT tupla_name (#PCDATA)>

<!ELEMENT tupla (tupla_attributes, tupla_elements, comments, sources)>
  <!ELEMENT tupla_attributes
    (metric_name, explanation, tupla_name)>
  <!ELEMENT tupla_elements (tupla_item*)>
  <!ELEMENT tupla_item (name, metric_element)>

Example:

```xml
<metric id="9" kind="tupla">
  <metric_id>Tuple Metrics</metric_id>
  <kind>
    <tupla>
      <tupla_attributes>
        <metric_name> Tuple Metrics</metric_name>
      </tupla_attributes>
    </tupla>
  </kind>
</metric>
```
The function Element follows the structure of simple Metrics kind Elements, and contains a list of input parameters, as well as a list of output parameters (although Functions are restricted to have only one output parameter). Each parameter has a referenced metrics; and all two lists have at least one parameter. There is also an Element for the tuple name and the default value, but they are not actually used, so they should be always empty.

```
<!ELEMENT function_name (PCDATA)>
<!ELEMENT default_value (PCDATA)>

<!ELEMENT function (function_attributes, default_value?,
in_types,
    out_types, comments, sources)>
  <!ELEMENT function_attributes
    (metric_name, explanation, function_name)>  
  <!ELEMENT in_types (metric_element+)>
  <!ELEMENT out_types (metric_element+)>
```

Example:

```
<metric id="10" kind="function">
  <metric_id>Function Metrics</metric_id>
  <kind>
    <function>
```

159
Visualització de models de qualitat mitjançant una interfície web

<function_attributes>
  <name>Function Metrics</name>
  <explanation/>
  <function_name/>
</function_attributes>
<default_value/>
<in_types>
  <metric_element id="2">Boolean Metrics</metric_element>
  <metric_element id="3">Integer Metrics</metric_element>
  <metric_element id="4">String Metrics</metric_element>
</in_types>
$out_types>
  <metric_element id="9">Tuple Metrics</metric_element>
</out_types>
<comments/>
<sources/>
</kind>
</metric>

Formula Metrics are different from other composite Metrics, since they do no reference Metrics but Metrics Assignments to Entities (in fact, they reference Entities). The additional information to store in the formula Element is the text of the formula, and a list of the referenced Entities and their Metrics Assignment in the Function in the entity_metrics Element. Note that this Element's childs are entity and metric. The metric Element contains the Metrics Assignment Id to use in the formula for the Entity with Id equal to the content of the entity Element.

<!ELEMENT formula_definition (#PCDATA)>
<!ELEMENT formula (formula_attributes, comments, sources, entity_metrics)>
<!ELEMENT formula_attributes (metric_name, explanation, formula_definition)>
<!ELEMENT entity_metrics (entity, metric)>
<!ELEMENT entity (#PCDATA)>
<!ELEMENT metric (#PCDATA)>

Example:

<metric id="16" kind="formula">
  <metric_id>Formula</metric_id>
  <kind>
160
3.9. Relation Elements

<!--ELEMENT relation_list (relation*)>
<!--ELEMENT relation (relation_id, relation_context,
relation_attributes,
refinement?)>
<!--ATTLIST relation id ID #REQUIRED>
<!--ELEMENT relation_id (#PCDATA)>
<!--ELEMENT relation_context
(entity_name,entity_name,relation_value)>
<!--ELEMENT entity_name (#PCDATA)>
<!--ATTLIST entity_name id ID #REQUIRED>
<!--ELEMENT relation_value (#PCDATA)>
<!--ATTLIST relation_value id ID #REQUIRED>
<!--ELEMENT relation_attributes (degree_value?)>
<!--ELEMENT degree_value (#PCDATA)>
<!--ELEMENT refinement (relation_refines)>
<!--ELEMENT relation_refines (#PCDATA)>
<!--ATTLIST relation_refines id ID #REQUIRED>

<!--ELEMENT relation_scale_list (relation_scale*)>
<!--ELEMENT relation_scale (relation_scale_id,
relation_scale_attributes,
relation_scale_elements?,
user,sources?)>
<!--ATTLIST relation_scale_id ID #REQUIRED>
<!--ATTLIST relation_scale kind (numeric|textual) #REQUIRED>
<!--ELEMENT relation_scale_id (#PCDATA)>
<!--ELEMENT relation_scale_attributes (name, explanation)>
<!--ELEMENT relation_scale_elements (relation_scale_ref)>
<!--ELEMENT relation_scale_ref (#PCDATA)>
<!--ATTLIST relation_scale_ref id ID #REQUIRED>
Visualització de models de qualitat mitjançant una interfície web

<!ELEMENT relation_kind_list (relation_kind_element*)>
<!ELEMENT relation_kind_element (relation_kind_id, 
relation_kind_attributes)>
  <!ATTLIST relation_kind_element id ID #REQUIRED>
<!ELEMENT relation_kind_id (#PCDATA)>
<!ELEMENT relation_kind_attributes 
(value,relation_scale_name)>
  <!ELEMENT value (#PCDATA)>
<!ELEMENT relation_scale_name (#PCDATA)>"+
  <!ATTLIST relation_scale_name id ID #REQUIRED>