Japanese Function Point User Group

‘COSMIC’ - the COmmon Software Measurement International Consortium

Introduction and Overview of Principles

Charles Symons

(on behalf of the COSMIC Core Team *)

16th April 1999

(* Alain Abran, Charles Symons, Carol Dekkers, Jean-Marc Desharnais, Peter Fagg, Pam Morris, Jolijn Onvlee, Risto Nevalainen, Grant Rule, Denis St Pierre)

Aims of this presentation

- To introduce the COSMIC project, its aims, structure and status
- To describe the agreed main basic principles of the family of new software functional sizing methods
- To invite interested organisations to participate in the next stage of field-testing of the COSMIC principles

COSMIC Aims

COSMIC is a project which aims to develop, test, bring to market and gain acceptance as an industry standard, a new generation of software sizing methods which are applicable

- **for performance measurement, e.g. productivity (=size/effort)**
- **as a component of estimating methods from early in a software item's life**
- **in as wide a range of software 'domains' as possible; priority to be given to business and real-time software (e.g. process control, telephony, embedded, etc.)**

COSMIC methods: desirable characteristics

COSMIC size measurement methods should

- **derive sizes from user requirements**
- **be academically sound and compatible with modern ways of stating requirements, but independent of specific methods**
- **be compatible with emerging ISO standards 14143**
- **draw on the best ideas of the IFPUG 4.1, NESMA, MkII and FFP methods**
- **produce sizes with known confidence levels**
- **be non-subjective, repeatable and thus automatable, but easy to apply manually**

The COSMIC project is necessary because better software sizing methods are needed

- **Current software sizing methods based on user requirements (i.e. 'Function Point' methods) are widely accepted in the MIS domain but are unsatisfactory as long-term solutions**
  - **limited acceptance outside the traditional MIS world**
  - **do not deal satisfactorily with 'non-functional' (= technical and quality) requirements**
  - **rules are not always compatible with modern software requirements and development methods and are sometimes subjective in interpretation**
- **The subject is of huge importance for software management and contracting.**

COSMIC project structure and status

- Project Team from six nations
- Started 1st November 1998
- Private initiative, currently largely self-funded
- Proposed basic principles now largely established
- Target to issue COSMIC outline Measurement Standard Manual in the second quarter of 1999
- UQAM Lab. (Montreal) working on Full Function Points V.2 to adopt COSMIC principles
- We will soon be ready to start wide-scale data collection for testing the proposed basic principles

The COSMIC Core Team comprises a broad range of academic and practitioner experience

		Experience	
Alain Abran	Canada	IFPUG, FFP, ISO WG12	Joint Project Leaders
Charles Symons	UK	MkII FP, ISO WG12	
Carol Dekkers	USA	IFPUG, ISO WG12	
Jean-Marc Desharnais	Canada	IFPUG, FFP	
Peter Fagg	UK	IFPUG, MkII FP, ISO WG12	
Pam Morris	Australia	IFPUG, FFP, ISO WG12	
Jolijn Onvlee	Netherlands	IFPUG, NESMA	
Risto Nevalainen	Finland	IFPUG, Laturi, ISO WG12	
Grant Rule	UK	IFPUG, MkII, FFP	
Denis St Pierre	Canada	IFPUG, FFP	

(and expanding)

Aims of this presentation

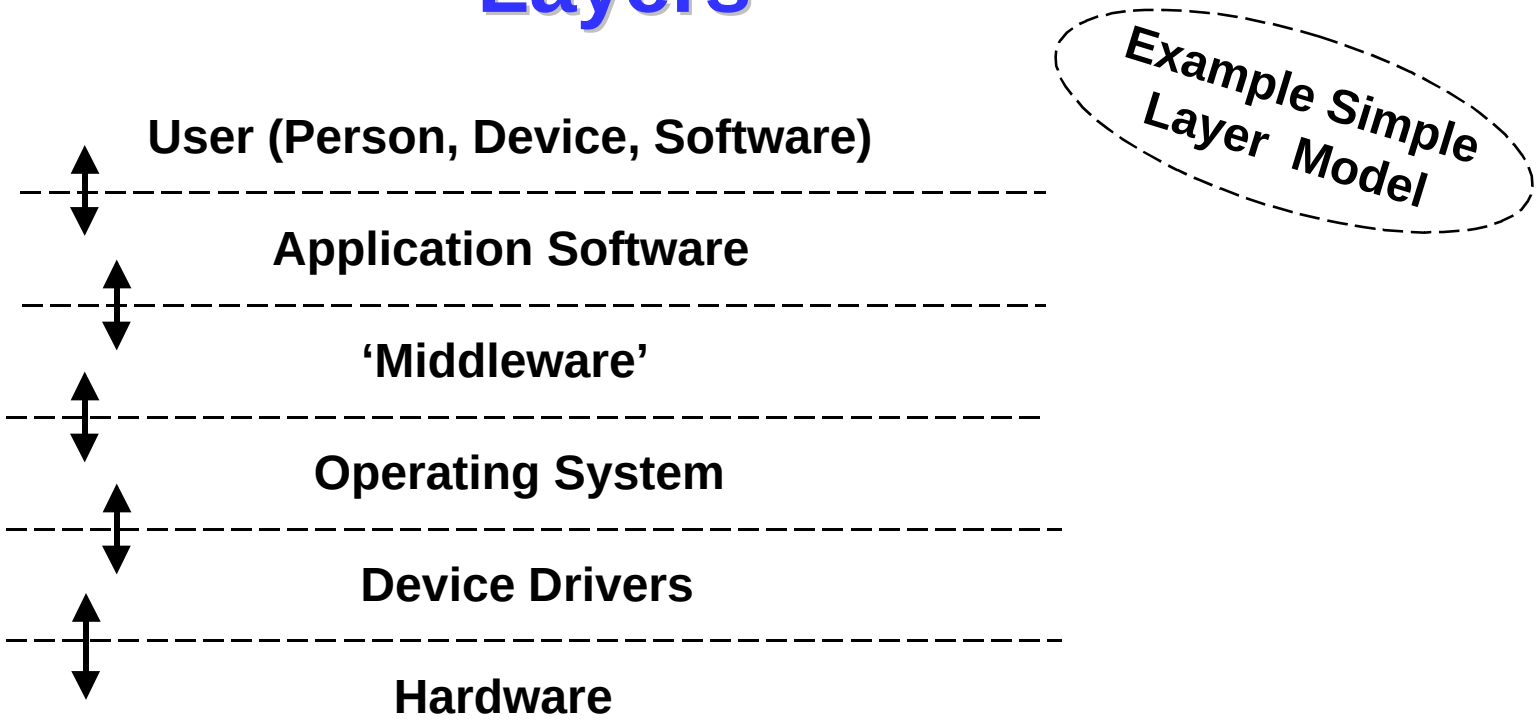
- To introduce the COSMIC project, its aims, structure and status
- To describe the agreed main basic principles of the family of new software functional sizing methods
- To invite interested organisations to participate in the next stage of field-testing of the COSMIC principles

COSMIC Basic Principles

Topics:

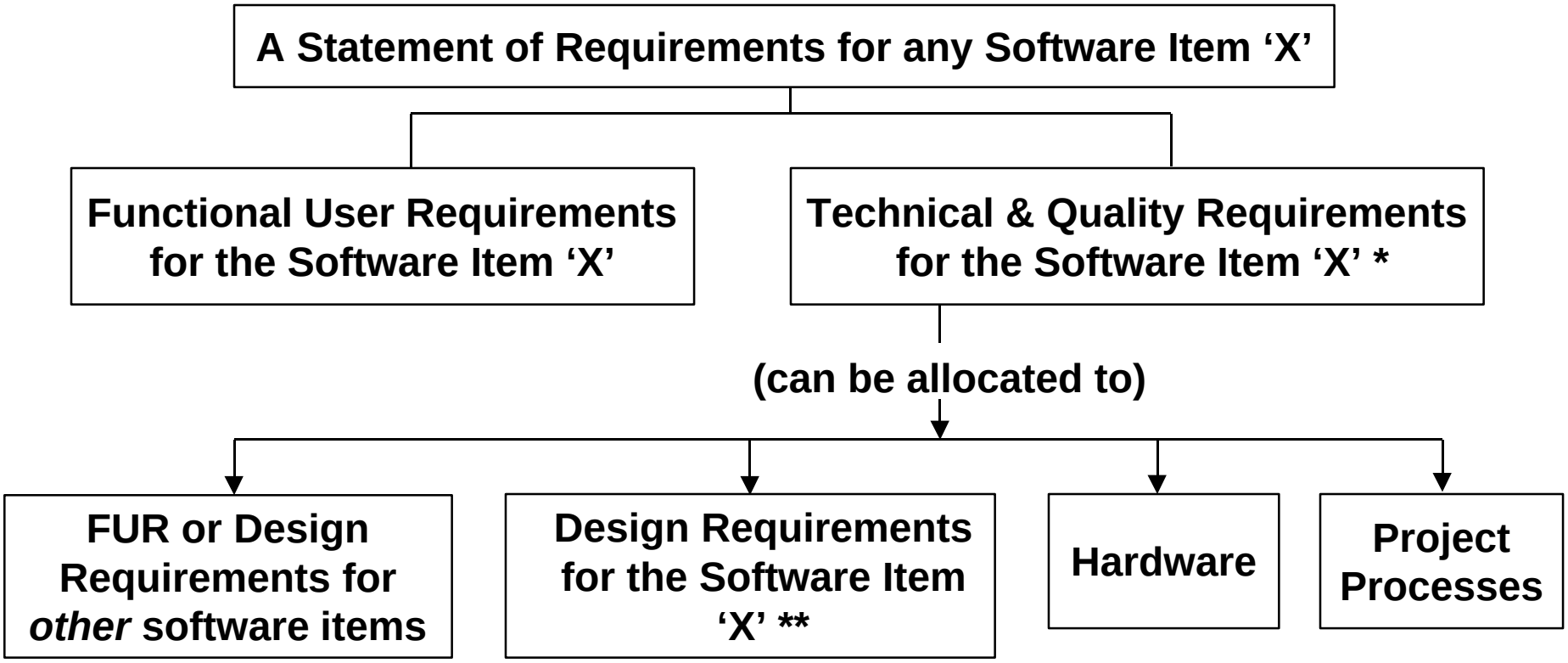
- **A general model of the structure of all requirements for a ‘Software Item’**
- **Layers of software and the boundary of the software to be sized**
- **A general model of ‘Base Functional Component Types’ of requirements**
- **Possible sizes of software requirements**
- **Determining the size of Base Functional Components**

Software is typically structured in several 'Layers'



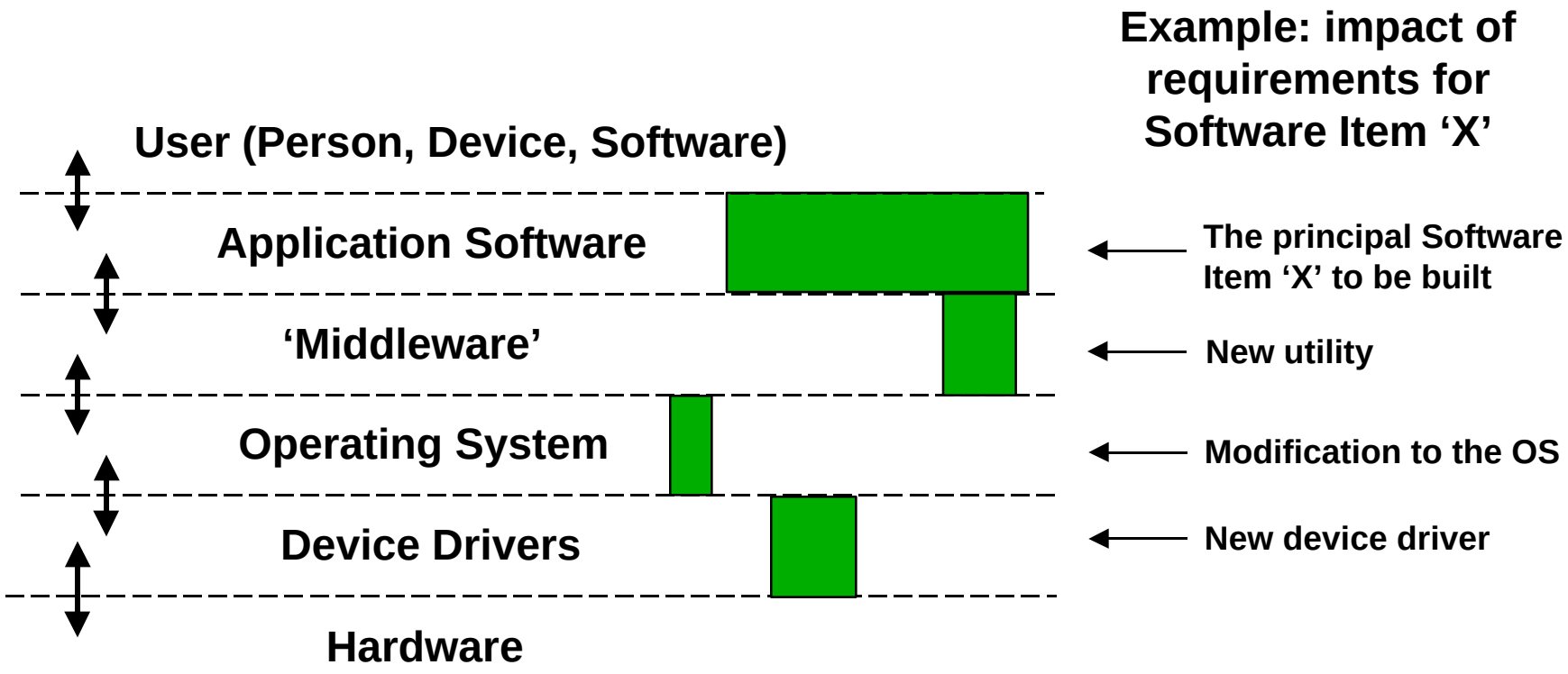
Existing Function Point methods were designed to size Application Software

A Statement of Requirements may be satisfied in many ways



*** e.g. a Technical Requirement such as 'process in batch mode' or 'Unix compatible', or a Quality Requirement such as 'Criticality' of the Software Item**
**** e.g. a requirement to perform at high efficiency**

The Requirements may impact software in several Layers



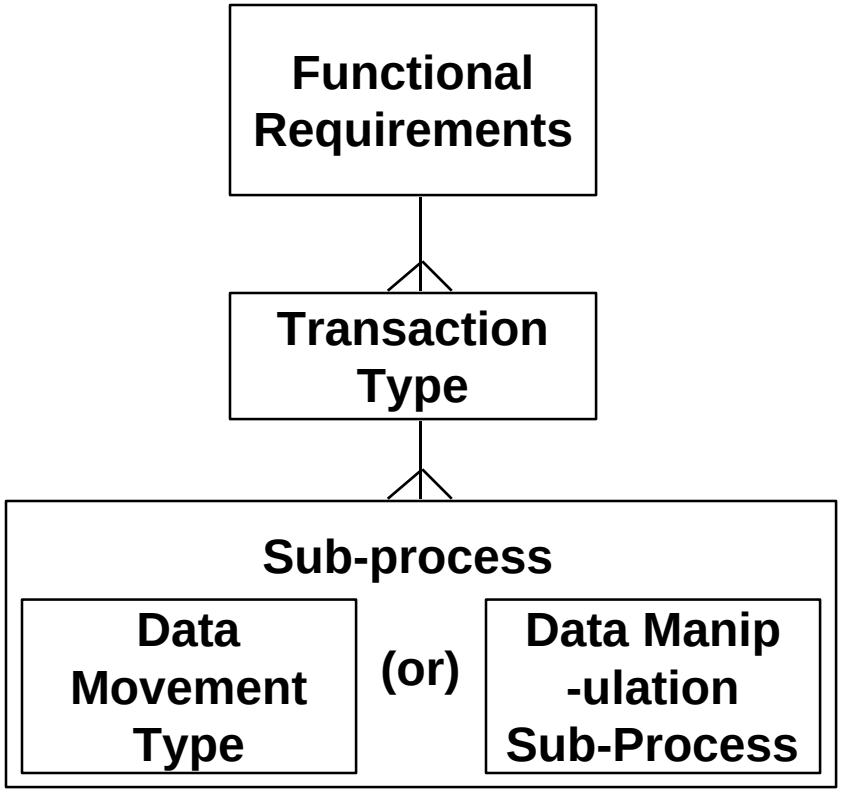
COSMIC aims to be able to measure the size-impact of requirements on software in any layer

Recognising the size of software components in other layers should allow COSMIC to simplify its sizing approach

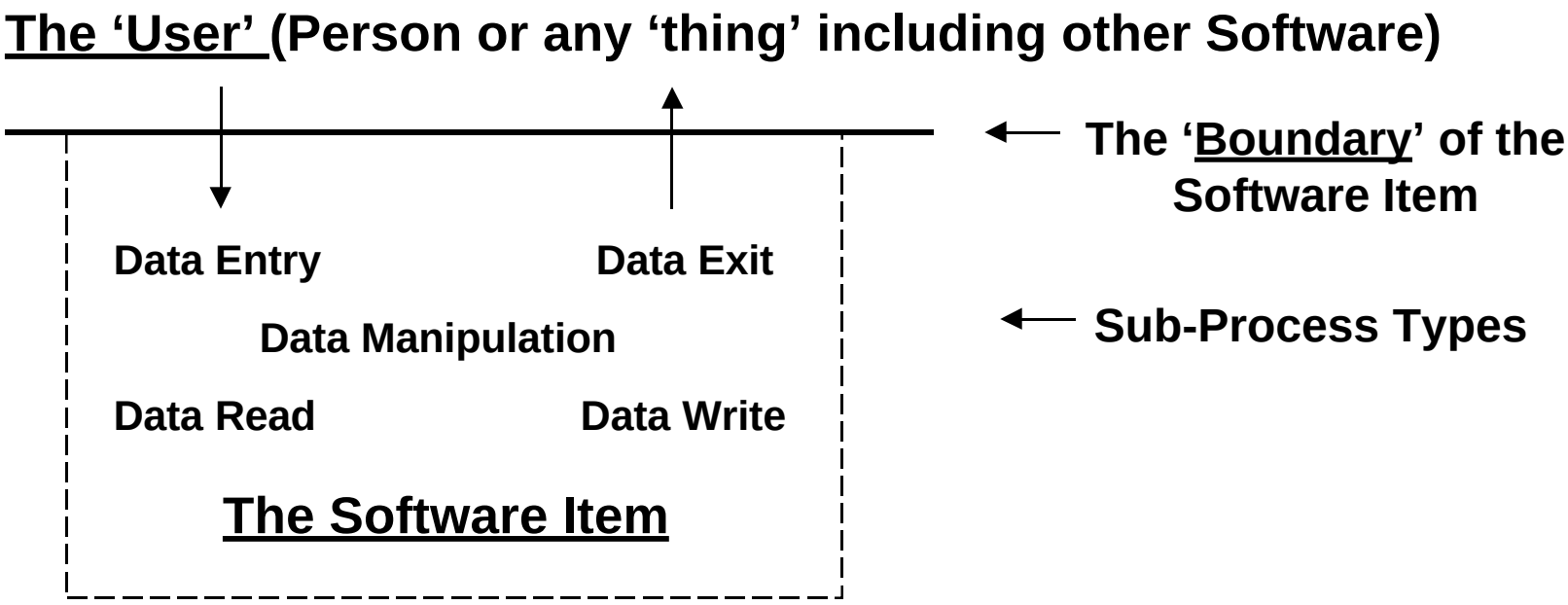
- **Existing FP methods account for Technical and Quality requirements via a 'Value Adjustment Factor' ('VAF') or suchlike. We know that this factor is not very satisfactory.**
- **If COSMIC can measure the size of components, or of changes to components, in any layer resulting from a given set of total requirements, then there should be less need for a VAF-like factor, or it should be less important**

But our priority is to measure Functional User Requirements

We propose a general model for the structure of Functional Requirements for a software item in any layer

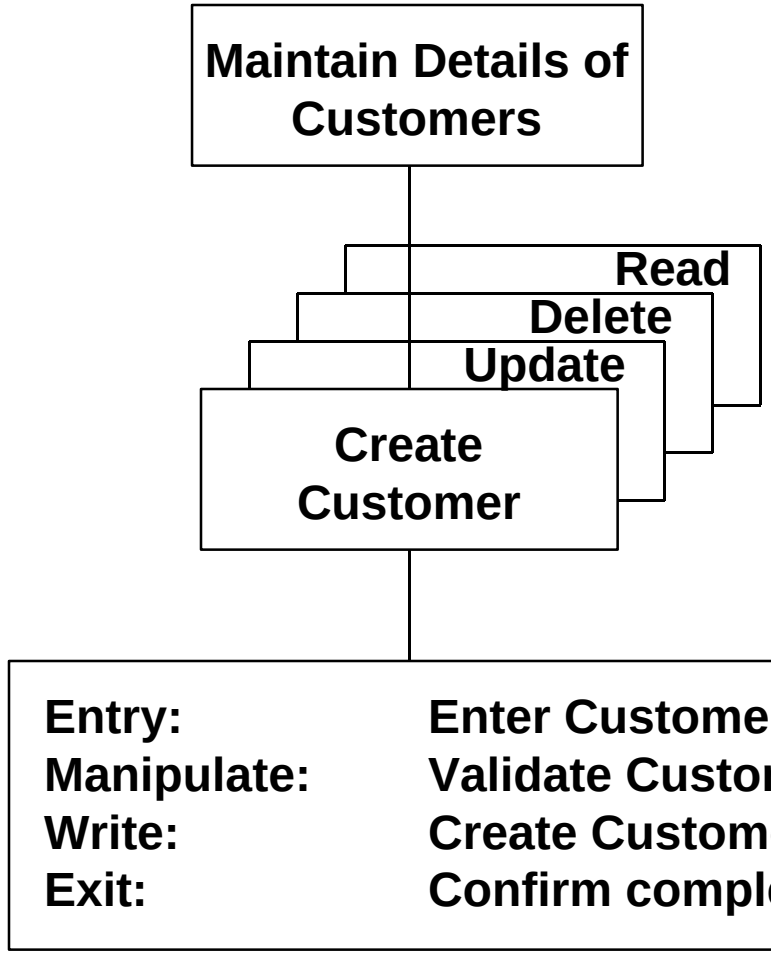


The general model also shows the interaction of the Software Item with the 'Boundary' of its Layer



Data Entries and Exits move data across the User/Software Boundary
Data Reads and Writes store and retrieve data within the software

A simple example of the general structure:

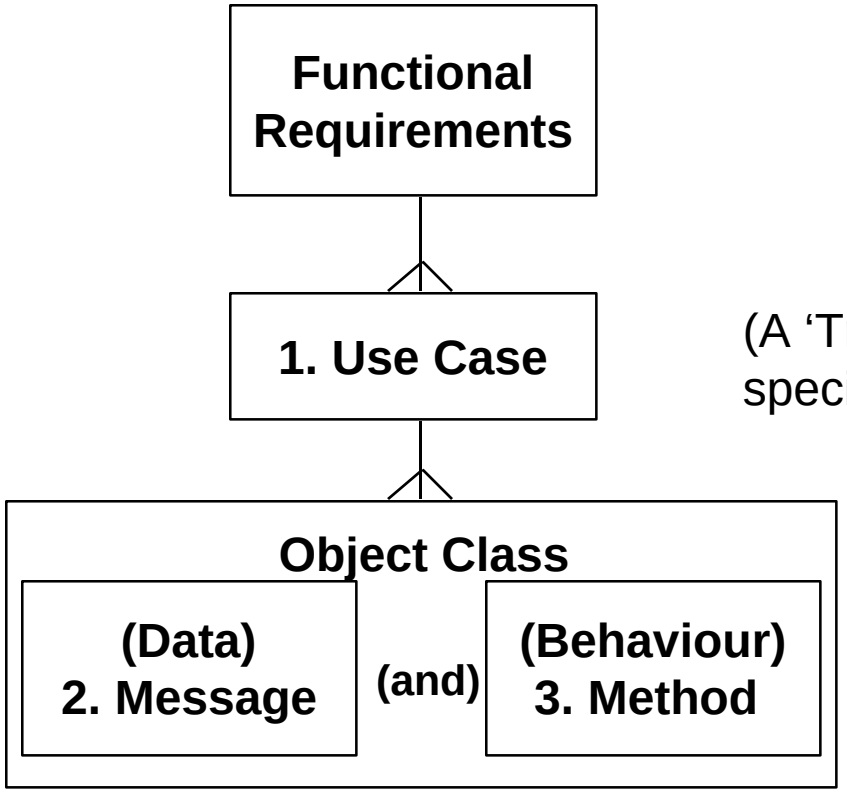


The Functional User Requirement

Four Transaction Types, each triggered by a unique Event in the real world

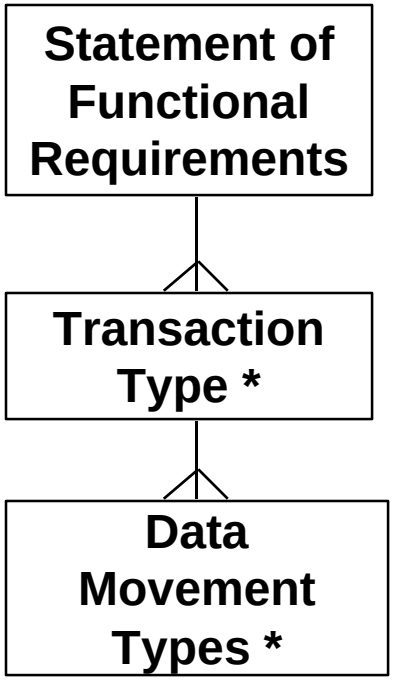
The sub-processes of the 'Create Customer' Transaction Type

(Side Observation: this model translates well to the Object-Oriented paradigm)



(A ‘Transaction Type is a very specific type of ‘Use Case’)

In practice, however, we have to simplify the ideal general model, at least initially



The ‘Data Movement Type’ (Entry, Exit, Read or Write) is now defined as:

‘a sub-process of moving a logically related block of data, AND

the data manipulation associated with the data in the block’

* the COSMIC ‘Base Functional Component Types’ or ‘BFC Types’

(Side observation: the COSMIC BFC Types map well to the BFC Types of existing 'Function Point' methods)

COSMIC	IFPUG	MkII FPA	FFP V1.0
Transaction Type	Transaction Type (EI, EO, EQ)	Logical Transaction	Transaction Type
Data Movement Types			
Entry/Exit	I/O parts of EI, EO and EQ	Input/Output	ECE/ECX
Read/Write	References to ILF's, EIF's	Entity Reference	ICR/ICW

COSMIC's difference will be in the rules for sizing the BFC Types

A set of software requirements has many possible sizes; COSMIC aims to explore four

Type of Size	Means of assigning Size Units to BFC Types	Possible Uses
Functional Size	Expert view of 'functionality'	('Functional') Productivity measurement Functional requirements control
Total Requirements Size	Expert view of all requirements	('Total') Productivity measurement Total requirements control
Standard Effort Size (1)	'Standard effort', where the standard is measured as the average over many types of technology, development environments, etc, and is therefore independent of any specific technology, etc.	Relative Productivity measurement Project size control
Standard Effort Size (2)	'Standard effort' measured for a specific technology and development environment	Estimating for a specific technology and development environment Project size control

We choose to measure Functional Size at the level of the Data Movement Type

- **Transaction Types can be unambiguously defined and identified. We need to go one level down to measure the size of their components**
- **Rules can be defined for identifying the logical Data Movement Types (Entries, Exits, Reads, Writes) unambiguously in Functional User Requirements**
- **We can propose possible Functional Size measures for the Data Movement Types**

We are studying whether our convention should be to allocate a Functional Unit Size of 'one' to a single 'DMT' or to a single 'DET'

Functional Size of a Data Movement Type = No. of Data Element Types ?

(or) Functional Size of a Transaction Type = No. of Data Movement Types ?

Factors to Consider:

- **Convertibility to existing FP methods**
- **Required precision and granularity may differ by Domain, and time in the software life-cycle**
- **The effort to measure Functional Size manually**

The granularity of Data Element Types also needs to be defined

Granularity may be Domain-specific, e.g.

- **In the MIS world a DET can be a single human-interpretable character or field, or a block of DET's treated by application software as a single unit (a 'logical record')**
- **In the real-time systems world, a DET can be a single bit, a byte, a bit string, a 'physical record' etc**

COSMIC design principles: summary

- **COSMIC's basic design principles are largely worked out and agreed by the Core Team**
- **The COSMIC method will achieve a number of "firsts" - the first Functional Sizing method to:**
  - **be designed by an international group of experts on a sound theoretical basis**
  - **draw on the practical experience of all the main existing FP methods**
  - **be designed to conform to ISO 14143 Part 1**
  - **be designed to work across MIS and real-time domains, for software in any layer**

Aims of this presentation

- To introduce the COSMIC project, its aims, structure and status
- To describe the agreed main basic principles of the family of new software functional sizing methods
- To invite interested organisations to participate in the next stage of field-testing of the COSMIC principles

The next phase will involve extensive field-testing of the design principles and detailed counting rules

The COSMIC Team is seeking software producers or users in the MIS and real-time domains who

- **can appoint a software measurement expert to review the applicability of the COSMIC approach in their organisation**
- **can provide example software requirements to help test and calibrate the COSMIC methods with support from a COSMIC Team member**
- **can contribute financially to the project costs**

Early participants in the COSMIC field-testing should gain significant benefits

- **Early access to the new ideas and opportunity to influence them**
- **Measurement of own pilot projects with the new method:**
  - **measurement data**
  - **confidential report of corporate measurements**
  - **mentoring during pilot projects & results analysis**
- **Pre-release technology transfer during pilot projects**

Organisations wishing to participate in the field-testing phase should contact any COSMIC Core Team member

Principle contact addresses:

Web-site: www.cosmicon.com

Alain Abran: abran.alain@uqam.ca

Charles Symons: charles_symons@compuserve.com