Characterizing a data model for software measurement

L. Chirinos, F. Losavio, and J. Bøegh

Journal of Systems and Software 2005

2006. 7. 19
Yoon, Kyung-A
Contents

- Introduction
- Software measurement elements
- Data model for software measurement
- Considerations for implementing the software measurement data model
- Applying MOSME to a case study
- Comparison with other models
- Conclusion
Many software measurement schemes fail due to poor definition

- Measurement mismatch [PSM ‘02]
  - Picking wrong, ambiguous, or inconsistent measures result in inconclusive data analysis
- Measurements should be objective, empirical and repeatable

Measurement data model

- Identify and define all the elements involved in measurement as well as the relationships existing among them
  - Kitchenham’s software measurement model
  - Bøegh’s data model for software quality measure
- “Everyone understands what the measured values present”
  - Not “the definition of measure is theoretically correct”
In the previous works,

- GQM, Ami
  - Provide methods for identifying the measures
  - Not define how such measures should be collected and stored
- Kitchenham’s work
  - Provide entity-attribute-unit structure (’95)
  - Provide a method for specifying models of software data sets (’01)
    - In order to capture the definitions and possible relationships among software measures
  - Counting rule and measurement context have not been explicitly considered
- In ISO/IEC 15939, the relations defined between elements involved in the measure definition are not normative
Background of MOdel for Software Measurement (MOSME)

- SQUAD(Software QUality Across Different regions)
  - Enrich the existing project database with information on the quality measurements of artifacts produced during the early stages
  - SQUID(Software QUality In the Development process)

- CLeAr and Reliable information For integration (CLARiFi)
  - Create a broker infrastructure to support the application of CBSE in the marketplace
  - Clear software measure concepts was absent with respect to the quality attribute definitions
    - Involving the counting rule to compute the attribute’s values
    - When the finding suitable software components for the planned system was failed
Software measurement elements (1/2)

- Measurement terminology
  - Measure(metric)
    - Rule for assigning a quantitative or categorical value from a defined scale to one or more attributes
    - Refer collectively to base measures, derived measures and indicators
  - Indicator
  - Measurement
    - Use of a measure or mapping to assign a value from the scale to an attribute of an entity
    - Mapping from the empirical world to the formal, relational world
Formalization of measurement elements

- Homomorphism
  \( \mu : (E, R) \rightarrow \mu(R, >) \), where
  \( \forall e \in E, \mu(a(e_1)) \in R \)
  - We require
    \[ a(e_1)R a(e_2) \iff \mu(a(e_1)) \succ \mu(a(e_2)) \]

\[ \{ (e_1, e_2) \in E \times E \mid a_1R a_2 \} \]
where \( R \) is a relation between objects \( a_1 \) and \( a_2 \) in the real world represented by \( e_1 \) and \( e_2 \) which are entities in the algebraic system.
Data model for software measurement (1/2)
Data model for software measurement (2/2)

- Counting rule
  - Procedure
    - Specify the set of actions to be performed to obtain the value of the attribute on the basis the selected unit
    - Expression method:
      - Descriptive text or formula
  - Context (context of use)
    - Circumstances or context under which the measurement is performed
      - Frequency of the measurement
      - Tools to be used to extract and store the data values
      - Responsible for the data extraction
      - Environmental elements
  - Conditions
    - Conditions on the target values for a specific attribute
Considerations for implementing the software measurement data model (1/3)

Identify and define the elements involved in the measure and their relationships.

Identify the measures that are applicable in a specific project domain providing a context for measurement.
Considerations for implementing the software measurement data model (2/3)

- Main activities for the measure definition
  - 1. Define the algebraic system
     - 1.1 Identify the entity type that shall be analyzed
     - 1.2 Identify and define the attributes involved in the entity type
     - 1.3 Identify the algebraic relations and their empirical interpretations for the entity type with respect to the attributes
  - 2. Construct the underlying numerical system to which the algebraic system will be mapped
  - 3. Define the mapping between the algebraic system and the numerical system (measure):
    - 3.1 Define the unit and scale type
    - 3.2 Specify the counting rule: procedure, context of use (if applied to all the entities) and condition (if applicable)
Considerations for implementing the software measurement data model (3/3)

- Main activities for defining the project measures
  - 1. Identify the project
  - 2. Identify the entity type to be controlled during the development process
  - 3. Assign target values to the attributes
  - 4. Define the entities to be measured in the project and link them to the corresponding entity type according to the development process adopted
  - 5. Define the context of use where the measure will be applied
Applying MOSME to a case study (1/3)

 Problem description

- COMERX
  - Building an Enterprise Application Portal (EAP) to articulate its main business activities (B2C, SCM, CRM)
  - Its main goals are to provide efficient answers to customer’s requests and to guarantee continuous availability of the COMERX functionality

- MOSME is applied
  - To define the measures for the attributes identified for the ISO/IEC 9126-1 quality model adapted to the EAP domain
  - As part of a quality requirements specification process
Applying MOSME to a case study (2/3)

- **Main quality characteristics, sub-characteristics and attributes**

<table>
<thead>
<tr>
<th>External quality characteristics (ISO/IEC 9126-1, 2001)</th>
<th>External quality sub-characteristics</th>
<th>External quality sub-sub-characteristics</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Efficiency</strong>: the capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions</td>
<td><strong>Time behavior</strong>: The capability of COMERX to provide appropriate response and processing times and throughput when performing its function, under stated conditions</td>
<td></td>
<td>• <strong>Response time</strong>: time taken by COMERX to answer to a user’s request after being processed</td>
</tr>
<tr>
<td><strong>Reliability</strong>: the capability of the software product to maintain a specified level of performance when used under specified conditions</td>
<td><strong>Availability</strong>: The capability of COMERX to be in a state to perform a required function at a given point in time, under stated conditions of use</td>
<td><strong>Fault tolerance</strong>: the capability of COMERX to maintain a specified level of performance in cases of software faults or of infringement of its specified interface</td>
<td>• <strong>Throughput (communication capacity)</strong>: amount of information transmitted through the portal over a given period of time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Size 1</strong>: number of interruptions that leave the system out of operation during a specified period of time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Size 2</strong>: number of functions implemented with the capacity of avoiding incorrect operations</td>
</tr>
<tr>
<td>Entity type</td>
<td>Attribute</td>
<td>Algebraic relation</td>
<td>Measure name</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Name: Service</td>
<td>Name: Response time</td>
<td>Name: “is more efficient (with respect to response time) than” denoted by &quot;&gt;&quot;</td>
<td>M1RT</td>
</tr>
<tr>
<td>Description: time taken by the application to answer to a service request</td>
<td>Empirical interpretation: the lower response time taken by a service invoked through the portal gives an idea of a greater efficiency of the different components involved in producing the service. It is an indicator that affects the acceptance of the portal in the user’s context</td>
<td>Descriptive text: counting the elapsed time from the application acknowledgement of the service request until the response is obtained</td>
<td></td>
</tr>
<tr>
<td>Context:</td>
<td></td>
<td>• Stage: testing, after completing the service coding</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Frequency: at the first compilation and after each fault correction</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Who: measurement engineer</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Operating system: UNIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Internet data transmission rate: 25 KBps (corresponds to the platform used by enterprise X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Channel bandwidth:</td>
<td></td>
</tr>
</tbody>
</table>

**Measures used in the COMERX Project**

<table>
<thead>
<tr>
<th>Entity</th>
<th>Project</th>
<th>Measure</th>
<th>Target value</th>
<th>Actual value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order management</td>
<td>COMERX</td>
<td>M1RT</td>
<td>5 s</td>
<td>To be determined by simulation or product execution</td>
</tr>
<tr>
<td>Order management</td>
<td>COMERX</td>
<td>M1TT</td>
<td>125 KBps</td>
<td>To be determined by simulation or product execution</td>
</tr>
<tr>
<td>Service request management</td>
<td>COMERX</td>
<td>M1RT</td>
<td>5 s</td>
<td>To be determined by simulation or product execution</td>
</tr>
<tr>
<td>Service request management</td>
<td>COMERX</td>
<td>M1TT</td>
<td>125 KBps</td>
<td>To be determined by simulation or product execution</td>
</tr>
</tbody>
</table>
Comparison with other models (1/3)

- Kitchenham’s software measurement model

![Conceptual E-R model]

< Conceptual E-R model >
Comparison with other models (2/3)

- Measurement information model of ISO/IEC 15939
### Comparison with other models (3/3)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Measurement definition</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity type</td>
<td></td>
<td>○</td>
<td>○</td>
<td>X</td>
<td>○</td>
</tr>
<tr>
<td>Entity</td>
<td>Object of evaluation</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Attribute</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Unit</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td><strong>Structure of counting rule</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Textual description or equation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Characterization of an entity with respect to the attribute</td>
<td>Not explicitly defined</td>
<td>Not explicitly defined</td>
<td>Not explicitly defined</td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Context of use</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>○</td>
</tr>
<tr>
<td>Conditions</td>
<td></td>
<td>○</td>
<td>X</td>
<td>X</td>
<td>○</td>
</tr>
<tr>
<td>Scale</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Scale type</td>
<td></td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Measure type</td>
<td>Direct/Indirect</td>
<td>Direct/Indirect</td>
<td>Base/Derived/Indicator</td>
<td>Direct/Indirect</td>
<td></td>
</tr>
<tr>
<td>Correspondence between empirical and formal worlds</td>
<td>Function</td>
<td>Function</td>
<td>Function</td>
<td>Homomorphism between algebraic and numerical systems</td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>○</td>
</tr>
<tr>
<td>Model of the development process</td>
<td></td>
<td></td>
<td></td>
<td>Entities represent the artifacts</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- Proposed data model focuses on
  - Definition and modeling of the elements involved in software measurement
    - Particularly the counting rule and the role played by the context of use

- Weak points
  - Exact and theoretically based definition of these critical elements for software data storage, collection and comparison had not been provided
  - Conditional model is not included in the data model in order to simplify this presentation
Critics

- MOSME does not include
  - Nominal, ordinal scale and the special interpretation of their unit
  - Indirect measurement
    - Authors said indirect measure can be described in the procedure part of counting rule in the form of a formula
    - However, MOSME can’t present the relationship between base measures
  - Goal
    - Authors said goal can be described in the context of counting rule
    - However, I think that goal should be the individual element from context
  - Process-oriented measure
    - It only focuses on the product-oriented measure
    - Then, are there any specific elements of process-oriented measure to be added to MOSME?

- Counting rule can be ambiguous