Manage the automotive embedded software development cost & productivity
with the automation of a Functional Size Measurement Method (COSMIC)

Author’s name: Alexandre Oriou, Eric Bronca, Boubker Bouzid, Olivier Guetta, Kevin Guillard.
RENAULT S.A.S., 1 avenue du Golf, 78 288 Guyancourt cedex, France

ABSTRACT
Year after year, more and more cars' functionalities are performed by software: electrical vehicle, multimedia, connectivity with the outside world and so on. As its software development costs are increasing, Renault decided to develop metrics and an estimation process in order to be able to predict its software costs early in the vehicle or power-train project. At the same time, Renault is working with its major Electronic Control Units suppliers to contract with them on the basis of software metrics. After different studies, Renault chose the COSMIC method as its embedded software metric. COSMIC is for COnmon Software Measurement International Consortium, and is also the name of a Functional Size Measurement (FSM) method, ISO standard since 2003. To make its measurement process robust, Renault has decided to automate the functional size measurement.

Keywords: Software metrics, Software functional size, Software development workload estimation process, Productivity models, COSMIC.

1. INTRODUCTION
The use of Electronic Control Unit (ECU) software in cars has grown considerably in recent years. In the car development process, the ECU's software development and validation tasks are clearly identified upfront, and the corresponding milestones in global car development planning are strongly positioned. To manage more tightly the development costs of software suppliers, as well as its own validation activities, Renault SAS measures the functional size of their ECU software specifications using international standards, and uses this information as the main input for estimating software development costs and schedules.

Software measurement can be used to obtain software development productivity ratios and to build objective estimation models for predicting project effort and duration. In fact, software size, measured as function points, is highly correlated with project work effort.

Renault SAS has chosen the COSMIC–ISO 19761 standard for measuring the size of real-time embedded software and for estimating project costs. This function point-based measurement method is
applicable at both the beginning of the development process, in the requirements elicitation or specification phase, and at the end of the project, after implementation for benchmarking studies.

This paper presents the design of the FSM procedure based on the COSMIC method to obtain the functional size of software specifications expressed with the Simulink tool. Each element of the Simulink model had to be translated into COSMIC concept using the rigorous rules that avoid all ambiguities and help reduce the discrepancy between different measurers of the same software application. This paper presents also the automated measurement tool developed at Renault SAS implementing this approach, and applied on real power-train ECU software models.

2. COSMIC METHOD APPLY TO SIMULINK

2.1 General points

The COSMIC method is an ISO standard (19761, 2003). The whole COSMIC documentation is free and available on www.cosmicon.com.

The COSMIC method is based on data movements’ measurement. The COSMIC measurement process consists of three phases: The Measurement Strategy, the Mapping Phase and the Measurement Phase. The result of applying the measurement process to a piece of software is a functional size measure of the Functional User Requirements of the piece of software expressed in ‘COSMIC Function Points’ (or ‘CFP’).

During the Measurement Strategy phase, the measurer has to define the Goals and the Software Context Model applied to the software to be measured. The Mapping Phase allows to map the FUR (Functional User Requirement) to be measured onto the Generic Software Model. These two phases are followed by the Measurement Phase in which the size measurement results are obtained.

2.2 The measurement strategy phase

The purpose of this procedure is to apply the COSMIC method to the Simulink model, i.e. to measure the size of the FUR of any system based on its functional specifications described using the Simulink model. The size of the FUR can be used to estimate for example the effort required to develop the software.
(designed using Simulink). The scope of the measurement procedure is at the detail level of the subsystem of the Simulink Model and the level of granularity is at the block level of the Simulink Model.

2.3 The mapping phase

This second phase includes the identification of functional processes, data groups and data attributes. In our procedure, a subsystem containing an elementary block is a functional process: when triggered, it receives, manipulates, and moves data groups. StateFlow blocks are also functional processes. There is a boundary lying between any external software (functional user) and the software to be measured (application). There is also a boundary between any two subsystems (peer components in the same layer).

2.3 The measurement phase

2.3.1 Standard case :

In this section, the rules for identifying the data movements of a module designed using the Simulink model are introduced, and the rules that are used to measure the functional size of a new module are defined. The data movements of each functional process are identified and the standard value of 1 CFP is assigned to each data movement. The final step consists of aggregating the results to obtain the functional size of each functional process. The functional sizes of the functional processes are next aggregated to obtain the functional size of the software being measured.

2.3.2 Upgrade model case :

With the arrival of ECU’s architecture standards as AUTOSAR, there is now the possibility of developing the same code for different hardware targets. In this context, more and more software is reused from one ECU to another one. So beside new software developments there are more and more software enhancements.

A software enhancement functional size measurement is obtained by comparing the two specifications releases, what we called release N and release N-1. The COSMIC method measures differences between the enhanced specification and the previous one. When modifications affect functional processes, data movements or elementary blocks in the release N, they are caught by COSMIC measures.

3. AUTOMATED MESURE

The COSMIC measurement phase is all automated at Renault. A tool has been developed in Matlab language. The COSMIC Measurement Tool makes it possible to measure a size, in CFP, from a Simulink model and generates three kinds of reports (Classical, Detailed and Debug). It is possible to measure a group of specifications simultaneously.

The COSMIC Measurement Tool is configurable. Thus, the tool is able to adapt to different design rules (naming rules, specific memory blocks ...). It is possible to adapt the measurement strategy, depending on measurement goals, the configuration could be different. For example, for workload estimation,
libraries are considered as Elementary Blocks (COTS). But for memory size estimation, these blocks must also be measured.

The tool can generate three kinds of output files:

- A classical file containing following information: Number of functional processes in the measured specification, COSMIC Functional size of the specification (in CFP), Number of Reads, Number of Writes, Number of Entries, Number of Exits, The used configuration, Some information added by the measurer (Project, ECU …)
- A detailed file containing previous information plus: COSMIC Functional size of each functional process, Number of Reads in each functional process, Number of Writes in each functional process, Number of Entries in each functional process, Number of Exits in each functional process
- A debug file containing every measured data movement.

4. USES

The main goal of this part is to give a feedback on the Renault experimentations with the COSMIC method. Different uses of the COSMIC Function Points will be exposed.

4.1 Coding and Unit Testing workload estimation using COSMIC.

4.1.1 Workload estimation

The workload estimation using COSMIC is based on productivity models which are obtained by statistic methods. A linear relation between COSMIC size and related development workload is established. Based on past development, productivity models can allow to estimate future development workload.

Regarding Renault experience, the workload used was including coding and unitary tests. However, it could be possible to include validation workload.
Renault uses also productivity models for managing internally developers’ teams productivity in the Renault Nissan Technical Business India.

4.1.2 Supplier cost & productivity.

Productivity models are used to challenge suppliers. This process can go until contracting on a productivity model for future developments. It gives a clear rule and support for negotiation.

![Workload vs Purchase Department Negotiation diagram]

4.2 Specification development & productivity

The COSMIC size is also used in house to manage specifications’ productivity. The powertrain software department has a large database including the size of each specification in matlab/Simulink dedicated to Engine Control Module ECU. The relationship between available FTE (Full Time Equivalent) and volume of specification in matlab/Simulink is established.

<table>
<thead>
<tr>
<th>Delivery reference</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume of specification (basis on 100)</td>
<td>68</td>
<td>72</td>
<td>75</td>
<td>77</td>
<td>81</td>
<td>86</td>
<td>89</td>
<td>91</td>
<td>98</td>
<td>100</td>
</tr>
<tr>
<td>Volume of updated specification</td>
<td>10</td>
<td>16</td>
<td>13</td>
<td>12</td>
<td>15</td>
<td>17</td>
<td>17</td>
<td>19</td>
<td>23</td>
<td>20</td>
</tr>
<tr>
<td>Related specifications</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>added specifications</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Stability spec / L-1</td>
<td>85%</td>
<td>77%</td>
<td>82%</td>
<td>85%</td>
<td>82%</td>
<td>80%</td>
<td>81%</td>
<td>79%</td>
<td>76%</td>
<td>80%</td>
</tr>
<tr>
<td>Stability package / L-2</td>
<td>62%</td>
<td>59%</td>
<td>60%</td>
<td>59%</td>
<td>62%</td>
<td>60%</td>
<td>58%</td>
<td>59%</td>
<td>54%</td>
<td>55%</td>
</tr>
<tr>
<td>Volume of CFP</td>
<td>70</td>
<td>73</td>
<td>78</td>
<td>79</td>
<td>85</td>
<td>89</td>
<td>93</td>
<td>99</td>
<td>104</td>
<td>100</td>
</tr>
<tr>
<td>Workload (CFP)</td>
<td>14</td>
<td>20</td>
<td>24</td>
<td>19</td>
<td>23</td>
<td>25</td>
<td>26</td>
<td>29</td>
<td>31</td>
<td>26</td>
</tr>
<tr>
<td>FTE (Multi)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTE (Cycle)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of specification / FTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFP producted / FTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The performance of the Software Department is controlled and the objectives are updated.
4.3 Memory size estimation

COSMIC method is one of the elements to predict the memory size needed for embedded software. The COSMIC size is used for memory size prediction before obtaining code (several months earlier in a project). The process secures the ROM occupation in ECU at “Start of production” milestone and anticipates the needed margin for serial life.

The control is based on charts depending on selected compiler:

4.4 Value of software functions.

To deal with the increase of software development complexity, OEMs (Original Equipment Manufacturers) and Suppliers used to exchange software function.

Renault has chosen to integrate the COSMIC size in the official internal formula which evaluates the value of a software function (in addition to “innovation level” etc…).
4.5 Control of the COSMIC model precision.

- For Workload estimation use, the productivity models are realized on data set based on past projects with statistical methods as linear regression method.

At the end of a new software development, the difference between initial used correlation and real new COSMIC stored values has to be lower than 5%.

- For the value of software functions, the Renault method based on COSMIC has been compared with other Partner method based on software and tuning workload measures. The precision of Renault result is 12%.

5. CONCLUSION

The automation tool should help in applying and using the selected measurement procedure in industry. In fact, the automation allows us to avoid mistakes and measurers interpretations. In addition, it is now possible to get the size of an entire large ECU in few hours. At Renault SAS, the developed tool helps to deploy the COSMIC method in different ECU project teams. This deployment is made progressively with the deployment of the model based design approach. The deployment in the Power-train ECU was very beneficial. Some other uses appear (COSMIC size prediction, managing specification productivity ...). COSMIC is a good metric to help us to predict workloads of software projects. COSMIC is becoming our software reference metric.

References: