Visual fsm

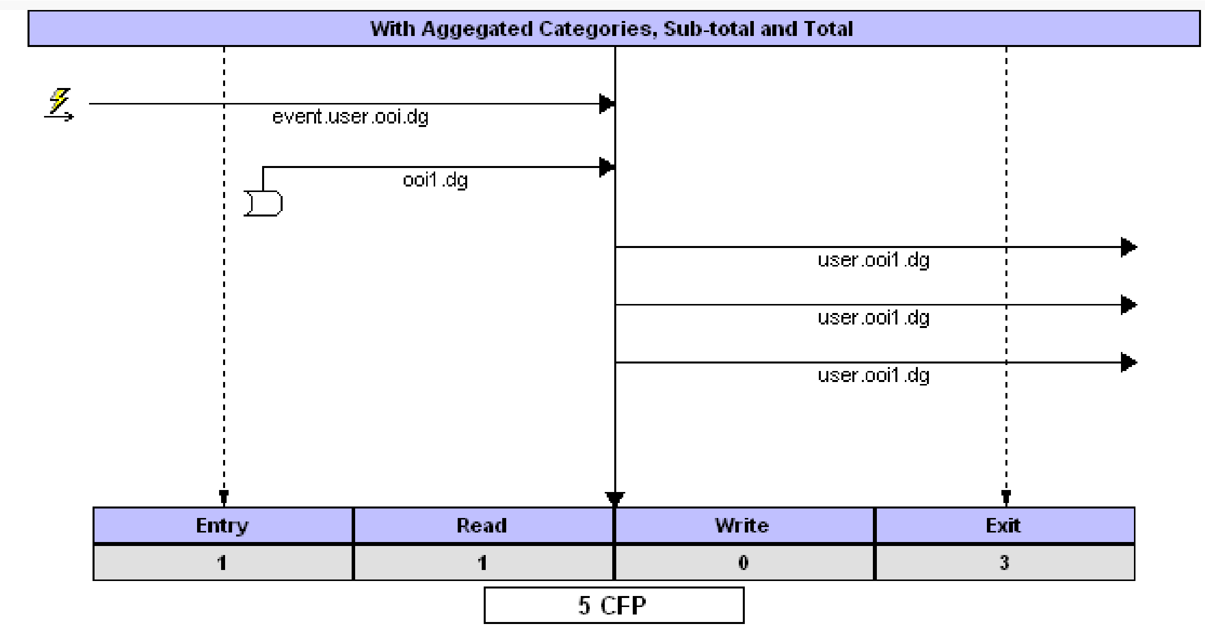
fsmPatterns

Overview

1 Introduction

COSMIC measures functional size by counting the number of movements of data to and from persistent storage, and movements of data to and from the users of a given function (Users in the COSMIC sense includes humans, devices, sensors, other system etc ie, anyone or anything that communicates with the software) the aggregate of all the individual functions delivered by the software being its total size

To depict these movements Pentad devised the Data Movement Sequence Diagram (DMSD), a variation of the UML sequence Diagram which shows in a diagrammatic form the movements of data for each individual function delivered by the software as described above, an example is shown below



Entry	Read	Write	Exit
1	1	0	3

5 CFP

Counting these movements by careful analysis of the Functional User Requirements (FUR) for a given function is relatively easy, however it takes time and on occasions an estimate of the size would satisfy the purpose of measuring the size. There are various ways in which this can be achieved and can be researched by referring to published literature on the subject.

However from experience gathered measuring many software systems, Pentad found that the pattern of movements for many different functions take the same diagrammatic form. It was possible therefore to design a set of generic patterns that were common in a given software domain. Pentad has devised such sets and it calls them fsmPatterns©

The advantage of fsmPatterns is that its size can be determined in advance, This means that the Functional Size Analyst need only identify each individual function then assign a pattern to them. In this way it is possible easily to derive an estimate of the COSMIC functional size of a set of functions,

The VisualFSM software analysis and measurement tool is designed to measure both the actual functional size, and for COSMIC, fsmPatterns can be imported from standard libraries or designed from scratch. When an fsmPattern is assigned to a Functional Process, it adopts the size of the Pattern; this is referred to in VisualFSM as the Pattern Size to distinguish it from the Measured Size.

If subsequently a Functional Process is measured, the Measured Size is reported, but the fsmPattern size is retained, which is useful for analyzing the variance of the fsmPattern size from the Measured Size.

The purpose of this guide is to describe the fsmPattern Design Module, then to demonstrate the use of fsmPatterns to derive the Pattern size of the software, or a hybrid of both Measured and fsmPattern Sizes.

2 fsmPattern Libraries

fsmPattern Libraries are a means of categorizing fsmPatterns according to their application.

The VisualFSM analysis and measurement tool has a categorization scheme for Software Applications and Systems, fsmPatterns are categorized using the same scheme, meaning they can be implemented directly by VisualFSM.

There are 3 levels of categorization, The Domain and Sub-domain of the software and the Type of Pattern applicable to the sub-domain

Domain and Sub-domain

A Domain groups together software systems with similar applicability and characteristics such as business applications which are largely data rich, real time application which are largely message rich with time and response constraints. A Sub-domain is a further level of categorization within a Domain

fsmPattern Type

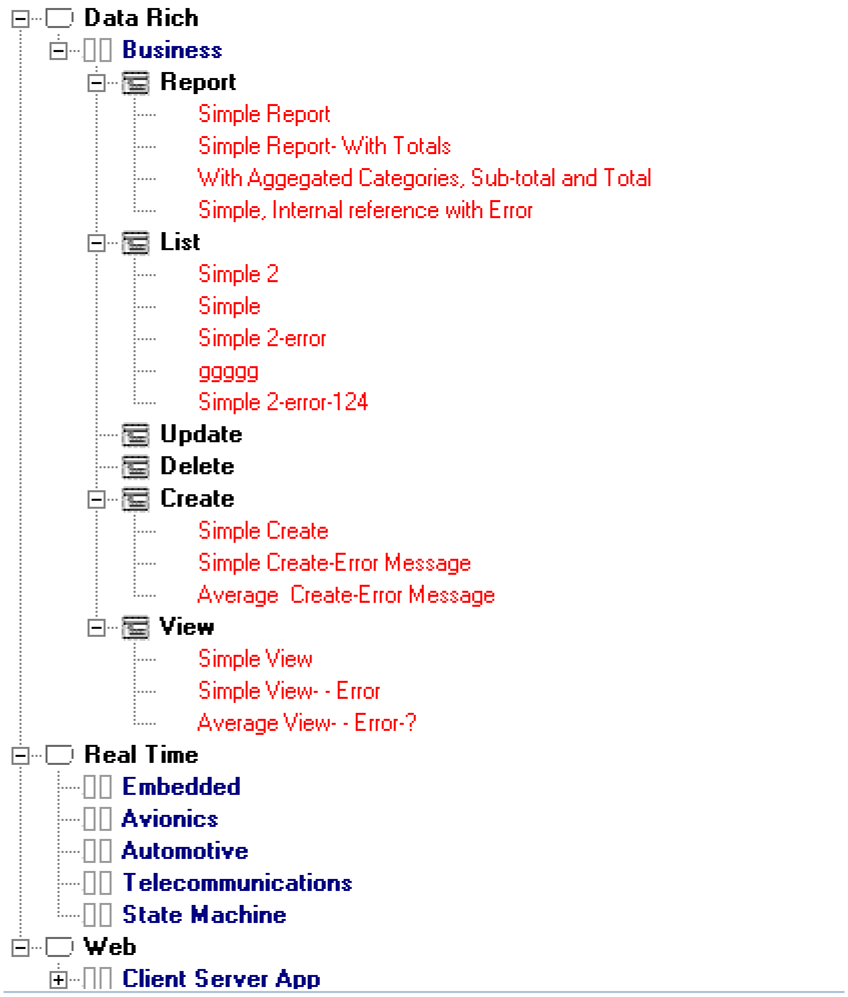
In a particular Domain and Sub-domain there are groups of patterns that fulfill a similar purpose but vary in the amount of data handled. For example in the Data Rich domain reporting is an important function and there are are many fsmPatterns that are concerned only with the different types of report. These are called fsmPattern Types.

fsmPattern

a Pattern is a set of Data Movement represented as a Data Movement Sequence Diagram. VisualFSM Includes a function to assign an fsmPattern to a Functional Process. The Functional Process adopts the size of the fsmPattern

Selection and Navigation

As the categorization is a hierarchical structure VisualFSM uses a tree to represent the structure as shown below.



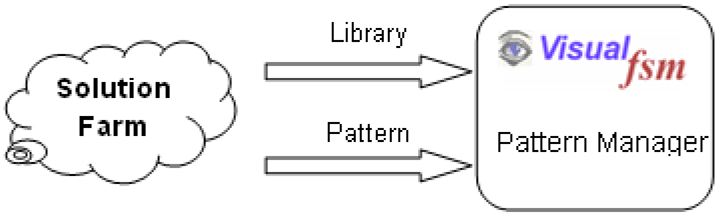
- Data Rich
  - Business
    - Report
      - Simple Report
      - Simple Report- With Totals
      - With Aggegated Categories, Sub-total and Total
      - Simple, Internal reference with Error
    - List
      - Simple 2
      - Simple
      - Simple 2-error
      - ggggg
      - Simple 2-error-124
    - Update
    - Delete
    - Create
      - Simple Create
      - Simple Create-Error Message
      - Average Create-Error Message
    - View
      - Simple View
      - Simple View- - Error
      - Average View- - Error-?
- Real Time
  - Embedded
  - Avionics
  - Automotive
  - Telecommunications
  - State Machine
- Web
  - Client Server App

3 fsmPattern Management

The VisualFSM Pattern Manager is use the manage libraries and patterns. Pattern libraries and Patterns can of course be built for private use, and how this is done is explained later. However one of the aims of VisualFSM is to be able to share information with other users. This sharing of information can be performed in a variety of ways, all controlled directly from within the VisualFSM Pattern Manager

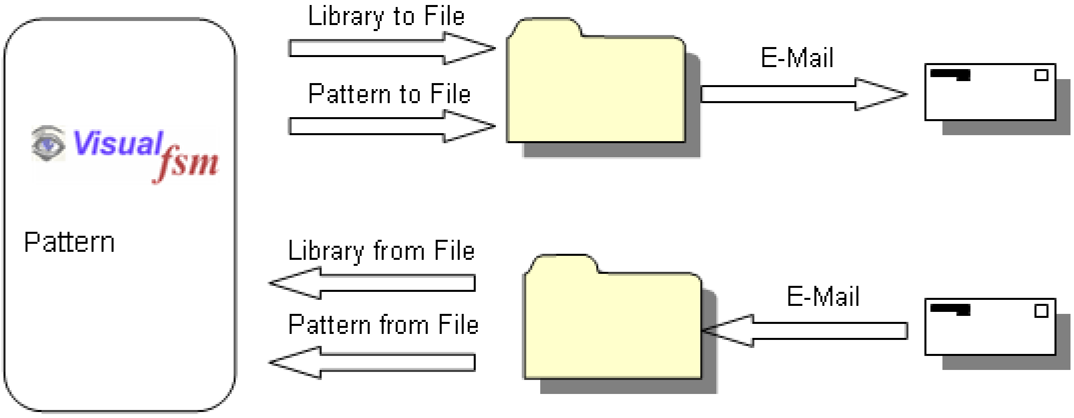
Download from the VisualFSM Solution Farm

Libraries and individual Patterns can be downloaded as an install-able file from the VisualFSM Solution Farm



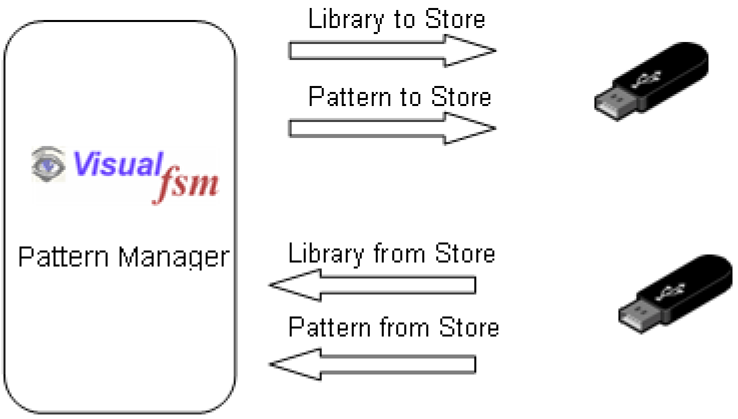
Send and receive via e-mail

fsmPatterns can be exported to a file then transferred either by e-Mail or physical media. The file can then be imported and added to the library



Send and receive via physical media

fsmPatterns can be exported to a file which can be transferred either by e-Mail or physical media.The file can then be imported and added to the library.



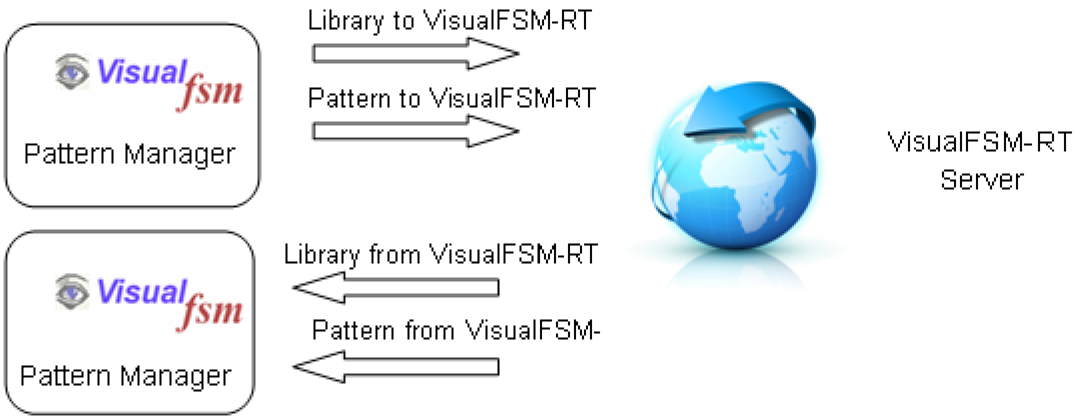
Realtime collaboration

VisualFSM offers a subscription service called VisualFSM-RT which provides for realtime collaboration between users subscribed to the service over the Internet. The messages exchanged are encoded then sent over the internet via the VisualFSM-RT Server.
Note that VisualFSM-RT is only a broker and cannot decode any message sent to it, it simply routes it to the intended recipients.

Subscribed users can:

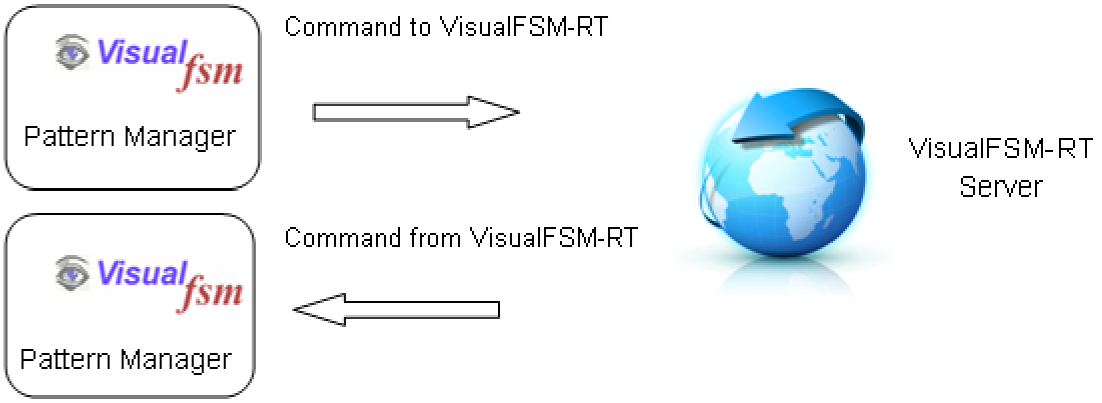
Exchange patterns directly

The fsmPattern Manager encodes the library or pattern then sends it via VisualFSM-RT to the other user where the process is reversed.



Collaboration

When a connection is established between two subscribed users, their fsmPattern Managers are synchronized. Any selection or navigation in the fsmPattern Manager by one user is immediately repeated by the other user's fsmPattern Manager; similarly if one user changes a pattern the other user's pattern will be updated immediately.



***Notes**:*

In the Community and Personal Editions the collaboration is peer-to-peer and the messages are encoded providing a basic level of security.

In the Professional, Team and Agile versions collaboration is performed between several team members, controlled by a team leader and the messages are encrypted using RSA Public and Private keys for better security

4 fsmPattern Exploitation

Introduction

One of the main uses of the Functional Size metric is to use it with other metrics to derive indicators which enable software developers to estimate development time and effort. The drawback is that measuring functional size itself takes time and effort and requires the acquisition of additional skills not needed for the main development activity. This time and effort and potential training costs must be added to the overall cost.

Software developers need a way to use their existing prediction method but use a reliable estimate of Functional Size instead of a size measured by direct measurement. If the result is derived from an estimated size and is fit for purpose then it would help control the development process.

The VisualFSM tool analyses and measure functional size using IFPUG, Mk1FPA, COSMIC and SiFPA. The SiFPA method is relatively new and is very easy to apply but its claim that it yields the same result as measuring using the IFPUG Method still needs more independent research to test that claim. If the claim proves to be true it could be a candidate for the estimated Functional Size.

Other methods have techniques to provide estimate of Functional Size, they too could be candidates but that is outside the scope of this document and the reader in encouraged to research these techniques. They could be useful for use in the early stages of the development life-cycle.

The purpose of this chapter is to outline how fsmPatterns can be used to provide an estimated Functional Size metric using artefacts that come directly from the development process and which requires only the normal skills of a trained Analyst.

Estimation Process

The process assumes that a library for the domain of the software has been built or acquired, that value for Productivity in terms of Hours per Function Point is known and that the Functional User Requirements are detailed enough to be able to extract the functions performed by the software.

There are 6 main tasks in the process

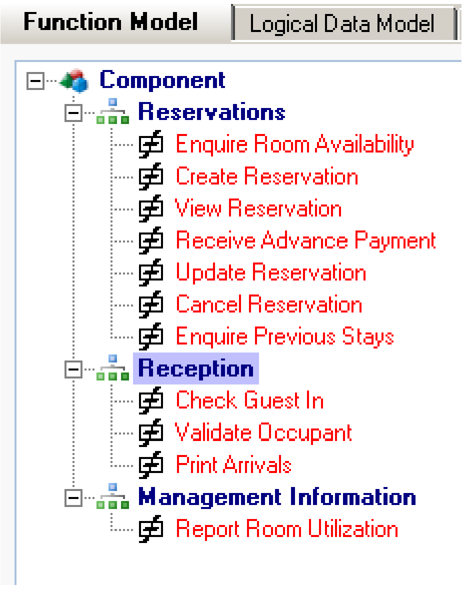
1. Extract the Function Model from the Functional user Requirements
2. Create an Analysis Project and set the scope
3. Create a COSMIC Analysis project
4. Assign an fsmPattern to each Functional process
5. Produce the fsmPattern Size Report
6. Use the fsmPattern Size in the productivity equation to derive estimated effort

Preparation

Refer to the ***VisualFSM COSMIC Quickstart Guide*** , load the Hotel Management Case Study, and follow the instructions to perform the tasks described below

4.1 Task 1 - Extract the Function Model

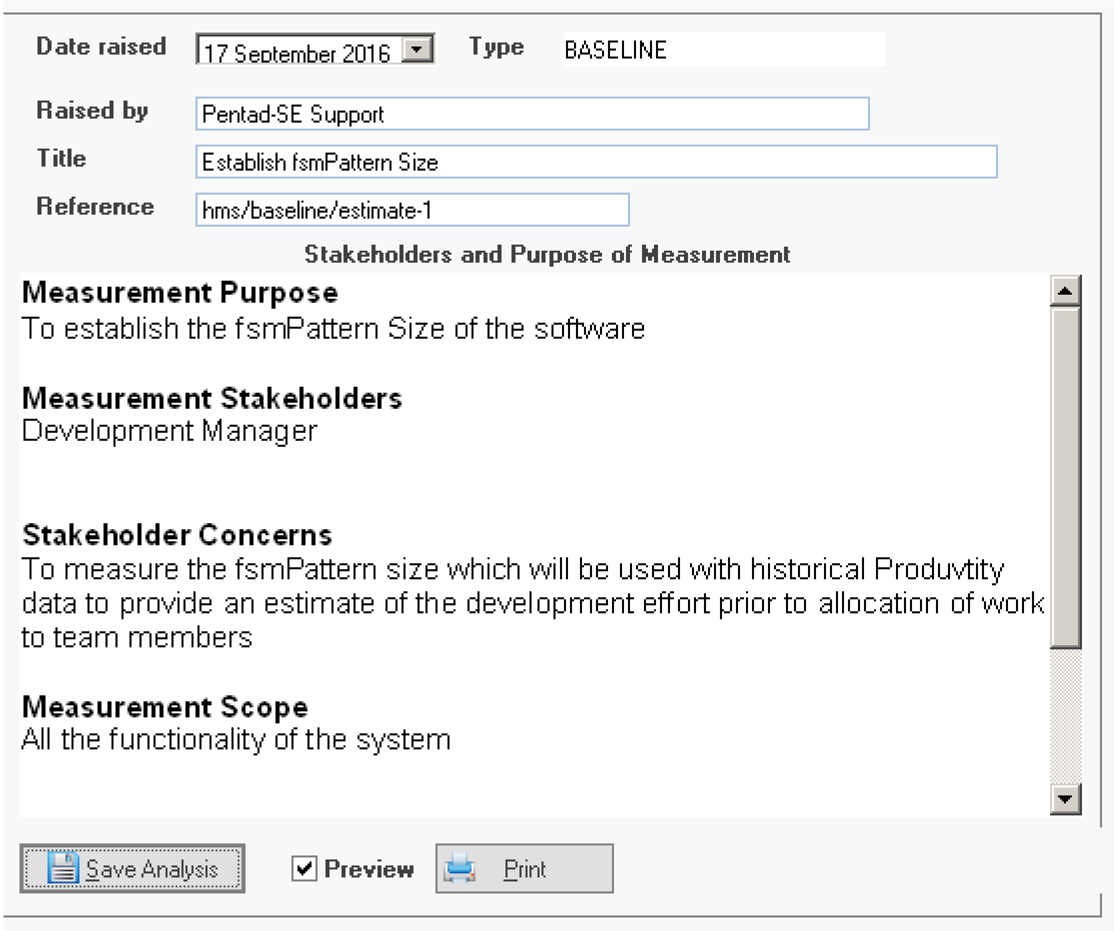
Build the Function Model. This is a hierarchical structure presented in the form of a tree and looks like this when completed



- Component
  - Reservations
    - Enquire Room Availability
    - Create Reservation
    - View Reservation
    - Receive Advance Payment
    - Update Reservation
    - Cancel Reservation
    - Enquire Previous Stays
  - Reception
    - Check Guest In
    - Validate Occupant
    - Print Arrivals
  - Management Information
    - Report Room Utilization

4.2 Task 2 - Create an Analysis Project

Create an Analysis Project as follows:



Date raised	17 September 2016	Type	BASELINE
Raised by	Pentad-SE Support		
Title	Establish fsmPattern Size		
Reference	hms/baseline/estimate-1		

Stakeholders and Purpose of Measurement

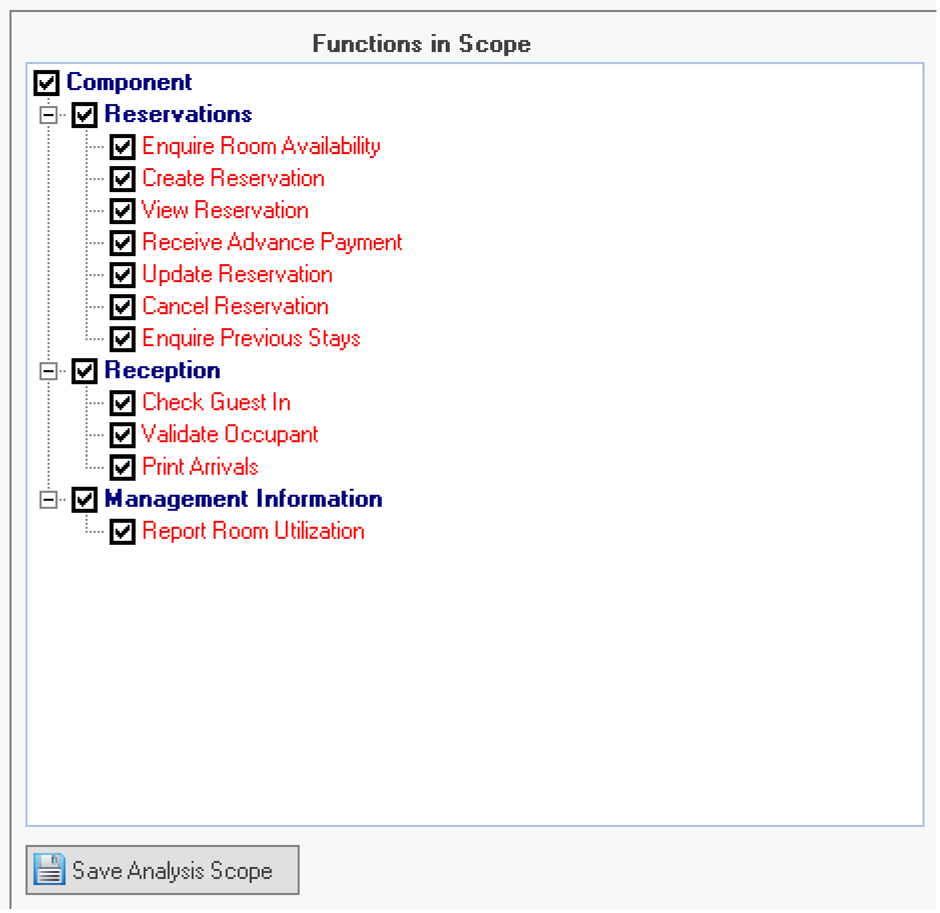
Measurement Purpose
To establish the fsmPattern Size of the software

Measurement Stakeholders
Development Manager

Stakeholder Concerns
To measure the fsmPattern size which will be used with historical Produvtity data to provide an estimate of the development effort prior to allocation of work to team members

Measurement Scope
All the functionality of the system

Set the Scope to include all functions by ticking the boxes and saving:

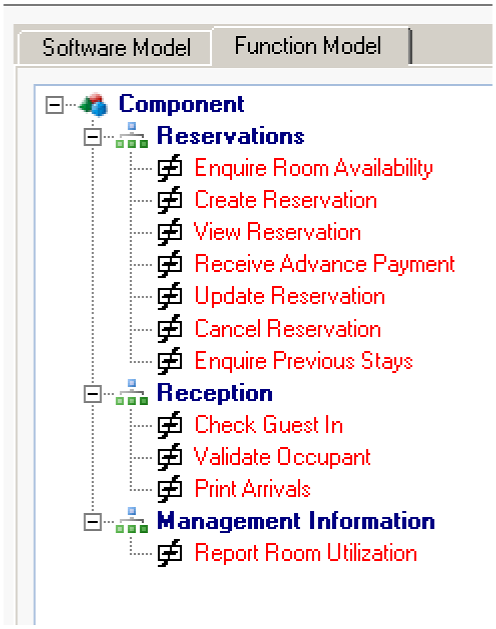


Functions in Scope

- ☑ Component
  - ☑ Reservations
    - ☑ Enquire Room Availability
    - ☑ Create Reservation
    - ☑ View Reservation
    - ☑ Receive Advance Payment
    - ☑ Update Reservation
    - ☑ Cancel Reservation
    - ☑ Enquire Previous Stays
  - ☑ Reception
    - ☑ Check Guest In
    - ☑ Validate Occupant
    - ☑ Print Arrivals
  - ☑ Management Information
    - ☑ Report Room Utilization

4.3 Task 3 - Create a COSMIC Analysis Project

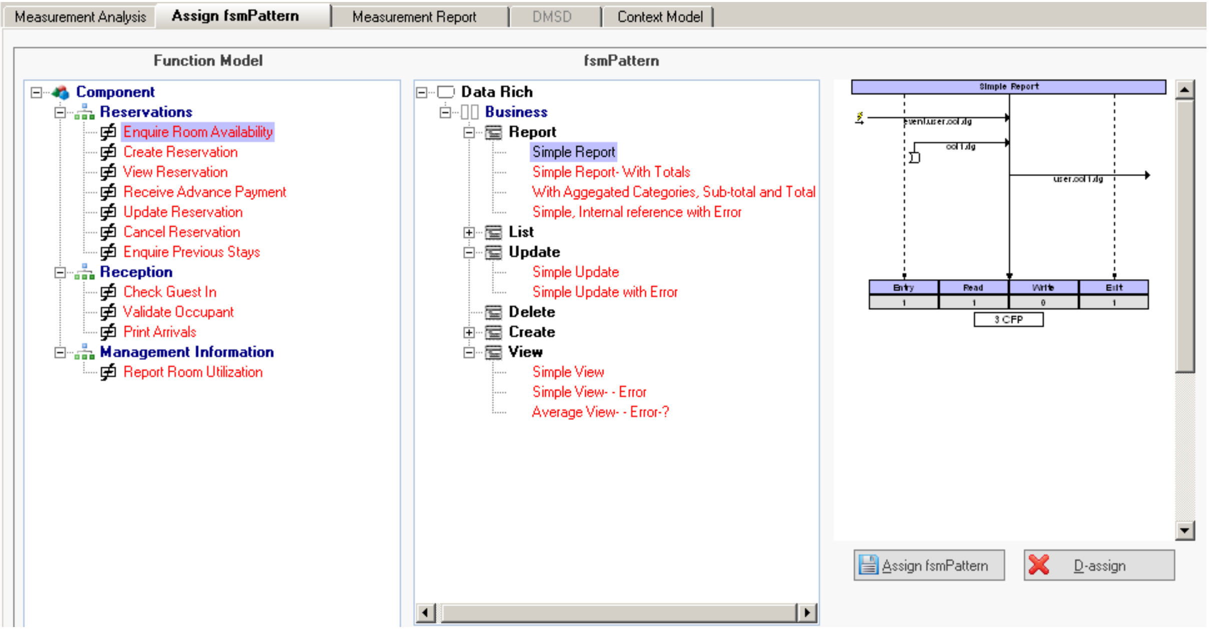
Create a COSMIC Analysis project, VisualFSM will do most of the work for you, automatically converting the generic Function Model to a set of Functional Processes.



4.4 Task 4 - Apply the fsmPatterns

The COSMIC analysis screen enabled the user to perform a detailed analysis to derive an actual COSMIC Functional Size, however as stated in the purpose, the aim is to provide an quick estimate, select the fsmPattern Tab

Select each Function in turn and assign a fsmPattern. When an fsmPattern is selected a thumbnail is displayed for confirmation. When assigned the function automatically adopts the pattern size,



Function Model

- Component
  - Reservations
    - Enquire Room Availability
    - Create Reservation
    - View Reservation
    - Receive Advance Payment
    - Update Reservation
    - Cancel Reservation
    - Enquire Previous Stays
  - Reception
    - Check Guest In
    - Validate Occupant
    - Print Arrivals
  - Management Information
    - Report Room Utilization

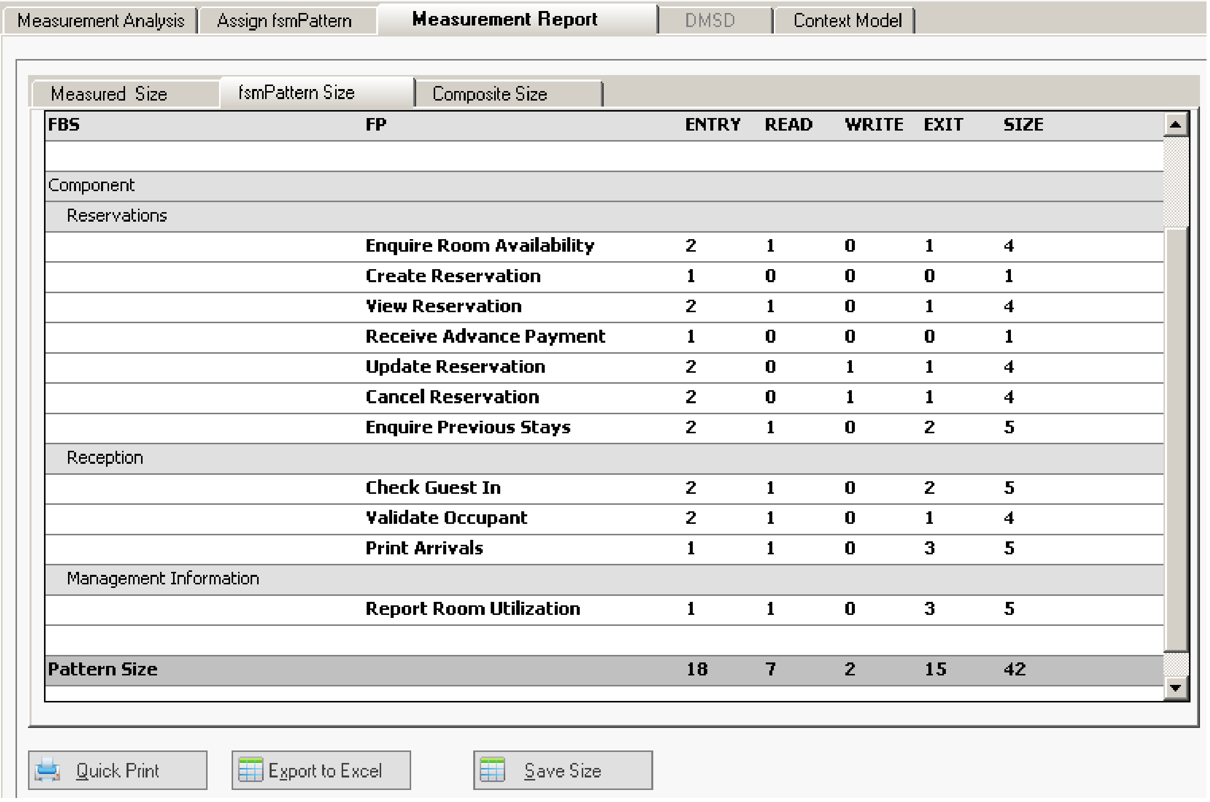
fsmPattern

- Data Rich
  - Business
    - Report
      - Simple Report
      - Simple Report- With Totals
      - With Aggegated Categories, Sub-total and Total
      - Simple, Internal reference with Error
    - List
    - Update
      - Simple Update
      - Simple Update with Error
    - Delete
    - Create
    - View
      - Simple View
      - Simple View- - Error
      - Average View- - Error-?

Entry	Read	Write	Exit
1	1	0	1

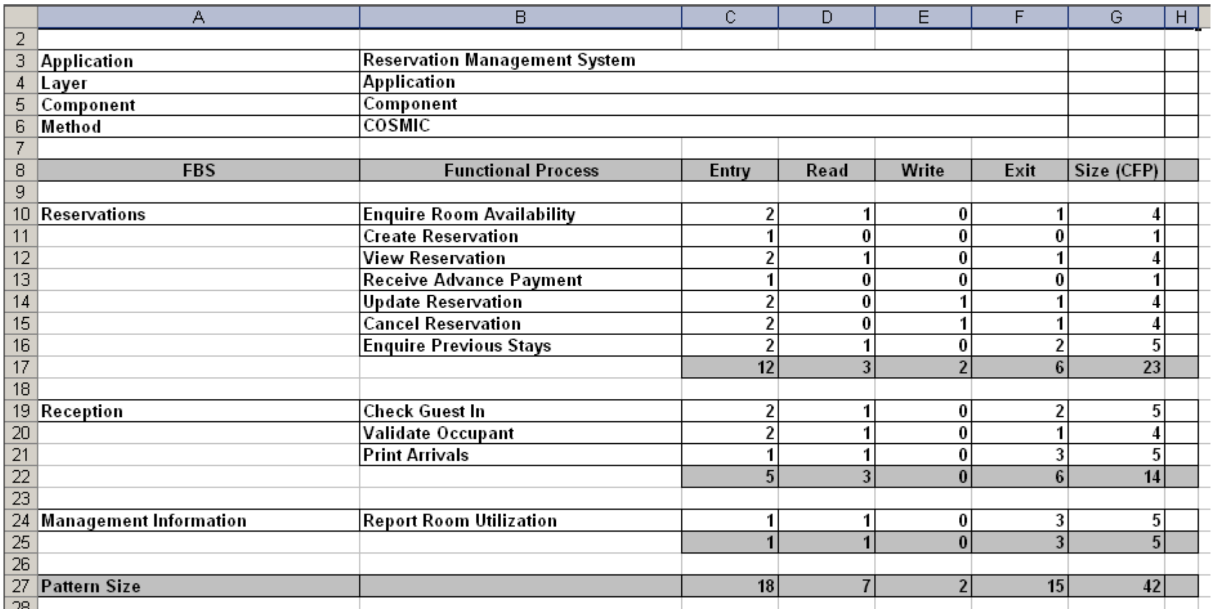
4.5 Task 5 - Produce the fsmPattern Size Report

Produce the fsmPattern Size Report



FBS	FP	ENTRY	READ	WRITE	EXIT	SIZE
Component						
Reservations						
	Enquire Room Availability	2	1	0	1	4
	Create Reservation	1	0	0	0	1
	View Reservation	2	1	0	1	4
	Receive Advance Payment	1	0	0	0	1
	Update Reservation	2	0	1	1	4
	Cancel Reservation	2	0	1	1	4
	Enquire Previous Stays	2	1	0	2	5
Reception						
	Check Guest In	2	1	0	2	5
	Validate Occupant	2	1	0	1	4
	Print Arrivals	1	1	0	3	5
Management Information						
	Report Room Utilization	1	1	0	3	5
Pattern Size		18	7	2	15	42

Export the report to an Excel Spreadsheet



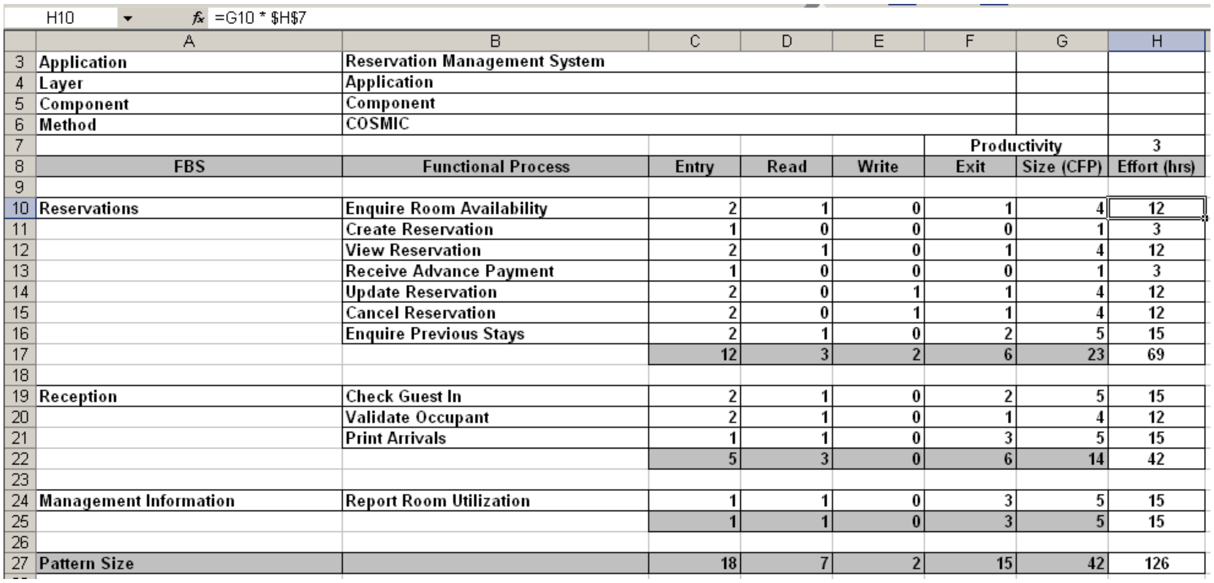
	A	B	C	D	E	F	G	H
2								
3	Application	Reservation Management System						
4	Layer	Application						
5	Component	Component						
6	Method	COSMIC						
7								
8	FBS	Functional Process	Entry	Read	Write	Exit	Size (CFP)	
9								
10	Reservations	Enquire Room Availability	2	1	0	1	4	
11		Create Reservation	1	0	0	0	1	
12		View Reservation	2	1	0	1	4	
13		Receive Advance Payment	1	0	0	0	1	
14		Update Reservation	2	0	1	1	4	
15		Cancel Reservation	2	0	1	1	4	
16		Enquire Previous Stays	2	1	0	2	5	
17			12	3	2	6	23	
18								
19	Reception	Check Guest In	2	1	0	2	5	
20		Validate Occupant	2	1	0	1	4	
21		Print Arrivals	1	1	0	3	5	
22			5	3	0	6	14	
23								
24	Management Information	Report Room Utilization	1	1	0	3	5	
25			1	1	0	3	5	
26								
27	Pattern Size		18	7	2	15	42	

4.6 Task 6 - Add Effort Calculation Formula

It is assumed that the reader is familiar with constructing Excel spreadsheets and emter formalae

In column (h) for each Functional Process enter the effort calculation formula (For the purposes of illustration we shall apply a global value for Productivity if 3 Hours per CFP

Effort = size * productivity



H10 fx =G10 * H7

	A	B	C	D	E	F	G	H
3	Application	Reservation Management System						
4	Layer	Application						
5	Component	Component						
6	Method	COSMIC						
7						Productivity		3
8	FBS	Functional Process	Entry	Read	Write	Exit	Size (CFP)	Effort (hrs)
9								
10	Reservations	Enquire Room Availability	2	1	0	1	4	12
11		Create Reservation	1	0	0	0	1	3
12		View Reservation	2	1	0	1	4	12
13		Receive Advance Payment	1	0	0	0	1	3
14		Update Reservation	2	0	1	1	4	12
15		Cancel Reservation	2	0	1	1	4	12
16		Enquire Previous Stays	2	1	0	2	5	15
17			12	3	2	6	23	69
18								
19	Reception	Check Guest In	2	1	0	2	5	15
20		Validate Occupant	2	1	0	1	4	12
21		Print Arrivals	1	1	0	3	5	15
22			5	3	0	6	14	42
23								
24	Management Information	Report Room Utilization	1	1	0	3	5	15
25			1	1	0	3	5	15
26								
27	Pattern Size		18	7	2	15	42	126

Col (x) will now contain the estimated hours for each FP and you can use your Excel skills to calculate aggregates and so on

***Notes**:*

In the Community and Personal Editions the exploitation is performed outside of VisualFSM. In the Professional, Team and Agile versions the technique is embedded into VisualFSM and this task is automated

4.7 Review - Compare Pattern Size with Actual size

If required the project can be Cloned, and a detailed measure of each Functional Process can be performed. The fsmPattern Size is retained for reference.

As both the fsmPattern Size and the Measured Size are available at the Functional Process level, comparing the results may identify new fsmPatterns to be added to the library.