APPLICABILITY OF FULL FUNCTION POINTS FOR SIEMENS AT

-Technical Report, v1.0 (valid) -

In cooperation of
Reiner R. Dumke, Erik Foltin
University of Magdeburg
FIN/IVS
Universitätsplatz 2
D-39106 Magdeburg
E-mail: {dumke/foltin}@ivs.cs.uni-magdeburg.de

SIEMENS AG Automotive
AT PT GS TES2 "Software Factory Rbg", Stefan Weber
AT SE T2 "AT-OPAL", Ulrich Schweikl
Postfach 10 09 43
D-93009 Regensburg

1 MANAGEMENT SUMMARY

It is widely acknowledged that software metrics are a useful tool to support the decision making process in the IT industry.

There are thousands of metrics proposed in literature aimed at different artifacts of the software development process like product metrics (i. e. size of a program), process metrics (i. e. CMM level assessment) or resource metrics (i. e. experience of staff in a specific application area).

From a management point of view, productivity metrics are of particular interest

- to control the expenses,
- to analyze the performance of the amounts allocated to software development and
- to benchmark against other organizations.

In most software cost estimation and productivity models, software size is the key cost driver. Software size can be described from different perspectives. There are either

- technical measures, which are dependent upon technical development and implementation decisions (the measure "Lines of Code" is a typical example) or
- functional measures which assess or measure the size of a product or service from a user's (or functional) perspective.

Such a functional measurement technique, Function Point Analysis (FPA), has been (and still is) used extensively in productivity analysis and estimation in the MIS area. FPA, however, works best only in this area; it falls short to account for the additional level of complexity introduced by real-time requirements.

The Full Function Point technique is a recent attempt to fix this problem and to expand the applicability of FPA to other areas of software development.

Full Function Point is a functional measurement technique based on the standard Function Point analysis. The (rather small set) of Function Point rules dealing with control concepts have been expanded considerably.

The following report will give an overview of core concepts and definitions of the FFP technique and first experiences in applying the FFP technique to a typical Siemens AT product. A set of rules will be given to illustrate the feasibility of a mapping from the requirement specification document on to the FFP model. The report concludes with an estimation of the impact of FFP on the software development process at Siemens AT.

Resume: The Full Function Point technique combines the proven concepts of Function Points Analysis with real time extensions: First industrial experience has demonstrated, that it is a promising tool to capture the size of embedded and/or real time software.

2 FFP OVERVIEW

FFP was designed to measure the functional size of both MIS and real-time software. Since FFP is an extension of the FPA measurement method, most FPA rules are included as a subset for the measurement of the MIS functions within a measured software product. In the following we will concentrate on the real time extensions, for an overview of FPA concepts see for instance [IFPUG94].

The following summarizes the measurement concepts of the real time extensions of FFP:

- FFP measures the functional size from a functional perspective instead from a (narrower) perspective of an external user
- FFP measures the functionality of a process delivered not only to human users but also to mechanical devices
- Each process is decomposed into sub-processes responsible to read and write data to and from control data groups and to exchange data with the external users of the application
- Each individual sub-process is measured; therefore there is no limit to the size of a specific process.

Figure 1 shows all elements of the FFP technique. For the sake of completeness, the standard FPA elements are included as well.

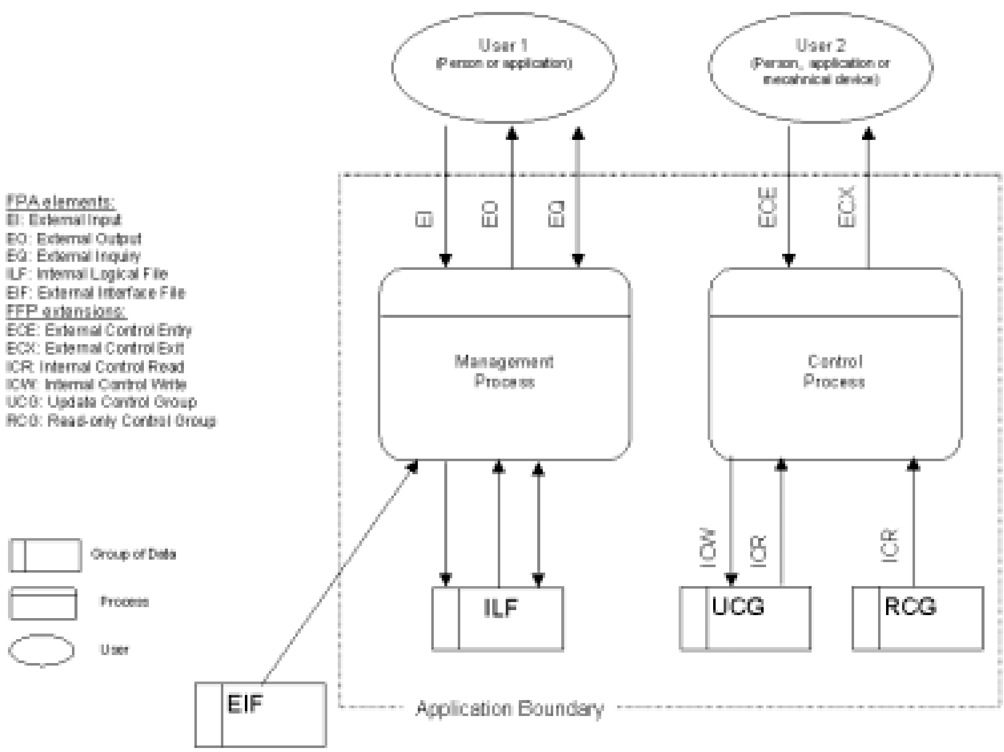

Figure 1- Diagram of FFP elements

New function types address the control aspect of real-time software: two new Control Data Function Types (UCG and RCG) and four new Control Transaction Function Types addressing the sub-processes of real-time software (ECE, ECX, ICR and ICW).

An **Update Control Group** (UCG) is a group of control data updated by the application being counted. It is identified from a functional perspective. This means, that the group of data appears in the (completed) requirement specification. The control data live for more than one transaction, this means it is stored for later use (by either the process creating the data or other processes of the application) and do not comprise intermediate processing results.

A **Read-only Control Group** (RCG) is a group of control data used, but not updated, by the application being counted. It is identified from a functional perspective; the control data live for more than one transaction.

Each control process is decomposed in sub-processes responsible to enter and exit control data into or from the process:

An **External Control Entry** (ECE) reads a group of logically related control data originating outside the application boundary.

An **External Control Exit** (ECX) sends a group of control data outside the application boundary to a human user or a mechanical device.

An **Internal Control Read** (ICR) reads a logically related group of control data from either a control data group (either UCG or RCG).

An **Internal Control Write** (ICW) writes a logically related group of control data to an Update Control Data group.

3 FIRST APPLICATION OF FFP IN THE SIEMENS AT ENVIRONMENT

3.1 Requirements for counting

For a successful (and repeatable) FFP count, the following conditions have to be met:

1. An adequate source of functional information must be available for the counter
2. The counter should be familiar with the FFP (FPA) concepts and definitions
3. (Preferably) the counter should either have some experience in the application area or application experts available for consultation

The following will demonstrate how to map a typical requirement specification to FFP concepts.

3.2 Guideline for mapping of FFP concepts to the requirements specification

3.2.1 Overall counting procedure

As the requirement specification contains a series of functional process descriptions, the following order should be used to conduct a FFP count.

1. Identify processes
2. Process evaluation for every identified control process:
   - 2.1. Identify the sub-processes
   - 2.2. Identify the ordering of sub-processes
   - 2.3. Count the sub-processes to determine their contribution to the overall function point count using the FFP rules
   - 2.4. Determine the UCG/RCG data elements defined or used by this process that have not been accounted for yet
3. Determine the contribution of the identified UCG/RCG control data groups to the overall function point count using the FFP rules

In general, the FFP method uses a functional perspective. This perspective may be biased by the counter's experience with the application area. Therefore, the following should not be treated as strict rules (this would require a formal specification) but to demonstrate a possible mapping of the Siemens AT requirement specifications to FFP concepts.

3.2.2 Identification of processes

- Every (sub-) chapter of the specification documentation including a description of delivered functionality is a potential candidate for a process.
- A process consists of a series of transactions. It generally processes input data and produces some output data. Each sub-chapter starting with a table of Output data followed by a table of Input data is very likely to be a process definition
- After the identification of process candidates, check if the candidates are indeed distinct, self contained processes or, instead, a succession of subordinated processes. If the latter is the case, merge all subordinated processes into one process.

- For each identified process, check whether it is a management or a control process. For management processes, the FPA counting rules have to be used. For control processes, the FFP counting rules have to be used.

3.2.3 Control Process evaluation

Once a control process has been identified, its sub-processes must be identified. The following elements of the process description in the requirement specification help to identify the sub-processes:

- The table "Output data". The table of output data contains control data that is either used for further processing by processes of the application or is used to control devices outside the applications boundary. To distinguish whether other processes use the control data, the index reference of the requirement specification should be consulted. If the output data item in question is used by other processes, it is definitively written to the UCG of the application by an Internal Control Write sub-process. If the data item in question is never used by this or any other process of the application it is possibly sent (by an External Control Write sub-process) outside the application boundary. These data definitions are usually to be found in chapter 2 (Basic SW Inputs and Outputs) of the requirement specification.
  If the data item in question is not used by other processes and it does also not exit the application boundary, check whether the data item comprises an inconsistency in the requirement specification or it is generated for debugging or diagnosis. If the latter is the case, treat the data item as written to the UCG by an ICR sub-process only if the debugging or diagnosis feature is an explicit functional requirement.
- The table "Input data". The table of input data contains control data items that are either read by the process from a group of control data (UCG or RCG) via an Internal Control Read sub-process or enter the process from outside the application boundary via an External Control Entry sub-process. To distinguish between Internal Control Read and External Control Entry the index reference of the requirement specification can be used.
- The tables "Configuration data" and "Calibration data". These tables contain data that control the behavior of a process. Configuration data and calibration data items remain constant and are not updated throughout the application execution.

The "Functional Description" section of the process description has to be examined to identify the order in which (groups of) control data enter or exit the process. The FFP model is closely related to the concept of signal flow. In case a Signal flow diagram is contained in the functional description, the groups of data entering or leaving the process can be derived directly from this diagram. Figure 2 shows a (simplified) process signal flow diagram.

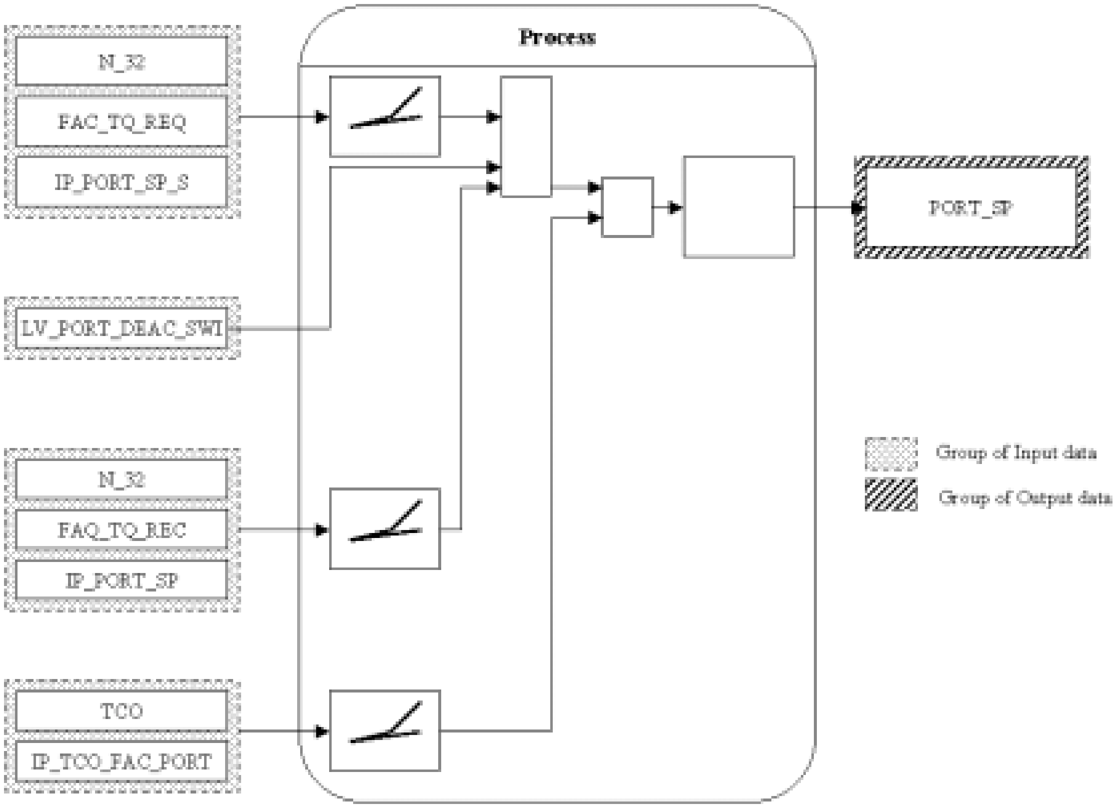

Figure 2- Sample Signal Flow Diagram

The input data for this process are not coming from outside the application boundary; the sub-processes entering them into the process are therefore Internal Control Reads. For FFP counting purposes there is no distinction between data items that are calibration data and data items that are intermediate results from other processes. The internal processing logic of a process is only considered insofar as it is necessary to identify groups of logically related (input and output) data. In this example, four groups of logically related input data (with one to four data items) entering the process via four distinct Internal Control Read sub-processes can be identified.[1]
The output of this process is identified as intermediate result: it will be stored for use by other processes and is not sent outside the application boundary. The sub-process that writes the output data is therefore an Internal Control Write.

If the Signal flow diagram is not provided for a process, an implicit signal flow model has to be derived from the (verbal) algorithm description and/or the formula section. Depending on the accuracy of the documentation more or less application knowledge is required.

3.3 Actual counting experience

Using the mapping above, a sample count has been conducted. Basis of the counting was the specification document [SIM 98]. From this specification the chapter 4 (System Variables) has been counted.

Within this chapter, 25 functional processes have been counted, amounting for a total of approximately[2] 130 FFP. All processes were identified as control processes. The processes contributed 40 items of updated control data and used another 60 items of read-only control data. The counting took about 4.5 hours with the

[1] Sub-processes dealing with rather small groups of data are typical for real-time/embedded systems.

[2] Some ambiguity in the specification prevented a reliable, exact number.

counter having only little insight into the application area.[3]

The following (positive) findings can be stated after this first application of FFP:

- In general, the Siemens AT specification document structure is very well suited to the FFP technique: the chapter structure is almost unambiguously identical to the FFP concept of processes.
- Using a few rules of thumb to map the specification to the FFP concepts allowed a count with almost no insight into the application area.[4]
- In fact, using these rules allowed detecting inconsistencies within the specification documentation.
- All elements of the FFP real-time concept are identifiable within the specification documentation.
- On the other hand, the specification documentation contained little information that seemed to introduce some complexity that could not be mapped to FFP concepts (besides algorithmic details: the algorithms are simplified and mapped to a sequence of sub-processes)
- Most problems with counting details could be identified (with the help of application experts) as a result of ambiguity in the specification documentation.
- Therefore, a more consistent documentation and/or more application experience should increase both the counting speed and accuracy.

It must be noted, however, that the above statements are the result of a first judgement and should be treated as hypothesizes. Some more counting experiments have to be carried out in order to:

- see whether there are still unclear counting details when using a more mature specification documentation (a brief inspection of Chapter 2 of the [SIM 99] specification looks promising)
- Verify if the above counting guidelines are applicable to all chapters of the requirement specification documentation.

It must be further noted that this judgement of the suitability of the FFP method for Siemens AT purposes is a mere technical one. A crucial issue is to discuss the results of the counting experiment(s) with application experts to see whether the FFP counts derived from the specification match their (perceived) idea of size for real time software.

4 Conclusions

4.1 General

- FFP is dependent on the quality of the functional information. Based on the counting experiment carried out, the quality of the functional information available for this experiment was adequate.
- The FFP count should be treated as estimates. The FFP concepts allow for some degree of interpretation to deal with uncertain and fuzzy information that is typical for early development stages[5].
- Identifying the groups of data is very crucial for a repeatable count. Signal flow diagrams are a very helpful tool to support the FFP counter to identify logically related groups of data. Verbal (informal) functional descriptions are likely to introduce ambiguity.
- Application experts have to be consulted to see if their perception of size correlates with the FFP count.
- The FFP count is one sole number. It can not be expected to be the magical number to control all aspects of the software development process.

The main aims of the Full Function Points technique are to provide a size indicator for real-time and embedded software systems and to

[3] This time does not include the time to become acquainted with the specification document. There was also no report created (

[4] It must be noted, however, that most of the evaluated processes were rather small (based on both documentation length and FFP count).

[5] Experiments with Function Point Analysis have shown, that the deviation in counting results based on the same information is in the range of 5 per cent for counters with similar experience. As FFP concepts are comparable, the same should be expected for FFP counts.

provide a consistent guideline to obtain the Function Point count.

The mere FFP number has to be put into perspective. It can be used to:

- Compare the size of different applications/versions
- Trace the requirement growth throughout the software development cycle
- Act as an indicator for specification maturity
- Serve as the basic input (key cost driver) in software cost estimation/productivity models. This embedding allows to:
    1. Assess the functional size of requirements delivered to software organizations
    2. Assess the performance of software organizations

There should be little additional effort required to obtain the FFP count. One possibility is to include the FFP counting in the inspection process. The FFP technique requires some structured way to look at the requirement definition, which actually seems to ease the review process.

The following strategy should be pursued to implement FFP at Siemens AT:

1. Conduct a few more counting experiments to
    - 1.1. Gain experience in FFP application
    - 1.2. Detail the counting guidelines
    - 1.3. Provide sample counts for discussion
2. Discuss the counting results with application experts to check whether their expectations are met
3. Identify milestones for conducting FFP counts within the development project
4. Identify candidates for pilot projects
5. Provide training and tools to assist FFP counting
6. Collect FFP and performance data to establish cost and productivity models

4.2 Transferability to other AT applications

This report is based on experience gained in the Siemens AT Powertrain environment. It should be expected that the FFP technique is applicable to other Siemens AT GG's/GZ's providing (compared to PT):

- A similar mixture of control, state driven and network functionality (or a subset) and
- A similar maturity of the specification structure

Some more effort has to be spent to evaluate the applicability of FFP for systems containing not only embedded functionality but also "classical" MIS functionality (e. g. data base handling, graphical user interfaces) like driver information systems.

5 References

[FFP 97] *Full Function Points: Counting Practices Manual,* Technical Report 1997-04, Université du Québec à Montréal, 1997.

[IFPUG 94] *Function Point Counting Practices Manual, Release 4.0,* International Function Points Users Group – IFPUG, Westerwille, Ohio, 1994.

[SIM 98] *Engine Management System SIMTEC 70*, SW version 710SB000, SIEMENS Automotive System Group, April 1998.

[SIM 99] *Engine Management System SIMTEC 81/82*, SW version 0203001A, SIEMENS Automotive System Group, April 1999.