Key maturity indicators for module libraries for PLC-based control software in the domain of automated Production Systems

Birgit Vogel-Heuser, Juliane Fischer, Eva-Maria Neumann*, Sebastian Diehm
*e@maria.neumann@tum.de

Technical University of Munich
Institute for automation and information systems
Boltzmannstr. 15
85748 Garching near Munich, Germany
info@ais.mw.tum.de
Agenda

Motivation

State of the art

Introduction of code-based quality metrics for PLC control software modules

Application Example

Evaluation and Outlook
Motivation for maturity metrics for control software module libraries

Workflow of library module development

1.) High maturity: Green traffic light
2.) Medium maturity: Yellow traffic light
3.) Low maturity: Red traffic light
State of the art in mature PLC-based control software module evolution

- **Classification of changes / evolution in PLC code**

  C1: \[ f(ln) \]
  C2: \[ f(ln) \]
  C3: \[ g(ln) \]
  C4: \[ f(ln) \]
  C5: \[ f(ln) \]
  C6: \[ f(ln) \]
  C7: \[ f(ln) \]
  C8: \[ f(ln) \]

  \[ IM_{C1}, IM_{C2} > IM_{C3} \text{ and } IM_{C4} > IM_{C5} \text{ and } IM_{C7} > IM_{C6} \]

  Extended classification of Vogel-Heuser et al. (2014) of possible changes of a FB in IEC 61131-3 and comparison of the respective impacts IM on module maturity rated by industrial application experts

- **Code analysis for PLC code**

  - Questionnaire and code analysis of industrial legacy control code (Vogel-Heuser et al. (2017)) confirmed applicability of **Halstead program length** and **McCabe’s complexity** to PLC languages (Capitán et al. (2017))

- **Code-based software quality metrics in general software engineering**

  - Hristov et al. (2012): structuring of reuse metrics into availability, documentation, complexity, quality, maintainability and adaptability
  - Deniz et al. (2014): code-based quality metrics in industrial settings (quality of software products is improved as reuse rates of the product increases)
Code-based quality metrics for PLC control software modules

\[ \Delta F_B^{v+1} < \Delta F_B^{v} < \Delta F_B^{v-1} \]  

- Changes decrease with a rising version number indicating a higher maturity level of the FB

\[ \Delta F_{B, \text{Operation}} < \Delta F_{B, \text{Start-up}} < \Delta F_{B, \text{Design}} \]  

- Changes to one specific FB decrease as it advances in the software engineering life cycle

\[ \Delta F_{B, \text{level}} > \Delta F_{B, \text{level-1}} > \Delta F_{B, \text{level-2}} \]  

- Maturity of a FB\(_{\text{level-2}}\) from a lower level of the software architecture is higher than the one of a higher level FB\(_{\text{level-1}}\)
Code-based quality metrics for PLC control software modules

Maturity metrics for evolving FBs in a module library

\[ LIB \_ M_{\text{Mat}} \_ FB_x = 1 - \frac{1}{n} \cdot \sum_{y=1}^{n} \Delta FB_{x,y} = 1 - \frac{1}{n} \cdot \left( \sum \text{variables} \right) \]

\[ + \sum \text{interface changes} \left\{ \sum \text{interface elements} \right\} \]

\[ + \sum \text{adaption of module implementation} \left( \sum \text{module implementations} \right) \]

\[ + \sum \text{adding/deleting of sub-modules} \left( \sum \text{sub-modules} \right) \]

\[ + \sum \text{adding/deleting of preprocessing of inputs external} \left( \sum \text{preprocessing of inputs external} \right) \]

\[ + \sum \text{changed designations} \left( \sum \text{variables} \right) \]

With:

\( n = \text{number of } \Delta FB \land \Delta FB \neq 0 \)

\( n_A \) number of changed analogue inputs \( I_A \)

\( k_{AI} \) Weighting factor for analogue inputs

\( n_D \) number of changed digital inputs \( I_D \)

\( n_C \) number of changed counter inputs \( I_C \)

\( m_A, m_D, m_C \) respective changes of the outputs \( O \)

\( k \) Weighting factors

\[ (4) \]

\[ LIB \_ M_{\text{Mat}} \_ FB_x = \begin{cases} 
1 & \text{if nothing is changed} \\
0 & \text{if everything is changed} \\
> 0 \land < 1 & \text{in case of changes}
\end{cases} \]

\[ (5) \]
Application of the proposed metrics

1) Original specific FB

2) General FB
Application of the proposed metrics

Specific FB → General FB

- The original FB comprises 14 interfaces whereas the adapted FB has 35 interface elements:

\[ FB_{1,x,1} = \frac{\Delta \text{interface changes}}{\sum \text{interface elements}} = \frac{14 - 35}{35} = 0.60 \]

- The original FB implementation comprises 47 LOC and 87 LOC were adapted, which means that they were either added, removed or edited:

\[ FB_{1,x,2} = \frac{\Delta \text{Module implementations}}{\sum \text{Module implementations}} = \frac{47 - 87}{87} \approx 0.46 \]
Application of the proposed metrics

Specific FB → General FB

- The original FB comprises 14 interfaces whereas the adapted FB has 35 interface elements:

\[
FB_{1,x,1} = \left| \frac{\sum \text{interface changes}}{\text{interface elements}} \right| = \left| \frac{14 - 35}{35} \right| = 0.60
\]

- The original FB implementation comprises 47 LOC and 87 LOC were adapted, which means that they were either added, removed or edited:

\[
FB_{1,x,2} = \left| \frac{\sum \text{Module implementations}}{\text{Module implementations}} \right| = \left| \frac{47 - 87}{87} \right| \approx 0.46
\]

\[
LIB_{-}M_{-}Mat \cdot FB_{1,x} = 1 - \frac{1}{2} \cdot \sum_{y=1}^{2} \Delta FB_{1,x,y} = 1 - \frac{1}{2} \cdot (0.60 + 0.46) = 0.47
\]
Previous feedback and outlook

Qualitative evaluation by industrial experts

- All experts agreed on the exponential behavior:

- The terms of (4) may need refinement, but the rationale was accepted by all of them.

\[
LIB_{\text{M}_n} = \frac{1}{n} \sum_{i=1}^{n} \Delta F_{B_i} = 1 - \frac{1}{n} \cdot \sum_{i=1}^{n} \Delta F_{B_i}
\]

\[
(4)
\]

Outlook:
- Adaption of the metrics to other IEC 61131-3 programming languages
- Introduction of weighting factors
- Evaluation of the metrics using industrial PLC code
Thank you very much for your attention!

Birgit Vogel-Heuser, Juliane Fischer, Eva-Maria Neumann*, Sebastian Diehm
* eva-maria.neumann@tum.de

Technical University of Munich
Institute for automation and information systems
Boltzmannstr. 15
85748 Garching near Munich, Germany
info@ais.mw.tum.de