Measuring the functional size of real-time software

Co-authored by:
A. Abran, J.-M. Desharnais, S. Oüigny
UQAM - Software Engineering Management Research Laboratory,
Centre d’Intérêt sur les Métriques (C.I.M.)

January 1999
Presenters profile

- Serge Oligny, M.Sc.

- Director - Technological innovations, UQAM-Software Engineering Management Research Laboratory

- CIM - Counting Practices, Executive Committee

- Formerly Corporate Manager of software development in the pulp & paper industry

- 13 years experience in the IT consulting market
Agenda...

- Introduction
- Characteristics of real-time software
- The measurement process model
- Measurement Procedures:
  - Measurement boundary
  - Measurement Scope
  - Identifying elements to be measured
  - Assigning points
- Overview of field tests results
- Conclusion
Introduction...

- Functional size measurement
- Origins and evolution
- Characteristics of FFP
- An analogy
Functional Size Measurement

- ISO/IEC/JTC1/SC7 Standard #14143 definition:

  “Functional Size: A size of software derived by quantifying the functional user requirements”
Origins and evolution...

A historical perspective...

1975, Allan Albrecht proposed Function Point based on study of MIS systems

1979, First publication on Function Points outside IBM

1983, IEEE contribute to publicize Function Points

1984, IFPUG created, Function Points become "users owned"

1994, current (4th) version of Function Points released

1993, First ISO meeting on FSM standard (14143)

1997, First version of Full Function Points released

1997, First version of Full Function Points released
Characteristics of FFP...

- FFP is a Functional Size Measure
- Focused on the ‘User functional view’
- Applied at any time during the software development life cycle
- Derived in terms understood by users
- Derived without reference to:
  - effort
  - methods used
  - physical or technical components.
Introduction...

An analogy...

2000 sq. ft.

500 FFP

4000 sq. ft.

1000 FFP

Software Functionality

Software Functionality
Characteristics of real-time software

- Different types of software
- Real-time or embedded software
- Limitations of IFPUG 4.0 Function Point
Characteristics of real-time software...

Different types of software

BUSINESS

MIS

Embedded or Real-time software

Utility

Users tools

Dev. tools

SYSTEM SOFTWARE

INFRASTRUCTURE
Characteristics of real-time software...

Real-time or embedded software

- **Timing**
  - Tight constraints on the rate of execution and on the timing of tasks
  - Explicit constraints on timing
  - Dedicated components to manage timing
  - Correctness of the result is linked to timing

- **Interaction with**
  - Mechanical devices
  - People
  - Other applications
Limitations of IFPUG 4.0 FP

Compared to MIS software...

<table>
<thead>
<tr>
<th>USERS</th>
<th>People</th>
<th>Other software</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Permanently stored (files, DB, ...)</td>
<td>Not stored permanently (signals, ...)</td>
<td></td>
</tr>
<tr>
<td>PROCESSES</td>
<td>No. of sub-processes varies a lot</td>
<td>Processes role is not easily classified as input, output or inquiry</td>
<td></td>
</tr>
</tbody>
</table>

IFPUG Function Points (4.0), do not adequately measure the functional size of real-time software
The measurement process model

- Overview of the measurement process
- Notes on measurement purpose...
- Notes on measurement strategy...
- Notes on documentation to be used...
Overview of the measurement process

The measurement process model...

WHY measure?
- estimate,
- support,
- replace,
- test,
- evaluate,
...
The measurement process model...

Notes on measurement purpose

- Identify the **business issue** which needs to be addressed
  - ...to estimate the size of a development project,
  - ...to determine the functionality supported by the maintenance team,
  - ...to determine the amount of functionality required to support day to day work activities of a user,
  - ...to determine replacement costs of software portfolio,
  - ...to assist in determining system testing strategies,
  - ...to assess the size of development backlog,
  - ...to determine mandatory functionality for package evaluation.
The measurement process model...

Notes on measurement purpose

- **Determine:**
  - what questions need to be answered by the size measure,
  - which software applications need to be sized
  - what components of the software will be included or excluded
Notes on measurement strategy

Identify:

- Which software is to be sized,
- How the sizing will be performed,
- Who will do the sizing,
- Who will assist as the application expert,
- Which Functional Size Measurement method will be used e.g. Full Function Points (FFP) Version 1.0,
- When and where will the sizing take place,
- Which software tools, counting forms, will be available.
Notes on documentation to be used

- **Planned Applications** (New development)
  - requirements specification
  - logical design specification
  - report layouts
  - screen layouts
  - logical data model

- **Existing Applications** (Enhancements)
  - all of the above plus
  - user manual
  - access to application online
Measurement Procedures

- APPLICATION BOUNDARY
- MEASUREMENT SCOPE
- IDENTIFYING ELEMENT TO BE MEASURED
- ASSIGNING POINTS
- EXERCICES
Application boundary

- **Definition of BOUNDARY**: 
  
  ‘a conceptual interface between the software under study and its users’

- **Definition of USER**: 
  
  ‘Any person that specifies Functional User Requirements and/or any person or thing (hardware, equipment, other applications) that communicates or interacts with the software at any time’

Application boundary

Boundary is:

- ‘membrane’ through which the transactions pass into and out of the software,
- external limitation of the software,
- point where the software stops and the external user world starts.
Application boundary

External “user world”

Internal “software world”
Application boundary

- Boundary may be illustrated on an application boundary diagram similar to a ‘context diagram’

- Identify all major groups of data movements between the boundary of this software and:
  - its human user operators,
  - and the boundaries of other applications or other hardware devices
Basic concepts - 1...

Application boundary

Operators

Application ‘A’

Software

Application ‘B’

Hardware Devices
Application boundary

Application Boundary

Equipment Control System

Other Software Applications

Configuration Parameters
Status
Control
Actuators

Incoming Calls
Sensors
Alarms
Status Parameters

灯光
蜂鸣器
使用数据
报告
警报

Buttons Parameters
Threshold Values
Responses

© 1998 UQAM-SEMRL, CIM
Definition of SCOPE:

“The set of functional features, inside the application boundary, for which the size have to be measured”

Measurement SCOPE is dictated by the PURPOSE of the measurement exercice.
SCOPE defines a sub-set of the software to be sized
Exercise

- Read the Case Study document and answer those questions:
  - what is the purpose of the measurement exercise?
  - what will be your strategy?
  - could you draw the boundary of the application?
  - what is the scope of the project?
Exercise : discussion

- What is the purpose of the measurement process?
- What will be your strategy?
- Could you draw the boundary of the application?
- What is the scope of the projects?
Identifying elements to be measured

- Identifying data
- Identifying transactions
Identifying elements to be measured

Identifying data

Key concepts

Identification rules

Summary
Identifying data

Key concepts

- Data selection
  Which ones are measured?

- Data occurrences
  How are they organized?

- Data activity
  How are data handled by the measured application?
Identifying elements to be measured

Identifying data

Key concepts - Data selection

تاندنوا Lit: a piece of data is processed but not saved or reused, it does not live for more than one transaction. This piece of data is not permanent and it is not measured.

تاندنوا Lit: If a piece of data is reused for multiple transactions, it lives for more than one transaction. This piece of data is measured.
Identifying elements to be measured

Identifying data

Key concepts - Data occurrence

- **Single occurrence** are groups of data which have one and only one instance of the record.

  ✓ Example: Data related to a time clock for a specific time.

- **Multiple occurrences** are groups of data which can have more than one instance of the same type of record. In real-time, multiple occurrences have the same structure than the one found in MIS System.

  ✓ Example: Flight record (black box)
Identifying elements to be measured

Identifying data

Key concepts - Data activity

- **Updated Groups of data**
  
  e.g.: add, change, delete, populate, revise, update, assign, create ...  
  A group of data may be updated by more than one application.

- **Read only Groups of data**
  
  The group of data is consulted by the application being measured without being updated. The group of data may be updated by other applications.
Identifying data

Identification rules

1- Select all logically related groups of data that live for more than one transaction.
   ✓ There is no formal definition of what is a logically related group of data
   ✓ From a normalization point of view our practice suggest that a logically related group of data could be at the second or third normal form, but not normalized more than the third normal form

2- Group data according to their structure
   ✓ Each multiple occurrences group is identified
   ✓ Merge all single occurrence together into one group
Identifying data activity for each identified group

- A **UCG** is a group of data updated by the application being counted.
- An **RCG** is a group of data used, but not updated, by the application being counted.

**UCGs and RCGs** are:
- groups of logically related data (multiple occurrences),
- groups of not necessarily related data (single occurrence)
- identified from a functional perspective and contain data that live for more than one transaction
Identifying data

Identification rules

4- Verify that Updated Control Group (UCG) and Read-only Control Group (RCG) ARE

✓ Files maintained by the user

BUT ARE NOT

✓ Sorting files
✓ Index files or secondary index
✓ Generated files sent to another application
Identifying elements to be measured

Summary

Identifying data

Single Occurrence

Live for more than 1 TXN

Multiple Occurrence

1 UCG

UCGs

1 RCG

RCGs
Identifying elements to be measured

Identifying transactions

- Key concepts
- Identification rules
- Summary
Identifying elements to be measured

Identifying transactions

Key concepts

- Process
- Transaction
- Trigger
- Sub-processes
Identifying elements to be measured

Identifying transactions

Key concepts - Process

“A set of operations or activities which acts on inputs to produce a result.”
Identifying transactions

Key concepts - Triggers

- An event which *initiates* a process from a functional perspective,
- An event occurring outside the application boundary,
- The manifestation of the event is data which enters the application boundary,
- Clocks and timing events can be triggers.
Identifying elements to be measured

Identifying transactions

Key concepts - Transactions

- A transaction is an instance of a process/sub-process,

- A transaction includes all processing associated with an occurrence of an external trigger.

Example: in a watch, each tick of the timing crystal is a trigger. All processing associated with each new tick is a separate transaction.
Identifying elements to be measured

Identifying transactions

Key concepts - Sub-processes

The smallest processing step identifiable from a functional perspective as either an entry, exit, read or write.
Identifying transactions

Key concepts - Sub-processes

- Identified from a functional perspective,
- Single sub-processes,
- Located at the lowest functional level of a process and acting on one group of data. If a sub-process acts on two groups of data, there are at least two sub-processes.
Identifying elements to be measured

Identifying transactions

Key concepts - 4 types of sub-processes

Process

ENTRY Sub-process

READ Sub-process

WRITE Sub-process

EXIT Sub-process

Transaction
Identifying transactions

Key concepts - 4 types of sub-processes

Users: Persons, Other Applications, Mechanical Devices

Entry (ECE)

Process

Exit (ECX)

Application Boundary
Identifying transactions

Key concepts - 4 types of sub-processes

- Read (ICR)
  - Users: Persons, Other Applications, Mechanical Devices
  - Groups of data read

- Process

- Write (ICW)
  - Users: Persons, Other Applications, Mechanical Devices
  - Groups of data written
Identifying elements to be measured

Identifying transactions

Identification rules: ECE

- The sub-process receives a group of data from outside the application boundary,
- The sub-process is associated with only one group of data,
- The sub-process does not exit, read, or write data,
- The sub-process is unique: processing and data element types identified are different from other ECEs within the same process,
- The primary trigger is an ECE.
Identifying transactions

Identification rules: ECX

- The sub-process sends data external to the application’s boundary.
- The sub-process sends only one group of data.
- The sub-process does not receive, read, or write data.
- The sub-process is unique: processing and data element types identified are different from other ECXs of the same process.
Identifying elements to be measured

Identifying transactions

Identification rules: ICR

- The sub-process reads a group of data.
- The sub-process reads only one group of data.
- The sub-process does not receive, exit, or write data.
- The sub-process is unique: processing and data element types identified are different from other ICRs of the same process.
Identifying transactions

Identification rules: ICW

- The sub-process writes to a group of data.
- The sub-process writes to only one group of data.
- The sub-process does not receive, exit, or read data.
- The sub-process is unique: processing and data element types identified are different from other ICWs of the same process.
Identifying elements to be measured

Identifying transactions

Summary

Each arrow is a sub-process.

* Entry 1 is the trigger
Identifying elements to be measured

Identifying transactions

Summary

- Trigger 1
  - Control Process 1
    - Sub process 1.1
    - Sub process 1.2
    - ...
  - Control Process 2
    - Sub process 2.1
    - Sub process 2.2

- Trigger 2
  - Control Process 1
    - Sub process 1.1
    - ...

Docum. ➔ Transactions ➔
Summary of Function Types

Identifying elements to be measured

**Management Process**
- EI: External Input
- EO: External Output
- EQ: External Inquiry
- ILF: Internal Logical File
- EIF: External Interface File

**Control Processes**
- ECE: External Control Entry
- ECX: External Control Exit
- ICR: Internal Control Read
- ICW: Internal Control Write
- RCG: Read-only Control Group
- UCG: Updated Control Group

**Symbols**
- : Process
- : Group of data
- : User
Assigning points to measured elements

○ Data points:
  ○ key concepts
  ○ Assigning points to data
  ○ Example
  ○ Quick validation of data measurement

○ Transaction points:
  ○ key concepts
  ○ Assigning points to transactions
  ○ Example
  ○ Quick validation of transaction measurement
Assigning points to measured elements

Data points: key concepts

Points are assigned to data as a function of two characteristics:

**DET:** The number of data elements

**RET:** The number of user recognizable subgroup of data elements
Assigning points to measured elements

Assigning points to data

- **Single Occurrence**
  - 1 UCG
  - 1 RCG
  - Live for more than 1 TXN

- **Multiple Occurrence**
  - UCGs
  - RCGs

- DET only
- DET & RET
Assigning points to measured elements

Assigning points to data

Single occurrence Updated data (UCG):

- Point assignment is based on the number of data element types (DET)

- Points = (number of DET / 5) + 5

Note: There is only one single occurrence UCG within an application. It comprises all the single occurrence updated values within the application being measured.
Assigning points to data

Assigning points to measured elements

Single occurrence Read-Only Data (RCG):

- Point assignment is based on the number of data element types (DET)
- Points = number of DET / 5

Note: There is only one single occurrence RCG within an application. It comprises all the single occurrence read-only values within the application being measured.
Assigning points to measured elements

Assigning points to data

Multiple occurrence RCG and UCG:

<table>
<thead>
<tr>
<th>DETs</th>
<th>1 - 19</th>
<th>20 - 50</th>
<th>51 +</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>L</td>
<td>L</td>
<td>A</td>
</tr>
<tr>
<td>2 - 5</td>
<td>L</td>
<td>A</td>
<td>H</td>
</tr>
<tr>
<td>6 +</td>
<td>A</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>
Assigning points to measured elements

Assigning points to data

Multiple occurrence UCG and RCG:

<table>
<thead>
<tr>
<th></th>
<th>UCG</th>
<th>RCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>L = Low</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>A = Average</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>H = High</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>
Assigning points to measured elements

Example - RCG mult. occ.

- Temperature Data
  - The temperature data for each cooking mode (Figure 2) is a multiple occurrence group of data, that is, there are more than one occurrence of the same type of record.
  - The temperature data are maintained outside the application boundary but referenced by the application to control the heater and the status indicators. This group of data could be therefore an RCG. The following table shows the evaluation of the RCG rules.

- Remember: All of the counting rules must apply (Yes) to count the group of data as a RCG.
### Example - RCG mult. occ.

#### RCG Counting Rules

<table>
<thead>
<tr>
<th>RCG Counting Rules</th>
<th>Does the Rule Apply?</th>
</tr>
</thead>
<tbody>
<tr>
<td>The group of data is either a logical related group of data or a single occurrence group of data</td>
<td>Yes. Temperature data are required to control the heater and the status indicator.</td>
</tr>
<tr>
<td>The group of data is not updated by the application being counted.</td>
<td>Yes. There is no process within the application that updates the temperature data.</td>
</tr>
<tr>
<td>The group of data is referenced by the application being counted.</td>
<td>Yes. The temperature data are referenced to control the heater and the status indicator.</td>
</tr>
<tr>
<td>The group of data lives for more than one transaction.</td>
<td>Yes. Each time the end user cooks rice, the temperature data are referenced.</td>
</tr>
<tr>
<td>The group of data has not been counted as an UCG, ILF or EIF for the application.</td>
<td>Yes. The rule applies because the group of data is not counted as an UCG, an ILF or EIF.</td>
</tr>
</tbody>
</table>
### Example - RCG mult. occ.

#### DET Counting Rules

<table>
<thead>
<tr>
<th>DET Counting Rules</th>
<th>Does the Rule Apply?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count a DET for each unique user recognizable, non recursive field on the RCG.</td>
<td>From the ‘temperature data by mode’ figure we can identify the following DET:</td>
</tr>
<tr>
<td></td>
<td>Mode                     Max. Temperature</td>
</tr>
<tr>
<td></td>
<td>Warming temperature     Cooking time</td>
</tr>
<tr>
<td>Count a DET for each piece of data in the RCG that exists because the user requires a relationship with another ILF or UCG to be maintained.</td>
<td>There is no data of this type.</td>
</tr>
<tr>
<td>Count physical implementation techniques as a single DET for the entire group of fields.</td>
<td>There is no field of this type.</td>
</tr>
</tbody>
</table>
Assigning points to measured elements

Example - RCG sing. occ.

- Selected cooking mode: This field keeps the cooking mode (fast, normal and gruel) selected by the end user. The default value of this field is ‘normal’. If the end user does not select a mode, the rice is cooked in normal mode. Various processes of the application need to reference the selected cooking mode. Therefore, the selected cooking mode lives for more than one transaction.

- Target temperature: During cooking, the application receives the actual temperature from a sensor and update the target temperature every 30 seconds. It is referenced every 5 seconds by the process which controls the heater. Therefore, target temperature lives for more than one transaction.

- Elapsed time: During cooking, the elapsed time is continuously updated. It is used by the processes which calculates the target temperature and controls the heater. Therefore, elapsed time lives for more than one transaction.
### DET Counting Rules

<table>
<thead>
<tr>
<th>DET Counting Rules</th>
<th>Does the Rule Apply?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count a DET for each unique user recognizable, non recursive, field on the UCG.</td>
<td>Selected cooking mode</td>
</tr>
<tr>
<td>Count a DET for each or piece of data in the UCG that exists because the user</td>
<td>Target temperature</td>
</tr>
<tr>
<td>requires a relationship with another ILF or UCG to be maintained.</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>Count physical implementation techniques as a single DET for the entire group of</td>
<td>There is no data of this type.</td>
</tr>
<tr>
<td>fields.</td>
<td>There are no fields of this type.</td>
</tr>
</tbody>
</table>

### Point assignment: RCG

Points = Integer part of (number of DET / 5)

Points = Integer part of (3 /5) = 5
Assigning points to measured elements

Quick validation of data measurement

Check if:

- All data live for more than one transaction
- Repeated fields have been counted only once
- Data updated in more than one application has been counted in each application
Points are assigned only at the level of the sub-process,

The functional size of a process is the sum of the points assigned to the set of its sub-processes.

Note that there is no upper limit to the size of a process.
Assigning points to measured elements

Assigning points to transactions

- Points assigned to a sub-process is a function of the number of DET manipulated by the sub-process

<table>
<thead>
<tr>
<th>DETs:</th>
<th>1 to 19 DETs</th>
<th>20 to 50 DETs</th>
<th>51 + DETs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Assigning points to measured elements

Example

Application Boundary

- Mode switch
- Start switch
- Clock 30 seconds
- Clock 5 seconds
- Temperature sensor

Process

- Indicator lamp
- Heater
Assigning points to measured elements

Example

Hints:

- Start with the triggers, identify all control processes link to each trigger,

- For each identified processes
  - Identify all sub-processes
  - Identify their transaction type
  - Assign points to each sub-process
Assigning points to measured elements

Example

Solution:

- Mode switch
- Start switch
- Temperature sensor
- Clock every 30 sec.
- Clock every 5 sec.
- Operation Status
- Indicator output
- Selected cooking mode
- Elapsed time
- Target temperature
- Difference calculation
- Difference information
- Heater ON/OFF
- Heater
Quick validation of transaction measurement

- Check that each process:
  - has at least one External Control Entry (ECE),
  - has at least one External Control eXit (ECX) or one Internal Control Write (ICW),
  - does not have duplicate sub-processes.
### Solution:

<table>
<thead>
<tr>
<th>Logical file - Process/sub-process</th>
<th>Function Type</th>
<th>RET</th>
<th>DET</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control data function type</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1: Temperature data (multiple occurrences)</td>
<td>RCG</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2: Single occurrence UCG (single occurrence)</td>
<td>UCG</td>
<td>N/A</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Control transactional function type</strong></th>
<th>Function Type</th>
<th>RET</th>
<th>DET</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Mode selection:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1: Receive cooking mode</td>
<td>ECE</td>
<td>N/A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.2: Update selected cooking mode</td>
<td>ICW</td>
<td>N/A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total points:</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2: Elapsed time:</th>
<th>Function Type</th>
<th>RET</th>
<th>DET</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1: Receive start signal</td>
<td>ECE</td>
<td>N/A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2.2: Update elapsed time</td>
<td>ICW</td>
<td>N/A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2.3: Read selected cooking mode</td>
<td>ICR</td>
<td>N/A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2.4: Read cooking time</td>
<td>ICR</td>
<td>NA</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2.5: Set the status indicator</td>
<td>BCX</td>
<td>N/A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total points:</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>
Assigning points to measured elements

Example

Solution:

<table>
<thead>
<tr>
<th>3: Target temperature:</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1: Clock trigger</td>
<td>ECE</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td>3.2: Read selected cooking mode and elapsed time</td>
<td>ICR</td>
<td>N/A</td>
<td>2</td>
</tr>
<tr>
<td>3.3: Read ‘temperature data’ file</td>
<td>ICR</td>
<td>N/A</td>
<td>4</td>
</tr>
<tr>
<td>3.4: Update target temperature</td>
<td>ICW</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total points:</strong></td>
<td></td>
<td></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4: Heater control:</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1: Clock trigger</td>
<td>ECE</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td>4.2: Receive actual temperature</td>
<td>ECE</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td>4.3: Read target temperature</td>
<td>ICR</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td>4.4: Set the heater ON/OFF</td>
<td>ECE</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total points:</strong></td>
<td></td>
<td></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

**Total FFP points = 25**
Overview of field tests results

- Sources of data
- First set: comparing FPA and FFP
- Second set: relevance and usability
- Third set: further comparisons FPA/FFP
Available resources

- **Complete documentation on the Web**
  - Concepts and definitions,
  - Counting Practice Manual,
  - Publications,
  - [http://www.lrgl.uqam.ca/ffp.html](http://www.lrgl.uqam.ca/ffp.html)

- **Support available**
  - Case Study
  - On site custom training
  - Consulting support
Overview of field tests results

First set

- Conducted by the research team in 1997,
- 3 RT or embedded products measured,
- 2 industrial partners participated,
- GOAL: Compare FFP with FPA (IFPUG 4.0)
# Overview of field tests results

## First set

### Results...

<table>
<thead>
<tr>
<th>PRODUCT 1</th>
<th>PRODUCT 2</th>
<th>PRODUCT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXN$^3$</td>
<td>Points</td>
<td>TXN$^3$</td>
</tr>
<tr>
<td>FPA$^1$</td>
<td>54</td>
<td>256</td>
</tr>
<tr>
<td>FFP$^2$</td>
<td>753</td>
<td>777</td>
</tr>
</tbody>
</table>

- **Note 1:** Using IFPUG 4.0 CPM, processes only
- **Note 2:** Using FFP 1.0 CPM, processes only
- **Note 3:** Number of processing transactions for which points are assigned
Overview of field tests results

First set

Observations:

- FFP results close to FPA when processes contain small number of sub processes,

- FFP yield larger size measures when processes contain large number of sub processes,

- Both methods require similar measurement effort
Overview of field tests results

Second set

- Conducted without assistance from the research team in 1997,
- Operational real-time products measured,
- 1 industrial partner,
- GOAL: Evaluate FFP for relevance and usability
Observations:

- Functional coverage established at 97%, based on expected number of functions to be measured.

- Concepts and procedures are:
  - Clear,
  - Easy to understand,
  - Usable without assistance of specialists
Overview of field tests results

Third set

- 4 industrial partners in North-America and Australia participated,

- 10 software products measured:
  - 8 products related to the telecom business
  - 1 product related to power utility
  - 1 product related to the military sector

- All products measured by the same individual (CFPS, 12 years exp. in FSM)
1st GOAL: Compare IFPUG 4.0 and FFP

RESULTS

<table>
<thead>
<tr>
<th>Product</th>
<th>Type</th>
<th>FPA size</th>
<th>FFP size</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Real-Time</td>
<td>210</td>
<td>794</td>
</tr>
<tr>
<td>B</td>
<td>Real-Time</td>
<td>115</td>
<td>183</td>
</tr>
<tr>
<td>C</td>
<td>Real-Time</td>
<td>N / A</td>
<td>2 604</td>
</tr>
<tr>
<td>D</td>
<td>Real-Time</td>
<td>43</td>
<td>318</td>
</tr>
<tr>
<td>E</td>
<td>Mostly MIS</td>
<td>764</td>
<td>791</td>
</tr>
<tr>
<td>F</td>
<td>MIS (batch)</td>
<td>272</td>
<td>676</td>
</tr>
<tr>
<td>G</td>
<td>MIS</td>
<td>878</td>
<td>896</td>
</tr>
</tbody>
</table>

Size is similar when measuring typical MIS software products.
Overview of field tests results

**Third set**

1st GOAL: Compare IFPUG 4.0 and FFP

## RESULTS

<table>
<thead>
<tr>
<th>Product</th>
<th>Type</th>
<th>FPA size</th>
<th>FFP size</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Real-Time</td>
<td>210</td>
<td>794</td>
</tr>
<tr>
<td>B</td>
<td>Real-Time</td>
<td>115</td>
<td>183</td>
</tr>
<tr>
<td>C</td>
<td>Real-Time</td>
<td>N/A</td>
<td>2604</td>
</tr>
<tr>
<td>D</td>
<td>Real-Time</td>
<td>43</td>
<td>318</td>
</tr>
<tr>
<td>E</td>
<td>Mostly MIS</td>
<td>764</td>
<td>791</td>
</tr>
<tr>
<td>F</td>
<td>MIS (batch)</td>
<td>272</td>
<td>676</td>
</tr>
<tr>
<td>G</td>
<td>MIS</td>
<td>878</td>
<td>896</td>
</tr>
</tbody>
</table>

One real-time software could only be sized with FFP
Overview of field tests results

Third set

1st GOAL: Compare IFPUG 4.0 FPA and FFP

RESULTS

<table>
<thead>
<tr>
<th>Product</th>
<th>Type</th>
<th>FPA size</th>
<th>FFP size</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Real-Time</td>
<td>210</td>
<td>794</td>
</tr>
<tr>
<td>B</td>
<td>Real-Time</td>
<td>115</td>
<td>183</td>
</tr>
<tr>
<td>C</td>
<td>Real-Time</td>
<td>N / A</td>
<td>2 604</td>
</tr>
<tr>
<td>D</td>
<td>Real-Time</td>
<td>43</td>
<td>318</td>
</tr>
<tr>
<td>E</td>
<td>Mostly MIS</td>
<td>764</td>
<td>791</td>
</tr>
<tr>
<td>F</td>
<td>MIS (batch)</td>
<td>272</td>
<td>676</td>
</tr>
<tr>
<td>G</td>
<td>MIS</td>
<td>878</td>
<td>896</td>
</tr>
</tbody>
</table>

Larger functional size for software products with numerous R-T processes (A, B and D); even for MIS with fewer direct user interactions (F).
Overview of field tests results

Third set

1st GOAL: Compare IFPUG 4.0 and FFP

What does it mean?

<table>
<thead>
<tr>
<th>MIS product</th>
<th>RT product</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPA</td>
<td>200</td>
</tr>
<tr>
<td>FFP</td>
<td>~ 200</td>
</tr>
<tr>
<td></td>
<td>&gt;&gt; 200</td>
</tr>
</tbody>
</table>

Obviously, when considering RT products, FFP is measuring functionality that is not measured by IFPUG 4.0.
Third set

2nd GOAL: Explore key economic ratios

RESULTS

<table>
<thead>
<tr>
<th>Product</th>
<th>Size (FFP)</th>
<th>Effort (ph)</th>
<th>Duration (mth)</th>
<th>Unit effort (ph/FFP)</th>
<th>Sched. del. Rate (FFP/ mth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>205</td>
<td>3 913</td>
<td>26</td>
<td>19</td>
<td>8</td>
</tr>
<tr>
<td>I</td>
<td>138</td>
<td>6 580</td>
<td>16</td>
<td>48</td>
<td>9</td>
</tr>
<tr>
<td>J</td>
<td>198</td>
<td>7 448</td>
<td>14</td>
<td>38</td>
<td>14</td>
</tr>
</tbody>
</table>

These 3 software products are all R-T software

Until further data is available to allow statistically significant analysis, these should be interpreted as “order of magnitude” figures.
Conclusion

- International recognition
- Benchmarking your results
- The future of Full Function Points
- Available resources
- Final remarks
- Acknowledgements
International recognition

In the Spring of 1998, FFP was accepted as a valid functional size measure by ISBSG*, an international benchmarking organization.

ISBSG: International Software Benchmarking Standards Group
Benchmarking your results

- Complete project
- Download Venturi
- Enter project data
- Send data to ISBSG
- Receive project benchmarking report (Designed for future estimating)
- Discount on ISBSG products

WWW

ISBSG Repository
The future of Full Function Points

- Version 2 on its way for 1999,
- Looking for more industrial partners for field testing,
- Looking for more industrial partners for data collection,
- International Counting Practice Committee,
- ISO 14143 certification to start in 1999.
Final remarks...

- FFP addresses a problem identified since 1986,
- FFP was designed for ISO compliance,
- FFP has been designed FOR the industry, WITH the industry,
- FFP is an open and transparent initiative, fully documented and easily available,
- FFP is already helping organizations manage their non-MIS software.
Overview of field tests results

Sources of Funding

Developing FOR the industry, WITH the industry, FFP industrial partners...

- Bell Canada, CANADA
- Northern Telecom, Canada & USA
- JECS Systems Research, JAPAN
- Hydro-Québec, CANADA
Acknowledgements...

- The Software Engineering Management Research Laboratory of the Université du Québec à Montréal is supported through a partnership with Bell Canada.

- Additional funding is provided by the National Science and Engineering Research Council of Canada.