Model-based Simplified Functional Size Measurement
An Experimental Evaluation with COSMIC Function Points

Vieri del Bianco, Luigi Lavazza, Geng Liu, Sandro Morasca

Università degli Studi dell'Insubria, Dipartimento di Scienze
Teoriche e Applicate
Varese, Italy

Abedallah Zaid Abualkishik

IT department, Al-Khawarizmi International College
Abu Dhabi, United Arab Emirates
Motivations

- Functional Software Measurement is supposed to quantify functional user requirements
  - IFPUG
  - COSMIC: a bit simpler than IFPUG

Need for simplification

- reduce counting effort
- reduce counting time

Most of simplification works are about IFPUG

- COSMIC can/should be simplified too
- we focus on COSMIC CFP
User requirements are in various notations
  • including UML
  • IFPUG Function Points and COSMIC Function Points can be applied to UML requirements

It is important to write UML models with Functional Software Measurement in mind
  • so that models represent all of the information required by Functional Software Measurement methods
UML models

- are collections of diagrams
- become more and more complete and detailed over time
- include an increasing number of diagrams
UML models convey an increasing amount of information, which can be used for

- *executing* Functional Software Measurement methods
- *simplifying* Functional Software Measurement, when we have a subset of the information needed to carry out the complete official measurement processes
During the requirements elicitation and specification phase, is it possible to write progressively more complete and detailed UML models that support progressively more accurate simplified CFP measurement methods?

What is the accuracy of different simplified CFP measurement methods, i.e., how close are the estimated sizes they provide to the actual ones?

Do simplified CFP measurement methods provide an accuracy level that increases with the amount of information required?
Identify the functional processes of the application being measured

Identify the data groups mentioned in the user requirements

Count the unique data movements involving each identified data group for each functional process
  - Entries and Exits (i.e., I/O movements)
  - Reads and Writes (to persistent storage)

A data group is persistent if its value is stable between two consecutive functional process executions
Measurement Method: Timeline

Sandro Morasca  
EESSMod, Miami, FL, USA  
October 1, 2013
Proposed by COSMIC Consortium

Perform only the first of the activities required for CFP measurement
  • identification of functional processes

Use historical data to compute the mean number of data movements per functional process \( M_{DM} \)

Compute \( CFP = M_{DM} \cdot \#FPr \)
  • \#FPr: \#Functional processes in the application
Simplified Measurement Method I: Timeline

Motivations
- Simplified Measurement Methods
- UML Diagrams
- Empirical Study
- Research Answers
- Threats to Validity
- Future Work

Identification of Functional Processes
Identification of Data Groups
Identification of Data Movements

#FPr is known here
CFP is known here

Sandro Morasca
EESSMod, Miami, FL, USA
October 1, 2013
Size in CFP will likely increase with the number of data groups (#DG) *in the application*

Compute CFP = f(#FPr, #DG)

- with f(#FPr, #DG) derived via regression analysis

Simpler than “full” COSMIC process

- data movements are not identified
- a conceptual model of application data involved is usually built very early in the requirements modeling process
Simplified Measurement Method II: Timeline

Motivations
- Simplified Measurement Methods
- UML Diagrams
- Empirical Study
- Research Answers
- Threats to Validity
- Future Work

Identification of Functional Processes
Identification of Data Groups
Identification of Data Movements

#FPr is known here
#DG is known here
CFP is known here

Sandro Morasca
EESSMod, Miami, FL, USA
October 1, 2013
Simplified Measurement Method III: #DG in Each Functional Process

- A more accurate estimate may be derived if information is available on each functional process individually
  - like the number of data groups involved in each functional process

Compute CFP = f(#FPr, AvDGperFPr)
  - AvDGperFPr: average number of data groups involved in each of the functional processes in the application
Simplified Measurement Method III: Timeline

- Identification of Functional Processes
- Identification of Data Groups
- Identif. of DG used in each FPr
- Identif. of each FPr Data Mov.

Motivations
- Simplified Measurement Method
- UML Diagrams
- Empirical Study
- Research Answers
- Threats vs. Future Work

#FPr is known here
#DG is known here
number of DG involved in each FPr is known here
CFP is known here

Sandro Morasca
EESSMod, Miami, FL, USA
October 1, 2013
Simplified approaches to CFP measurement can be used if certain UML models are built.

The UML models used for measurement are models of the functional user specifications:

- no design elements
- only user-relevant information
Compute CFP = $M_{DM} \cdot \#FPr$

- $\#FPr$: count of operations listed in User_interface
- $M_{DM}$: based on historical data
Compute $\text{CFP} = f(#FPr, #DG)$

- $#FPr$: count of operations listed in User_interface
- $#DG$: count of data groups in “System”
Simplified Measurement Method III
Component Diagram

Compute CFP = f(#FPr, AvDGperFPr)

Sandro Morasca
EESSMod, Miami, FL, USA
October 1, 2013
UML ports indicate
- which classes (i.e., data groups) are used
- in each functional process

Sets of functional processes that use the same set of classes are grouped into a single interface

Only the interfaces needed to add, change, and delete clients are shown
Grouping functional processes according to the used classes may lead to many interfaces
  • lower readability of the diagram

However, interfaces that are homogeneous with respect to the used classes
  • allow for a quite precise estimation of size
  • explicitly represent the logical relationship between interface elements and system data
    • identification of important traceability information when the design model is built
Alternatively

The sequence diagram represents a specific functional process and the involved class instances

Compute $CFP = f(#FPr, AvDGperFPr)$
Refinement

Model-based Simplified Functional Size Measurement

Sandro Morasca
EESSMod, Miami, FL, USA
October 1, 2013
Refinement

Sandro Morasca
EESSMod, Miami, FL, USA
October 1, 2013
Computation of COSMIC CFP based on the manual
Steps

- Modeling of a set of software applications
- Measurement of their CFP
- Application of the simplified measurement methods, obtaining size estimates
- Comparison of the size estimates with the measures obtained via the standard COSMIC method
We used 11 projects
  • sample projects provided by COSMIC to illustrate the counting process (5 projects)
  • academic examples used in teaching (5 projects)
  • project management tools (one project)

The UML models were built by a PhD student following a methodology proposed by two of the authors

Model quality was checked by two authors
### The Dataset

<table>
<thead>
<tr>
<th>Pid</th>
<th>CFP</th>
<th>#FPr</th>
<th>#DG</th>
<th>Avg #DG per FPr</th>
<th>Avg DM per DG per FPr</th>
<th>Avg FPr size (DM per FPr) others</th>
<th>Avg #DG others</th>
<th>Avg CFP/#DG others</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>86</td>
<td>16</td>
<td>6</td>
<td>2.88</td>
<td>1.90</td>
<td>5.12</td>
<td>3.01</td>
<td>8.7</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>11</td>
<td>11</td>
<td>3.55</td>
<td>1.60</td>
<td>5.14</td>
<td>2.98</td>
<td>9.6</td>
</tr>
<tr>
<td>3</td>
<td>91</td>
<td>15</td>
<td>10</td>
<td>4.00</td>
<td>1.57</td>
<td>5.08</td>
<td>2.93</td>
<td>9.1</td>
</tr>
<tr>
<td>4</td>
<td>69</td>
<td>19</td>
<td>12</td>
<td>2.32</td>
<td>1.72</td>
<td>5.26</td>
<td>3.06</td>
<td>9.6</td>
</tr>
<tr>
<td>5</td>
<td>103</td>
<td>19</td>
<td>16</td>
<td>3.06</td>
<td>1.93</td>
<td>5.09</td>
<td>3.00</td>
<td>9.5</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>14</td>
<td>7</td>
<td>2.64</td>
<td>1.71</td>
<td>5.17</td>
<td>3.03</td>
<td>9.2</td>
</tr>
<tr>
<td>7</td>
<td>116</td>
<td>20</td>
<td>14</td>
<td>3.60</td>
<td>1.65</td>
<td>5.07</td>
<td>2.95</td>
<td>9.3</td>
</tr>
<tr>
<td>8</td>
<td>124</td>
<td>20</td>
<td>10</td>
<td>2.80</td>
<td>2.38</td>
<td>5.04</td>
<td>3.02</td>
<td>8.9</td>
</tr>
<tr>
<td>9</td>
<td>117</td>
<td>19</td>
<td>9</td>
<td>3.47</td>
<td>1.78</td>
<td>5.05</td>
<td>2.97</td>
<td>8.8</td>
</tr>
<tr>
<td>10</td>
<td>90</td>
<td>13</td>
<td>14</td>
<td>3.92</td>
<td>1.99</td>
<td>5.04</td>
<td>2.95</td>
<td>9.5</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>3</td>
<td>10</td>
<td>2.07</td>
<td>2.40</td>
<td>5.43</td>
<td>3.30</td>
<td>9.1</td>
</tr>
</tbody>
</table>

Sandro Morasca  
EESSMod, Miami, FL, USA  
October 1, 2013
The $M_{DM}$ of the other projects was used to compute the size of an application with

$$CFP = M_{DM} \cdot \#FPr$$

Accuracy indicators:
- $MMRE = 17.8\%$
- $Pred(25) = 72.7\%$
- percentage error range = $[-27.8\%, 44.9\%]$
The mean number of data movements per data group of the functional processes, computed for each application, was fairly constant in the dataset
  • mean = 1.88 and standard deviation = 0.29

We defined an additional model

$$\text{CFP} = \text{NumFPr} \cdot \text{AvDGperFPr} \cdot \text{AvDMperDGperFPr}$$

Accuracy indicators
  • MMRE = 15.3%
  • Pred(25) = 81.8%
  • percentage error range = [-15.3%, 33.9%]
## Results

### Estimates via Simplified Measurement Method I

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>86</td>
<td>82</td>
<td>-4</td>
<td>-4.7%</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>57</td>
<td>1</td>
<td>1.8%</td>
</tr>
<tr>
<td>3</td>
<td>101</td>
<td>86</td>
<td>-15</td>
<td>-14.9%</td>
</tr>
<tr>
<td>4</td>
<td>69</td>
<td>100</td>
<td>31</td>
<td>44.9%</td>
</tr>
<tr>
<td>5</td>
<td>103</td>
<td>92</td>
<td>-11</td>
<td>-10.7%</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>72</td>
<td>8</td>
<td>12.5%</td>
</tr>
<tr>
<td>7</td>
<td>116</td>
<td>101</td>
<td>-15</td>
<td>-12.9%</td>
</tr>
<tr>
<td>8</td>
<td>124</td>
<td>101</td>
<td>-23</td>
<td>-18.5%</td>
</tr>
<tr>
<td>9</td>
<td>117</td>
<td>96</td>
<td>-21</td>
<td>-17.9%</td>
</tr>
<tr>
<td>10</td>
<td>90</td>
<td>65</td>
<td>-25</td>
<td>-27.8%</td>
</tr>
<tr>
<td>11</td>
<td>252</td>
<td>326</td>
<td>74</td>
<td>29.4%</td>
</tr>
</tbody>
</table>

### Estimates via Simplified Measurement Method IV

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88</td>
<td>2</td>
<td>2.3%</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
<td>19</td>
<td>33.9%</td>
</tr>
<tr>
<td>3</td>
<td>132</td>
<td>31</td>
<td>30.7%</td>
</tr>
<tr>
<td>4</td>
<td>85</td>
<td>16</td>
<td>23.2%</td>
</tr>
<tr>
<td>5</td>
<td>105</td>
<td>2</td>
<td>1.9%</td>
</tr>
<tr>
<td>6</td>
<td>71</td>
<td>7</td>
<td>10.9%</td>
</tr>
<tr>
<td>7</td>
<td>139</td>
<td>23</td>
<td>19.8%</td>
</tr>
<tr>
<td>8</td>
<td>105</td>
<td>-19</td>
<td>-15.3%</td>
</tr>
<tr>
<td>9</td>
<td>127</td>
<td>10</td>
<td>8.5%</td>
</tr>
<tr>
<td>10</td>
<td>97</td>
<td>7</td>
<td>7.8%</td>
</tr>
<tr>
<td>11</td>
<td>218</td>
<td>-34</td>
<td>-13.5%</td>
</tr>
</tbody>
</table>
During the requirements elicitation and specification phase, is it possible to write progressively more complete and detailed UML models that support progressively more accurate simplified CFP measurement methods?

YES
What is the accuracy of different simplified CFP measurement methods, i.e., how close are the estimated sizes they provide to the actual ones?

Look at MMRE, Pred(25), and %Err.
Do simplified CFP measurement methods provide an accuracy level that increases with the amount of information required?

YES
• Limited number of projects in our sample

• Just one modeler

• “Convenience” sample
  • typical threat for most empirical software engineering studies
  • we did not make any specific selection
  • note that we are interested in the performance of the techniques, not on identifying specific models

• Accuracy indicators
  • use of MMRE and the error bounds
Threats to Validity

Only one of the projects used within this empirical study is a real implemented project, however

- the requirements of our projects were realistic: they could all be implemented and size measurement and effort estimation could be carried out
- we compare measures obtained via simplified and full-fledged processes
  - some characteristic requirements affecting the standard size measure will affect in the same way the simplified measure, so that the results are hardly affected
Future activities

- measure more projects
- build better estimation models
- build better (modeling) models