CompSize: Automated Size Estimation of Embedded Software Components

Kenneth Lind
Saab Automobile AB/Chalmers University of Technology

Rogardt Heldal, Tony Heimdahl
Chalmers University of Technology

Tigran Harutyunyan
University of Gothenburg
Motivation


Background

- Flexibility for future requirements obtained through spare resources like processor capacity and memory space.
Background

Too much spare resources =>
Unnecessary piece cost

Too little spare resources =>
Premature redesign

Need for decision support!
Background

• Memory size in ECU estimated based on requirement specifications and historical data.

• COSMIC Function Points measured from Component Diagrams and textual requirements.
Background

- Categorization based on factors like (in our domain)
  - Functionality type
  - Quality constraints
  - Methods & tools

- Memory size in ECU estimated within 15% accuracy before application software is available.
This presentation is based on two publications:


Problem

Manual estimation approach

Example

Example

Requires 2.5 man years in effort for the complete application software in a car!
Problem

• RQ1: “How can UML support in modeling all information needed for automated estimation of Software Code Size?”

• RQ2: “How much manual effort can be saved by modeling all information needed for automated estimation of Software Code Size?”
Solution

Proposed estimation approach

- Domain expert
  - Requirement definition
    - Textual specification
- Architect
  - Decomposition into enhanced components
  - Enhanced Component Diagram
- Measurement engineer
  - Model-based & automated measurement & estimation
  - Introduced in this work!

CompSize

Bytes
Evaluation

RQ1: “How can UML support in modeling all information needed for automated estimation of Software Code Size?”

Categorization factors

<<Interface>>
IInteriorLightStatus_rsp

<<Signal>>
InteriorLightStatus_rsp

<<use>>

<<Category>>
LGT_ControlTruckBedCargoLamp

<<Category>>
functionality = Comf & Conv
constraints = no
decomposition_level = Distributable
granularity_level = Textual spec.
...

<<Parameter>>
Direction = in

<<Parameter>> CARGO LAMP PRESENT:Boolean

Variables

<<Category>>

<<Interface>>
IPickupBedCargoLamp osg

<<Signal>>
PickupBedCargoLamp osg
Evaluation

<<Interface>>
InteriorLightStatus_rsp
<<Signal>>
InteriorLightStatus_rsp
<<Category>>
functionality = Comf & Conv

<<Category>>
LGT_ControlTruckBedCargoLamp

Variables

<<Parameter>> Direction = in
<<Interface>>
IPickupBedCargoLamp_osg
<<Signal>>
PickupBedCargoLamp_osg

<<Category>>
constraints = nodecomposition
granularity_level = Distributable
Textual spec.

<<Parameter>> CARGO LAMP PRESENT:Boolean
### Evaluation

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Entry</th>
<th>Exit</th>
<th>Read</th>
<th>Write</th>
<th>DFP</th>
<th>CFP</th>
<th>Est.</th>
<th>Est.</th>
<th>Est.</th>
<th>Est.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGT_ControlTruckBedCargoLamp</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Component name**: LGT_ControlTruckBedCargoLamp
- **COSMIC data**: movement types
- **CFP**: 6
- **Categorization factors**:
  - Functionality: Comf & Conv
  - Constraints: No
  - Decomposition_level: Distributable
  - Granularity_level: Textual spec
RQ2: “How much manual effort can be saved by modeling all information needed for automated estimation of Software Code Size?”

Case study conducted by Master students
- Our manual estimations was replicated using the UML Profile
- 0.5 man years (instead of 2.5!) to estimate the size of the complete application software in a car
Results

• We have defined a UML Profile capturing the information needed for memory size estimation.
• We have developed a tool that can import the information captured in the UML Profile.
• Using the UML Profile requires 0.5 man years (instead of 2.5 man years!) to estimate the size of the complete application software in a car.
Backup
Textual Specification

The feature **Shall** be enabled when the calibration CARGO LAMP PRESENT is set true. <END>

If the vehicle power mode is “OFF”, and the cargo lights are illuminated, the SYSTEM **Shall** keep the cargo lamps active as long as Inadvertent Load Control power is active. <END>

<table>
<thead>
<tr>
<th>CUSTOMER “ACTION”</th>
<th>CUSTOMER PERCEIVABLE “OUTPUT”</th>
<th>MAXIMUM LATENCY “ACTION” to “OUTPUT”</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERIOR ILLUMINATION Lamps Switch On and Vehicle Parked.</td>
<td>Cargo Lamp Illuminates</td>
<td>100 ms</td>
</tr>
</tbody>
</table>

Back to Manual Estimation approach
Component Diagram
The feature **Shall** be enabled when the calibration CARGO LAMP PRESENT is set true. <END>

If the vehicle power mode is “OFF”, and the cargo lights are illuminated, the SYSTEM **Shall** keep the cargo lamps active as long as Inadvertent Load Control power is active. <END>
UML Profile

Already in use at Saab

Categorization factors

Back to Proposed Estimation approach
### Components

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Entry</th>
<th>Exit</th>
<th>Read</th>
<th>Write</th>
<th>DFP</th>
<th>CFP</th>
<th>Est. CodeSize</th>
<th>Real Code Size</th>
<th>Est. Datasize</th>
<th>Real Datasize</th>
<th>Est. Effort</th>
<th>Real Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reversing_Lamp_Outage</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>894</td>
<td>812</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Parklamp_Outage</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>10</td>
<td>1388</td>
<td>1388</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lowbeam_Outage</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>15</td>
<td>1594</td>
<td>1594</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Remote_PRNDL_Display</td>
<td>5</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>1594</td>
<td>1594</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EPN_Plant_Assembly_Mode</td>
<td>4</td>
<td>2</td>
<td>11</td>
<td>0</td>
<td>6</td>
<td>17</td>
<td>1226</td>
<td>1226</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Other_ID</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>8</td>
<td>1030</td>
<td>1030</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Categorization factors

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Factors</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reversing_Lamp_Outage</td>
<td>Team</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Functionality</td>
<td>Comf &amp; Conv</td>
</tr>
<tr>
<td></td>
<td>Constraints</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>MethodAndTool</td>
<td>Rhapsody</td>
</tr>
<tr>
<td></td>
<td>Compiler</td>
<td>GHforC</td>
</tr>
<tr>
<td></td>
<td>HardwareDiagnostics</td>
<td>No</td>
</tr>
</tbody>
</table>
CompSize

Back to Proposed Estimation approach

Estimated Code Size