



# **fsmPatterns**

## **Overview**

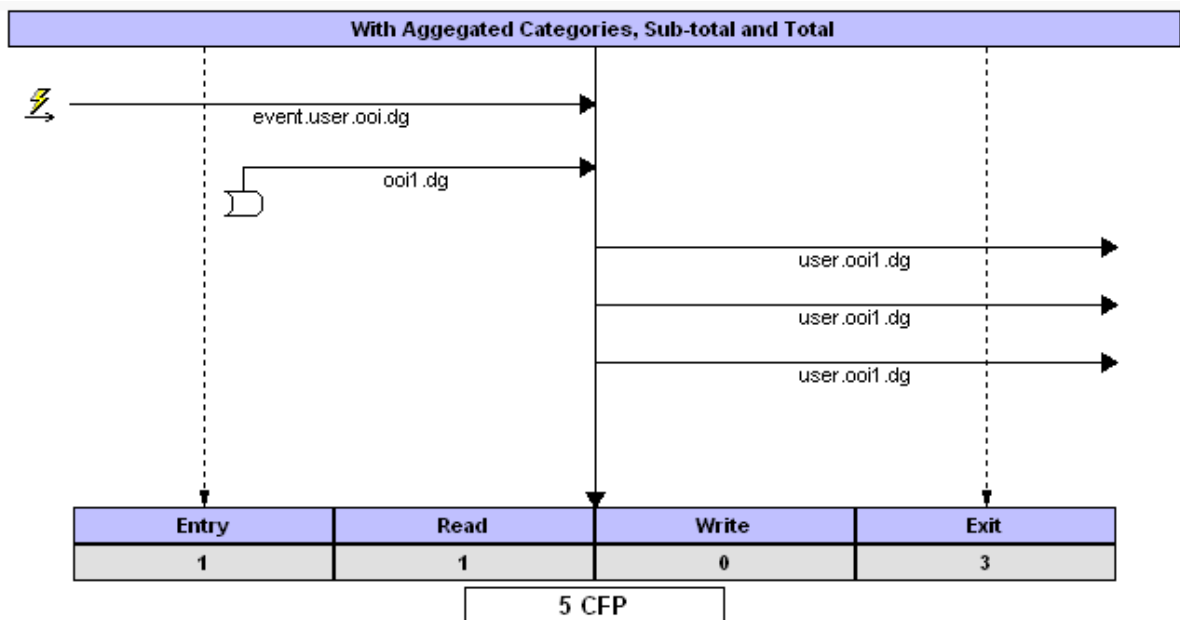
# Table of Contents

<b>Part I Introduction</b>	<b>1</b>
<b>Part II fsmPattern Libraries</b>	<b>3</b>
<b>Part III fsmPattern Type Maintenance</b>	<b>5</b>
<b>Part IV fsmPattern Maintenance</b>	<b>9</b>
<b>Part V fsmPattern Exchange</b>	<b>17</b>
1 Downloading from the Solution Farm .....	17
2 Export to VDIF File .....	20
3 Import frm VDIF File .....	22
<b>Part VI fsmPattern Exploitation</b>	<b>25</b>
<b>Index</b>	<b>0</b>

# 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



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

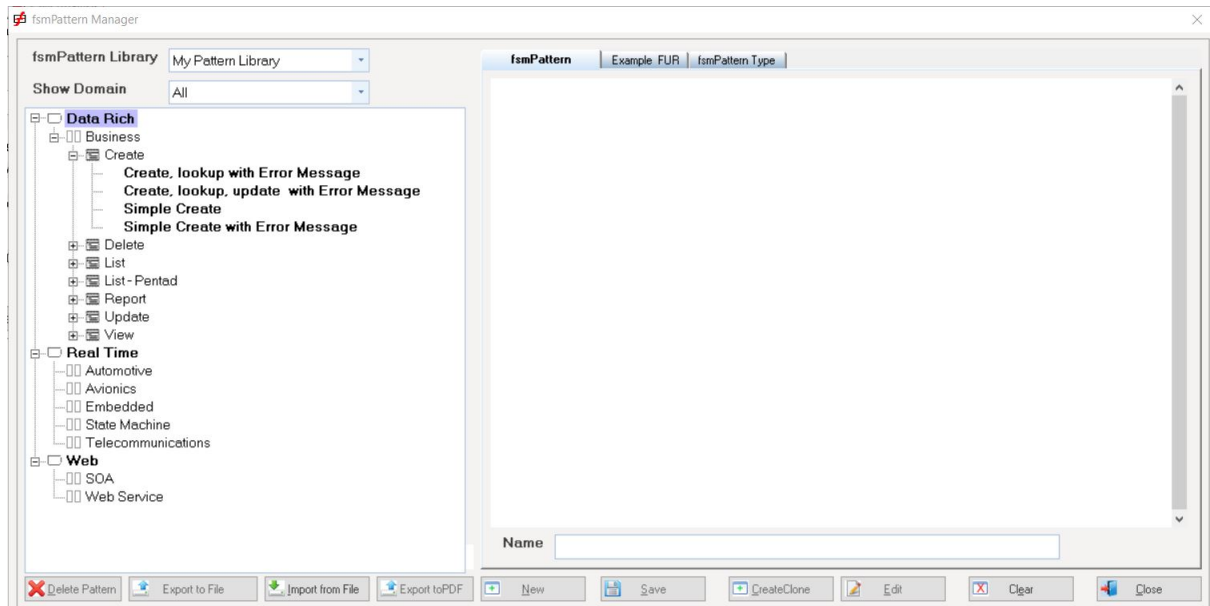
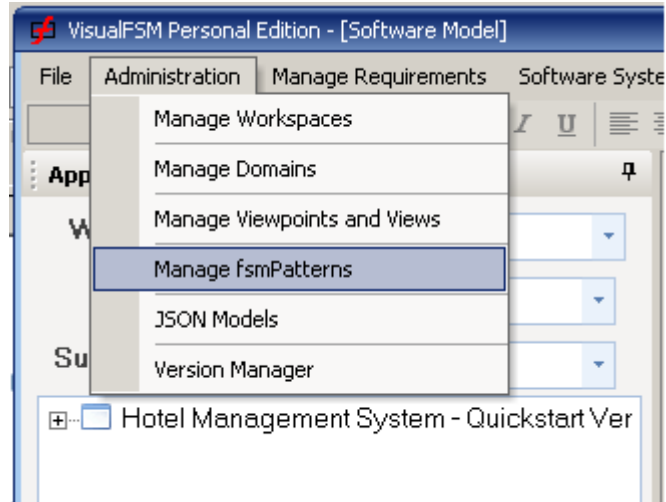
In a particular Domain and Sub-domain there are groups of patterns that fulfill a similar purpose but vary in the amount of data handled. For example in the Data Rich domain reporting is an important function and there are many fsmPatterns that are concerned only with the different types of report. These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

Load the VisualFSM Pattern Manager



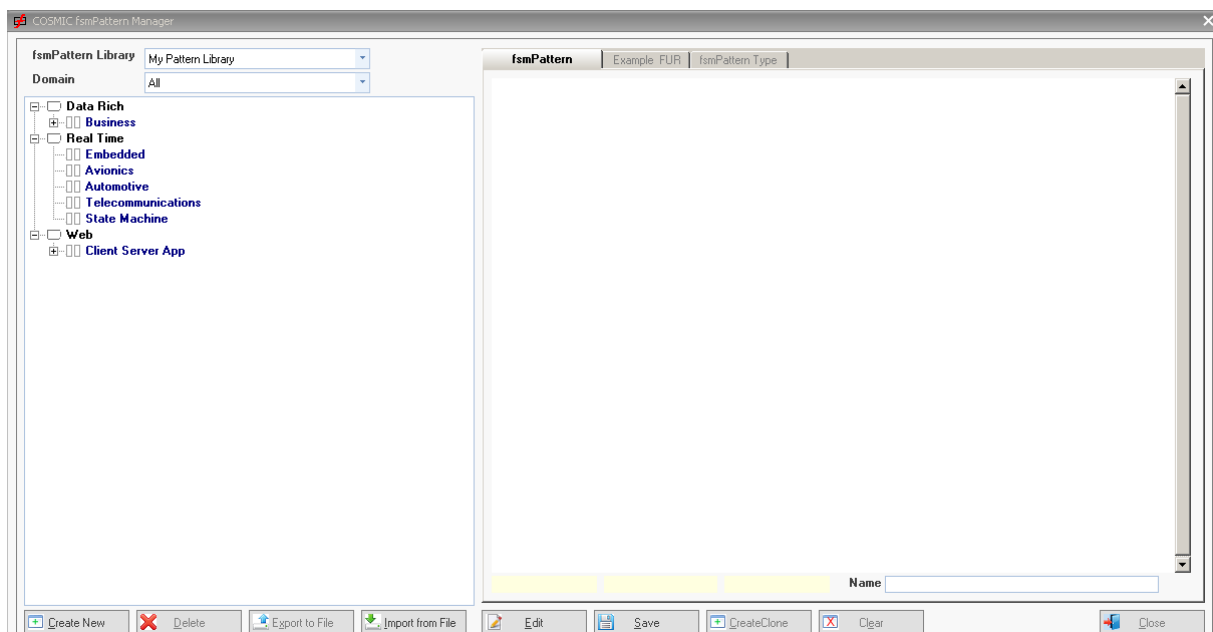
### 3 fsmPattern Type Maintenance

The Personal Edition of VisualFSM enables users to create and maintain fsmPattern Types. It is delivered with a pre-defined set of Domains as described in the previous chapter, and for each Domain a set of fsmPattern Types is defined.

Existing fsmPattern Types can be updated or new fsmPattern Types added. fsmPattern Types can also be deleted

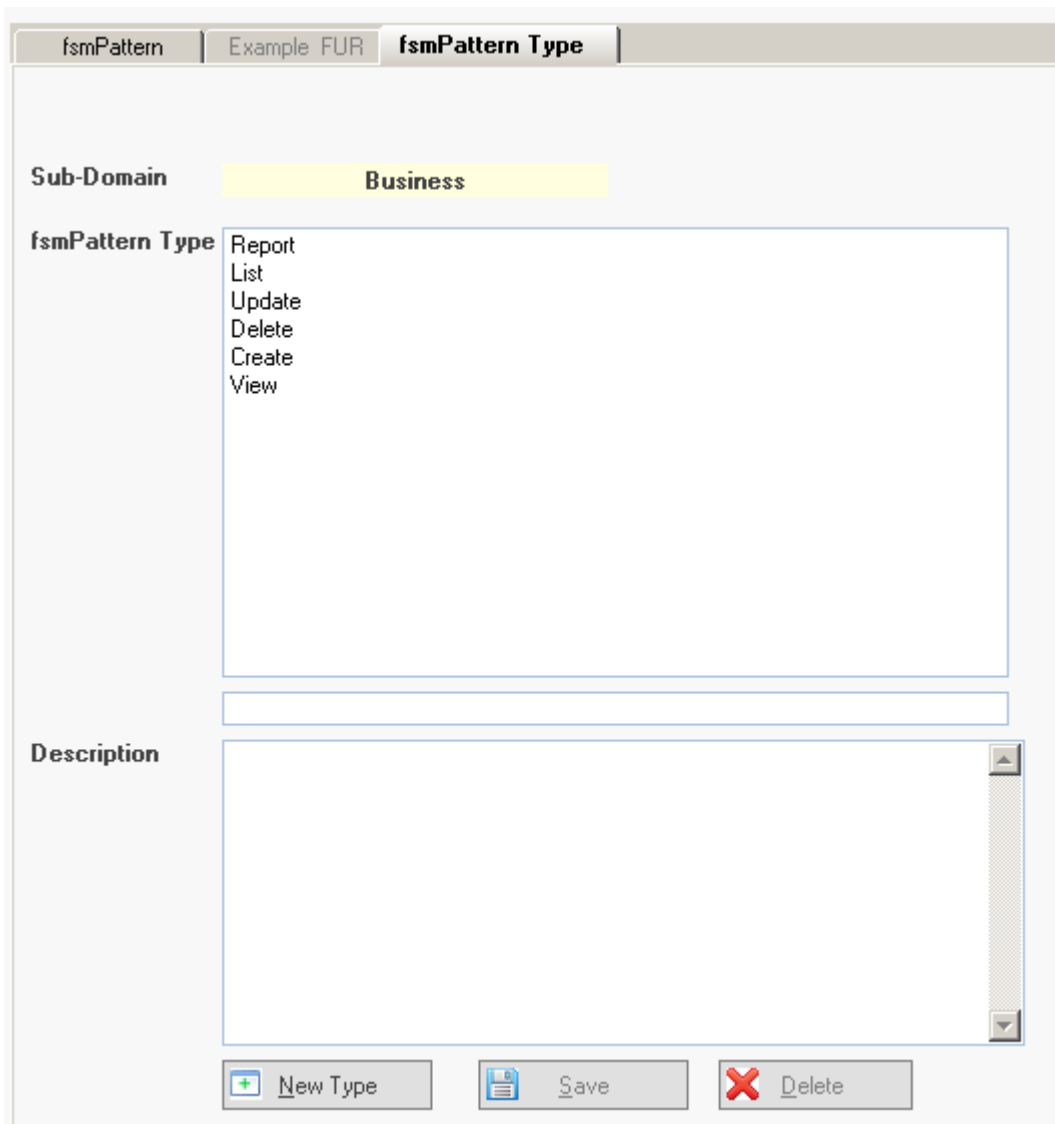
#### Create a new fsmPattern Type

Start the fsmPattern Manager by selecting **Administration** in the main menu and **Mange fsmPatterns** from the drop-down menu



The Navigator in the left panel has already been described. The right hand panel is where the fsmPattern Types and fsmPatterns for a given domain are created

For this demonstration select the **Data Rich - Business Domain** and select the **fsmPattern** Type Tab



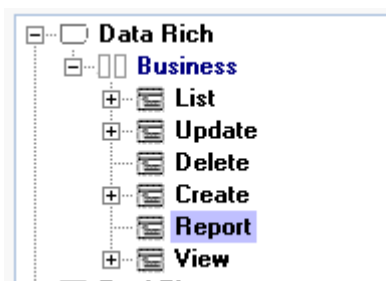
The list of existing pattern type is displayed, we need to add a new fsmPattern Type to described reports generated by the software

Press **New Type**, give the new type a name, enter a short description, then press **Save**.



The screenshot shows a software interface for maintaining fsmPattern types. At the top, there are tabs for 'fsmPattern', 'Example FUR', and 'fsmPattern Type'. The 'Sub-Domain' is set to 'Business'. A list of fsmPattern types is shown, with 'Report' selected. Below the list, a text box contains the name 'Report'. A larger text area below that contains the description 'A set of patterns describing report od differing complexities'. At the bottom, there are three buttons: 'New Type' (with a plus icon), 'Save' (with a floppy disk icon), and 'Delete' (with a red X icon).

The new type will be added to the Domain



## Update an existing fsmPattern type

To update an existing fsmPattern Type select it and the current name and description are

displayed and can be edited and saved

## **Delete an existing fsmPattern type**

To delete an existing fsmPattern Type select it and the current name and description are displayed. Press the Delete button. If confirmed it will be permanently deleted and remove from the navigator

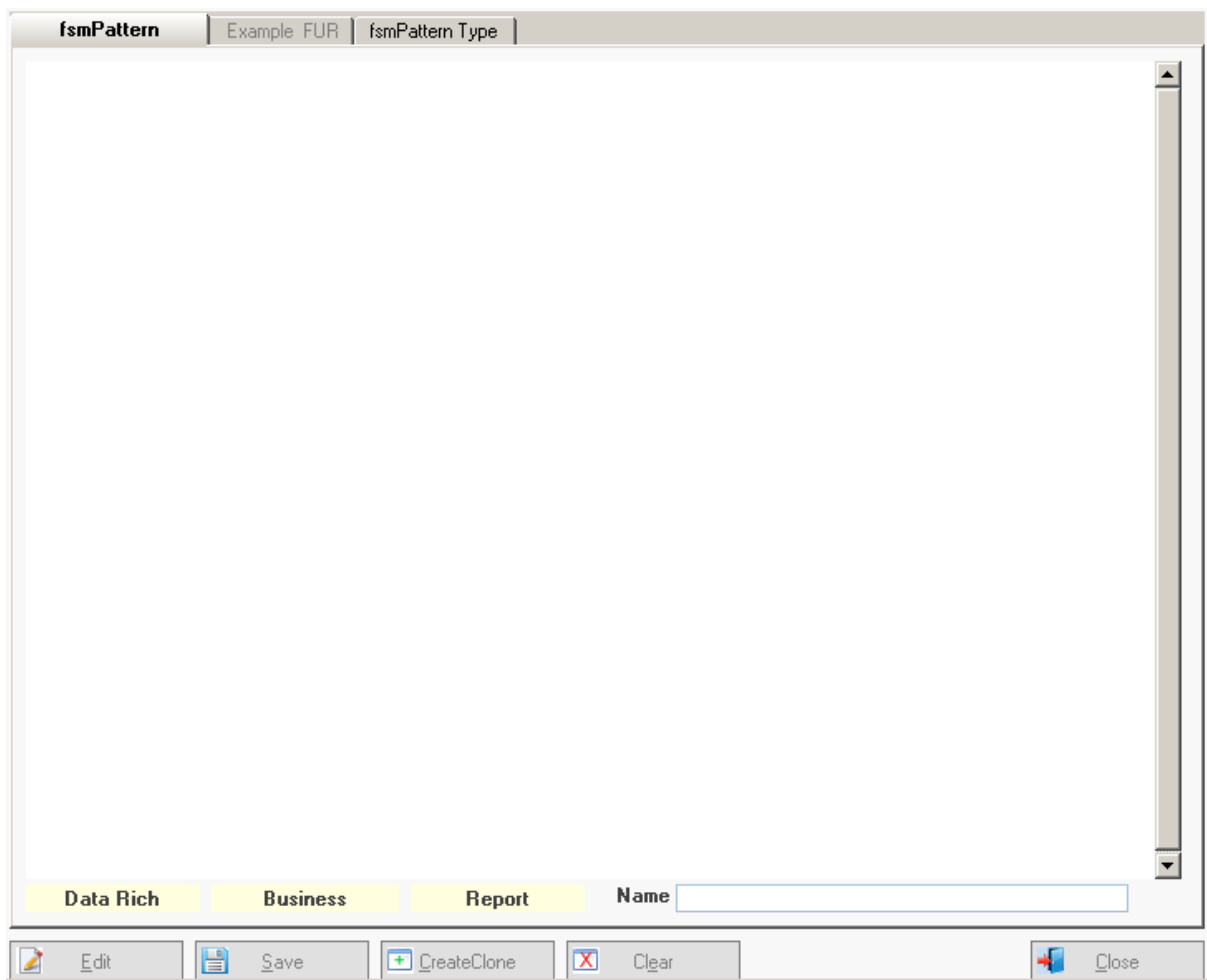
***Please note. All fsmPatterns of this type will also be deleted***

## 4 fsmPattern Maintenance

The Personal Edition of VisualFSM enables users to create and maintain fsmPatterns. It is delivered with a pre-defined set of Domains as described in the previous chapter, and for each Domain a set of fsmPattern Types is defined and within each type category a set of common fsmPatterns is defined

Existing fsmPatterns can be updated or new fsmPatterns added. fsmPatterns can also be deleted

If not already done so, start the fsmPattern Manager as described in the previous chapter. Select the **fsmPattern** tab



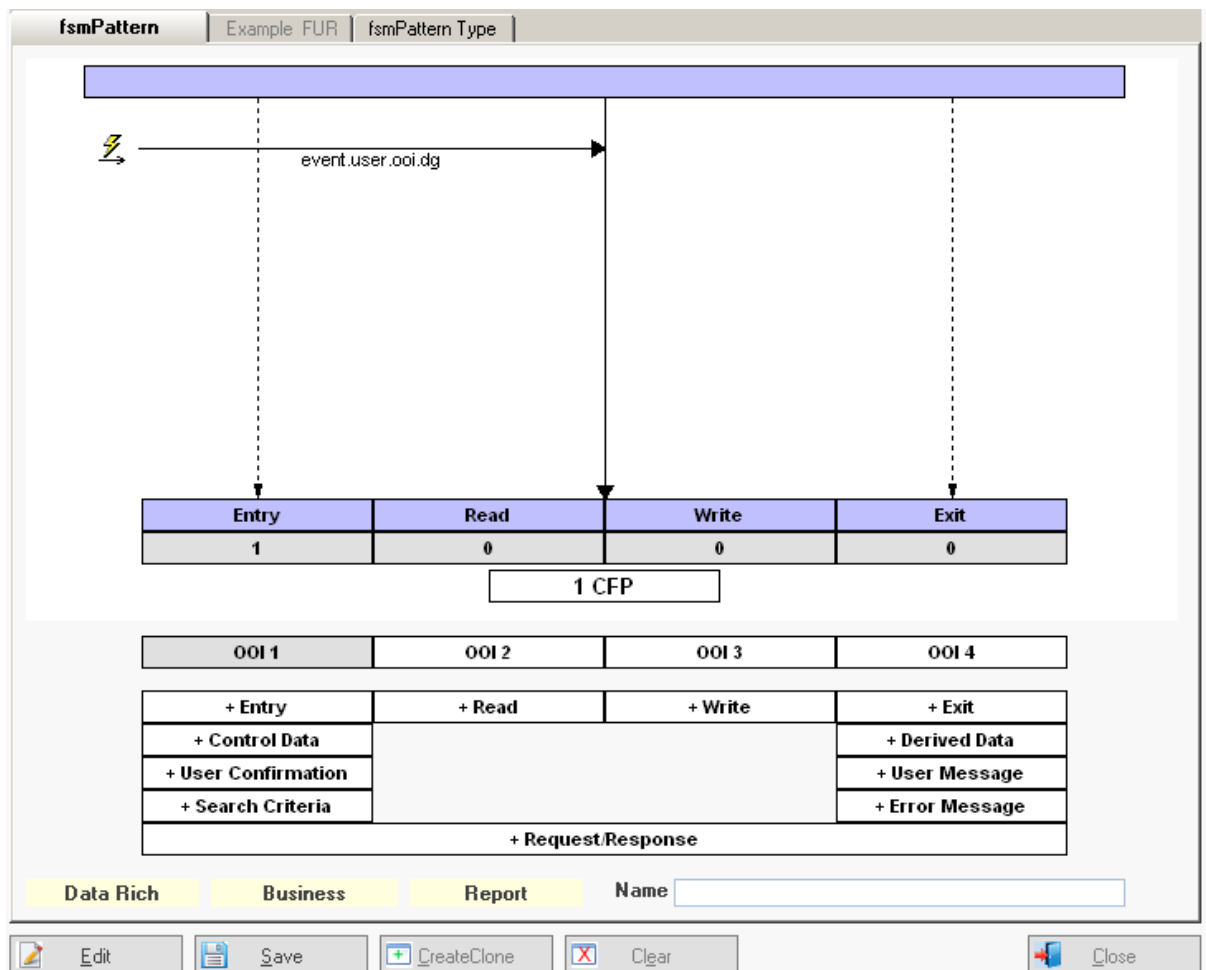
### Create a new fsmPattern

From the navigator select the fsmPattern Type, we will use the Report type just created

1. Expand the tree and select **Report**



2. Press the **Create New** button



The fsmPattern designer will be displayed showing the default Data Movement Sequence Diagram (DMSD) and the toolbox. The DMSD starts with the mandatory Triggering Entry.

3. Give the report a short meaningful name, then using the Toolbox build the Data Movement Sequence Diagram.

The Toolbox is used to add Data Movements to the diagram, it is largely self explanatory but a word about the OOI boxes

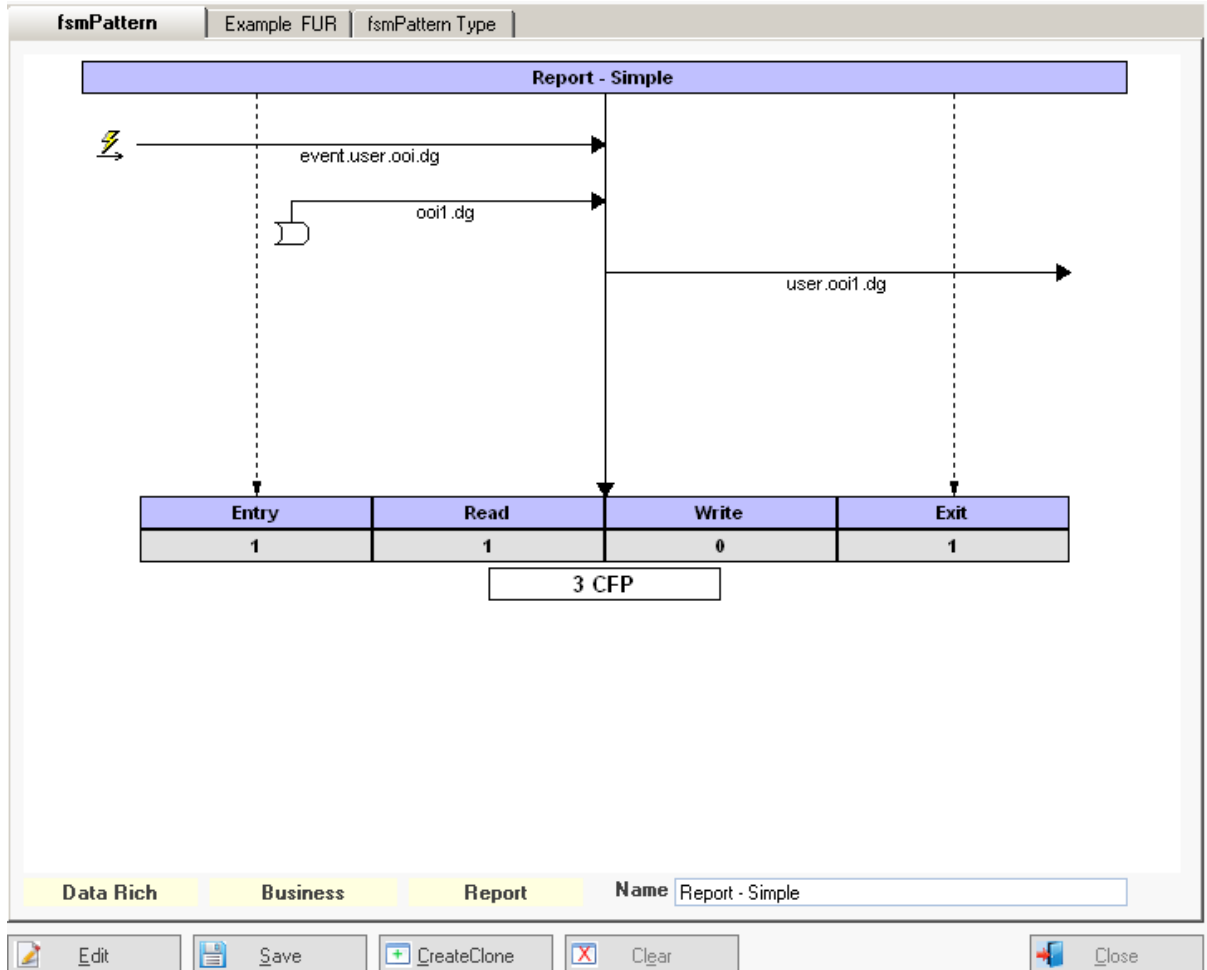
A function can involve many OOIs and for the purpose of the pattern there is no need to distinguish them. However for clarity VisualFSM can use 4 unique OOIs. if a function involves more OOIs simply re-use them.

Also, the +Request/Response box creates 2 Data Movements representing a request to another system or component and handling the response

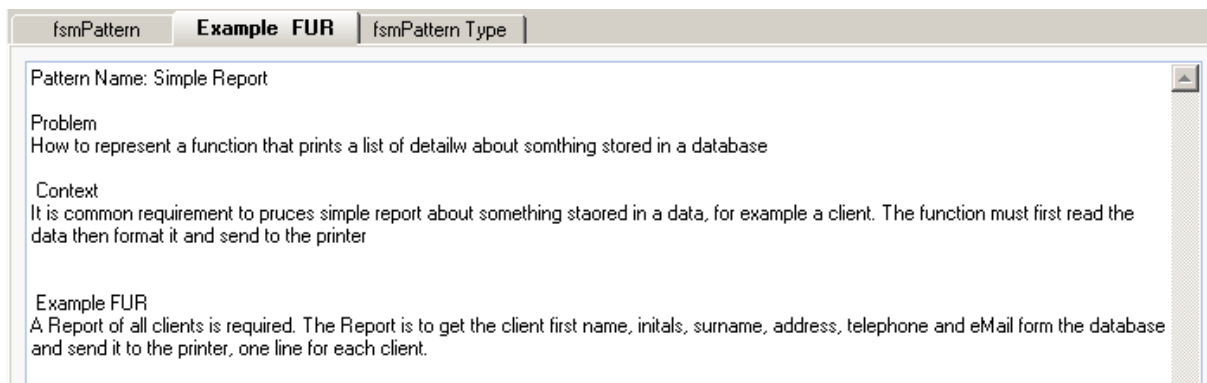
To create a pattern for a simple report we will get information from an OOI and output the data

4. Press the **OOI 1** box to make it the OOI being acted upon ( it can be changed any time)
5. Press the **+READ** box
6. Press the **+EXIT** box

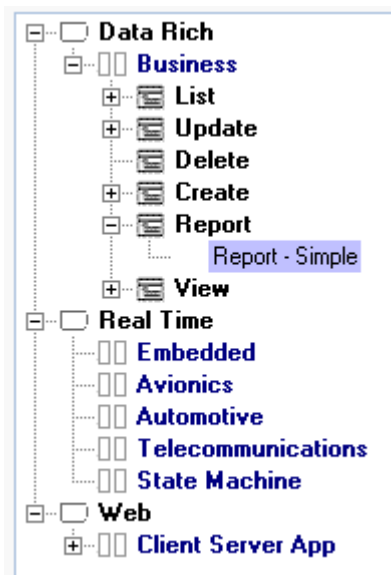
*The Data Movement Sequence Diagram is updated and the fsmPattern size is calculated.*



7. Select the **Example FUR** Tab, template is displayed. complete the details



8. Press the **Save** Button and the fsmPattern is added to the library...



## Elaborating an existing fsmPattern

When creating a set of fsmPatterns it is useful to start with the simplest then elaborate it to build a pattern with more functionality. To save having to start from scratch select and display the fsmPattern to be elaborated, in this case the one we have just created and press the **Create Clone** button.

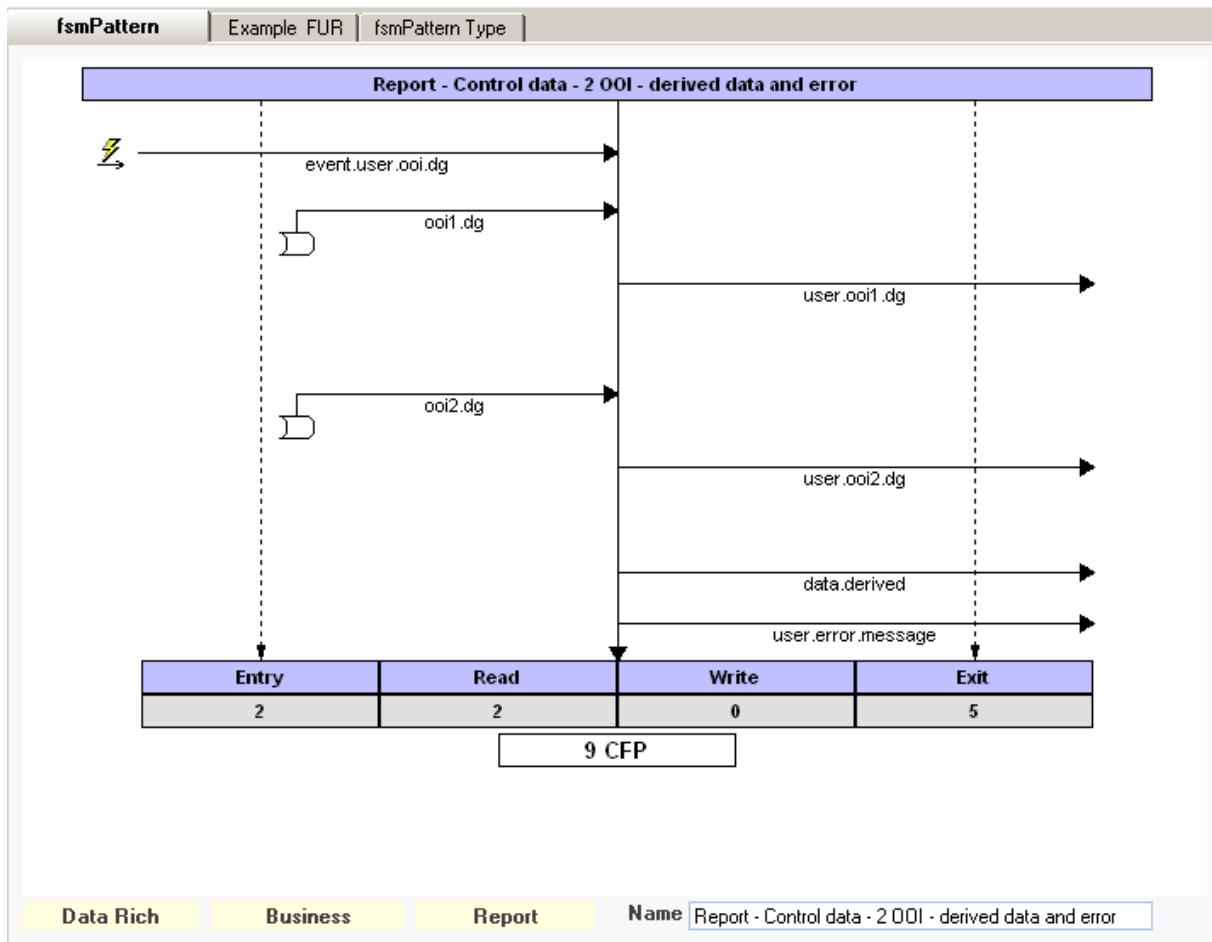
A new fsmPattern is created with the existing fsmPattern as the default. We will elaborate by adding control data, another OOI, multiple exits, derived data and an error message

1. Press the **Edit** button
2. Press the **+ Control Data** box
3. Press the **OOI 2** box to make it the OOI being acted upon ( it can be changed any time)

4. Press the + **READ** box
5. Press the + **EXIT** box
6. Press the + **Derived Data** box
7. Press the + **Error Message** box

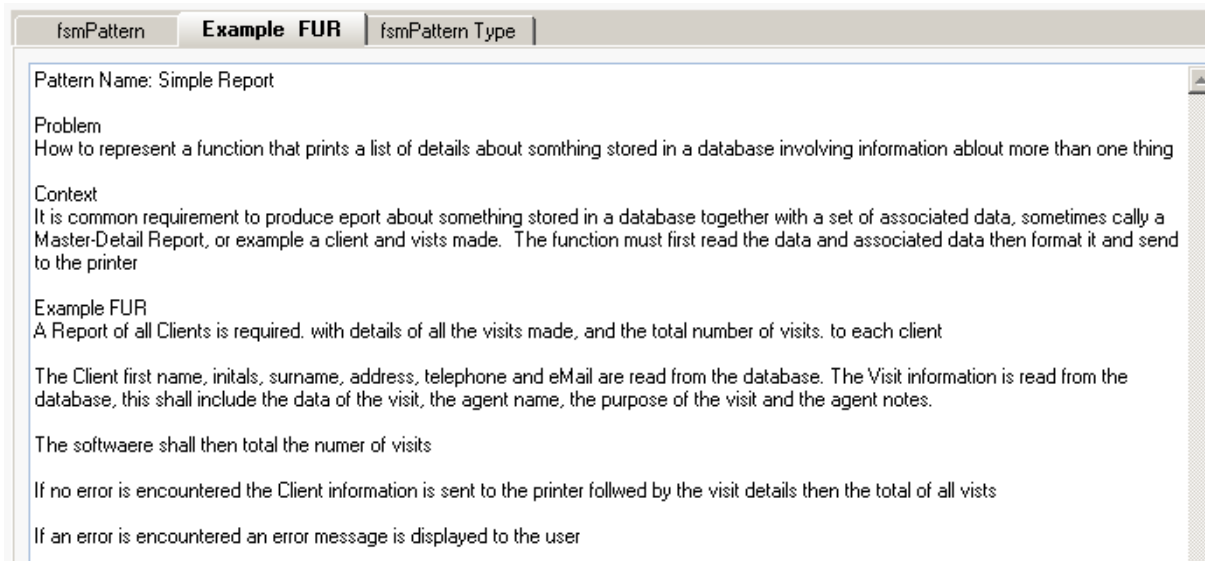
*The DMSD is built and the size is shown*

8. Give it a new name

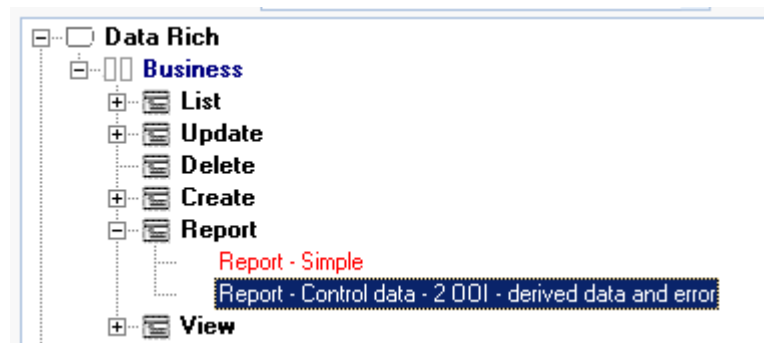


9. Press the **Example FUR** tab and complete the details ....





10. Press the **Save** button and it will be added to the library



## Changing the sequence

The sequence of the Data Movements does not affect the size but a correct sequence helps to understand the function.

To change the sequence

1. Left-Click on the Data Movement in the DMSD to select it
2. Hold down the **Shift** key
3. Place the mouse over the selected Data Movement and hold down the **Left-Mouse** button.
4. Drag the Data Movement moved up or down to the required position

*When the mouse button is released the DMSD will be redrawn (easier to do than explain)*

## Deleting a Data Movement

To Delete a Data Movement

1. **Left-Click** on the Data Movement to select it
2. **Right-Click** on the selected Data Movement,
3. The will display a pop-up menu, select **Remove Data Movement** option
4. It will be deleted from the DMSD and the size recalculated
5. Press the **Save** button

## 5 fsmPattern Exchange

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. fsmPatterns can be

- Downloaded from the Solution Farm
- Exported to a file and sent via E-Mail or Physical Media
- Imported from a File received via E-Mail or physical media

### 5.1 Downloading from the Solution Farm

#### Introduction

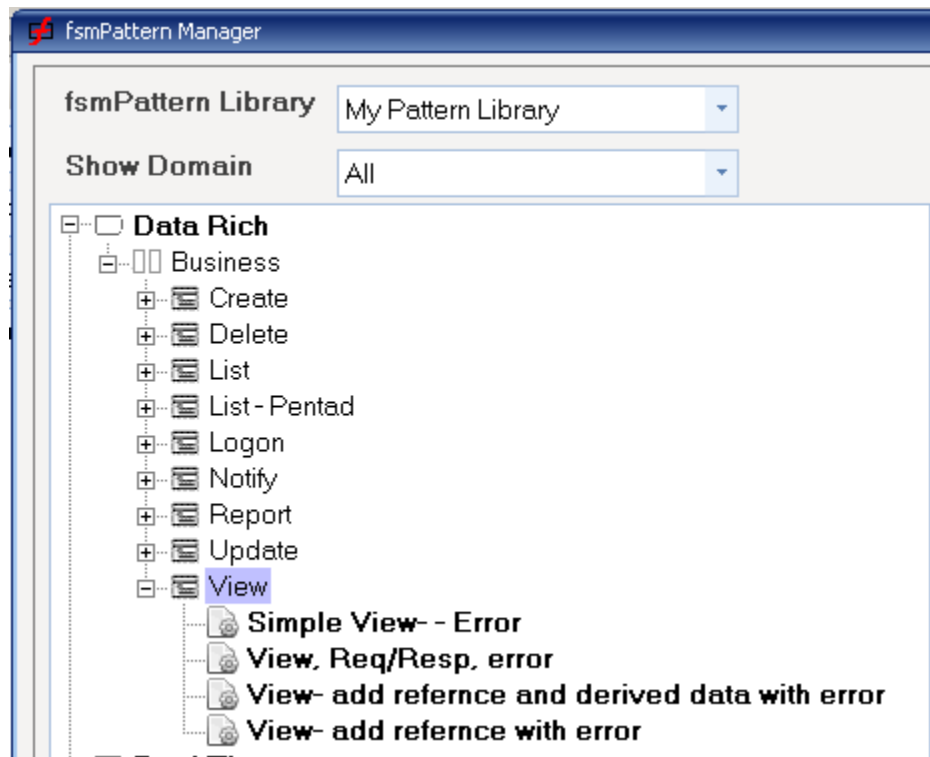
Libraries and individual Patterns can be downloaded in realtime from VisualFSM Solution Farm. The Solution farm is where resources such as Case Studies, Solutions, Documents, Architecture Patters and fsmPatterns can be downloaded in realtime.

Initially only VisualFSM are available, however as one of the aims is to allow collaboration, a separate Farm can be made available to qualifying organizations on request. Access to these farms can be universal, or controlled

#### To Upload an individual pattern

1. Open the fsmPattern as described previously,

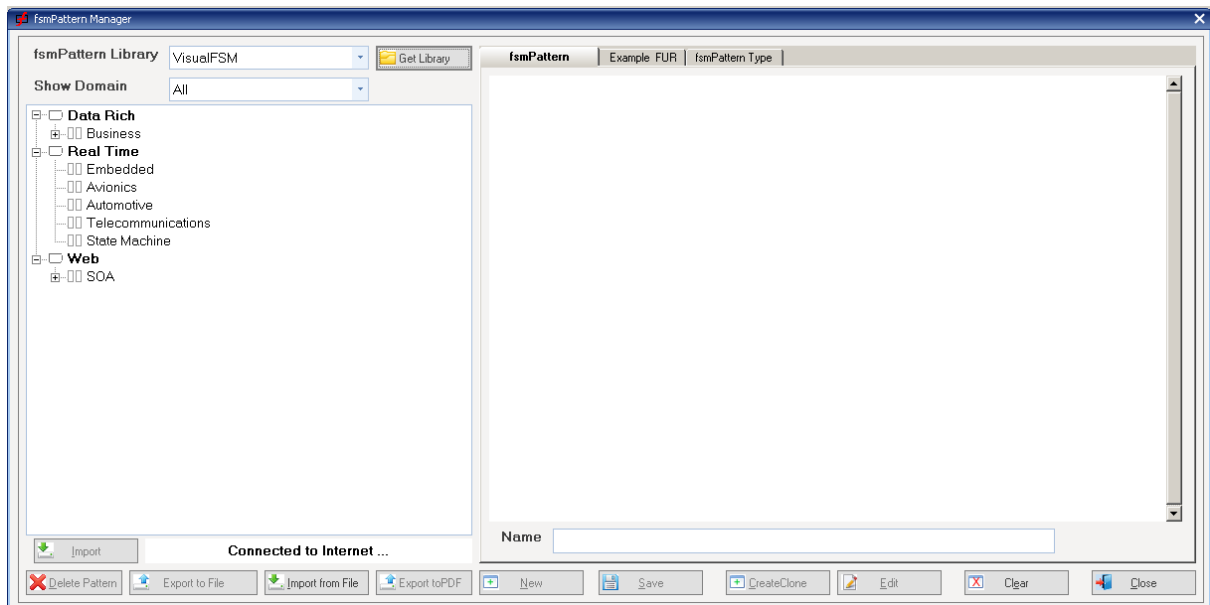
*Note: for demonstration purposes the **View fsmPatterns** have been removed from the **My fsmPatterns** library*



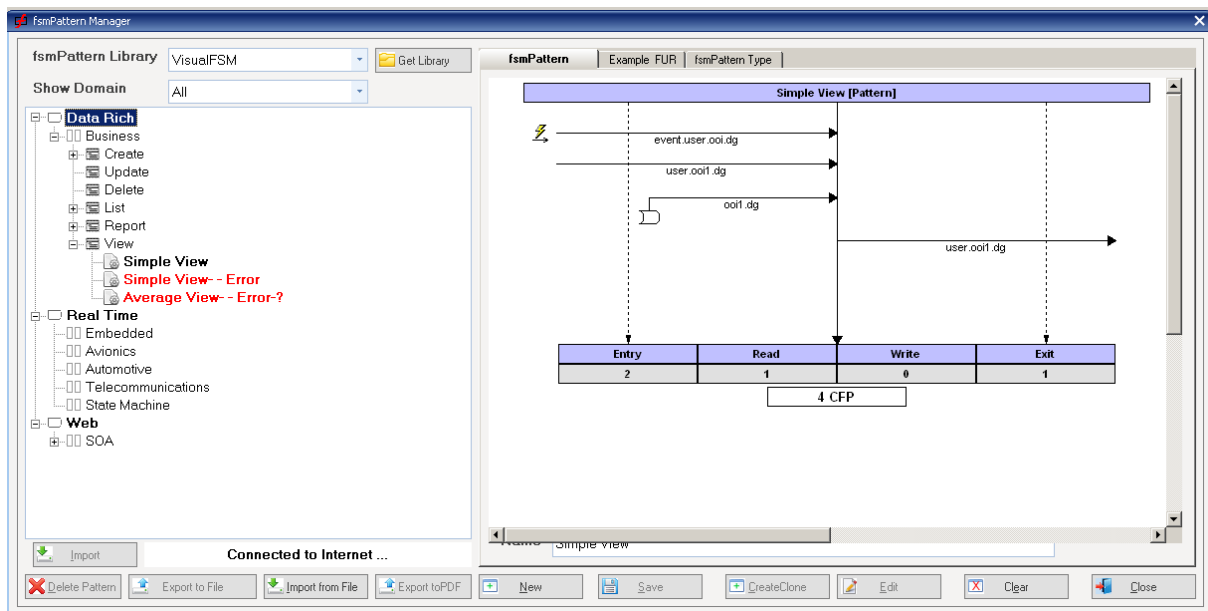
2. in the **fsmPattern Library** drop-down select **VisualFSM**

3. press the Get Library button

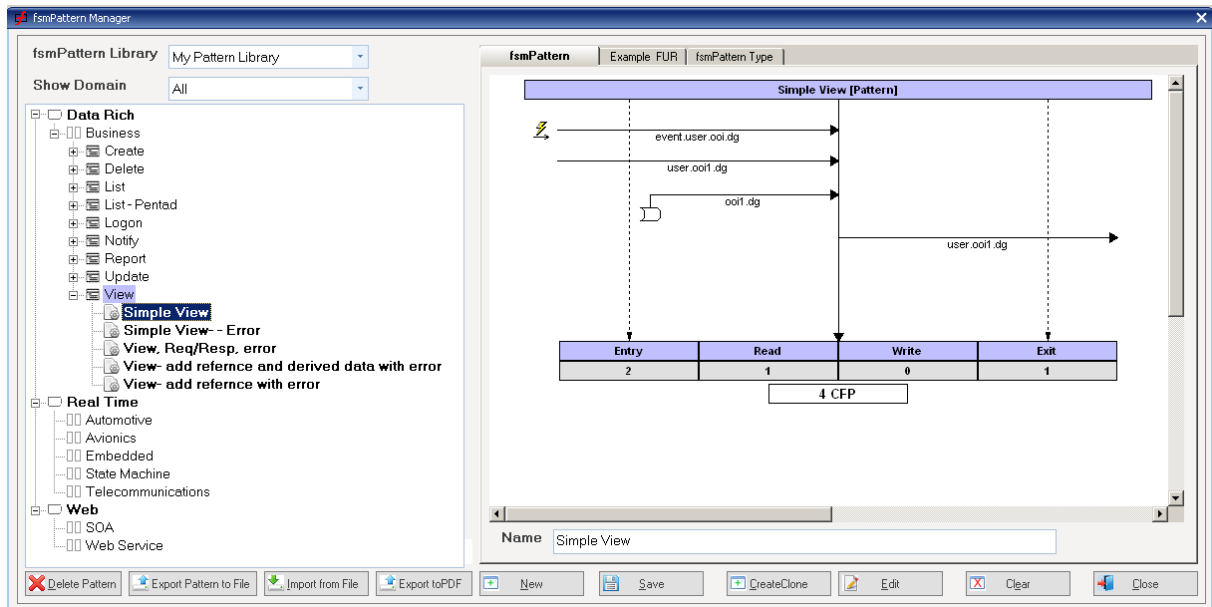
*The VisualFSM pattern Library will be displayed*



- Expand the **View** library.the Patterns in **red** already exist in you My Pattern Library.
- The **Simple View** pattern is in **black** and therefore be imported. select it  
*The selected fsmPattern will be downloaded and displayed.*



- press the **Import Selected** button  
If successful the fsmPattern will be added to the **My fsmPatterns** Library -
- select the imported **Simple View** pattern to show to show it

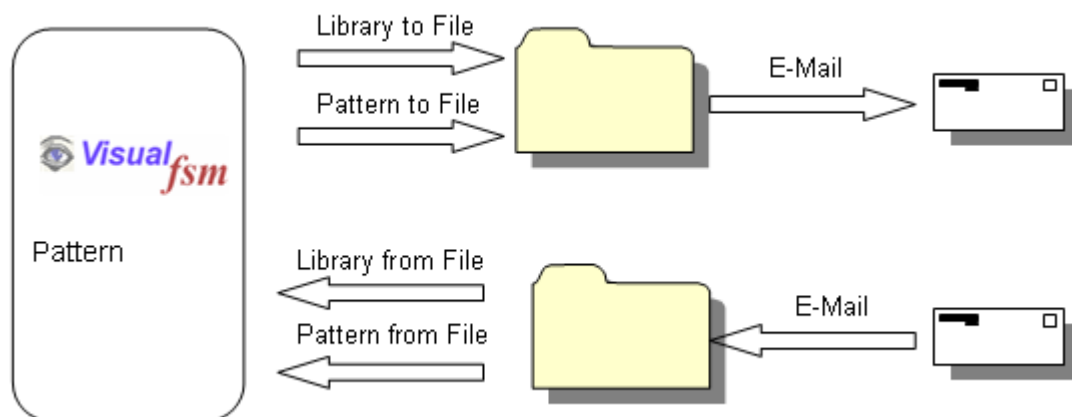


## 5.2 Export to VDIF File

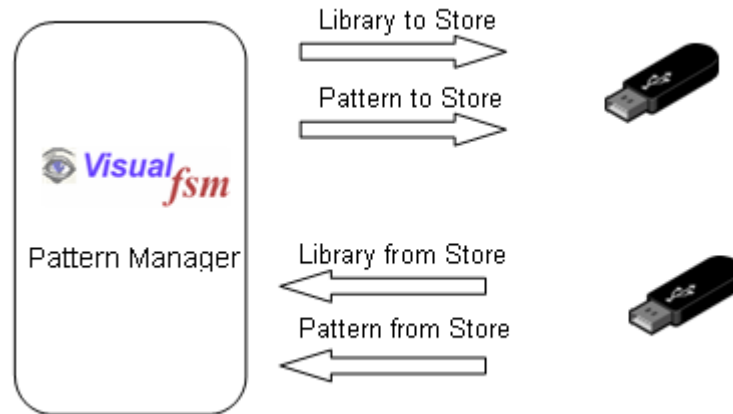
### Introduction

fsmPatterns can be exported to a file then transferred either by e-Mail or physical media. The current version will only handle individual fsmPatterns, the export and import of libraries is targeted for a future free upgrade

fsmPattern Files conforms to the **VDIF-fsmPattern** format and is saved as a JSON structure. This is text only format so can be transmitted without problem via E-Mail ....



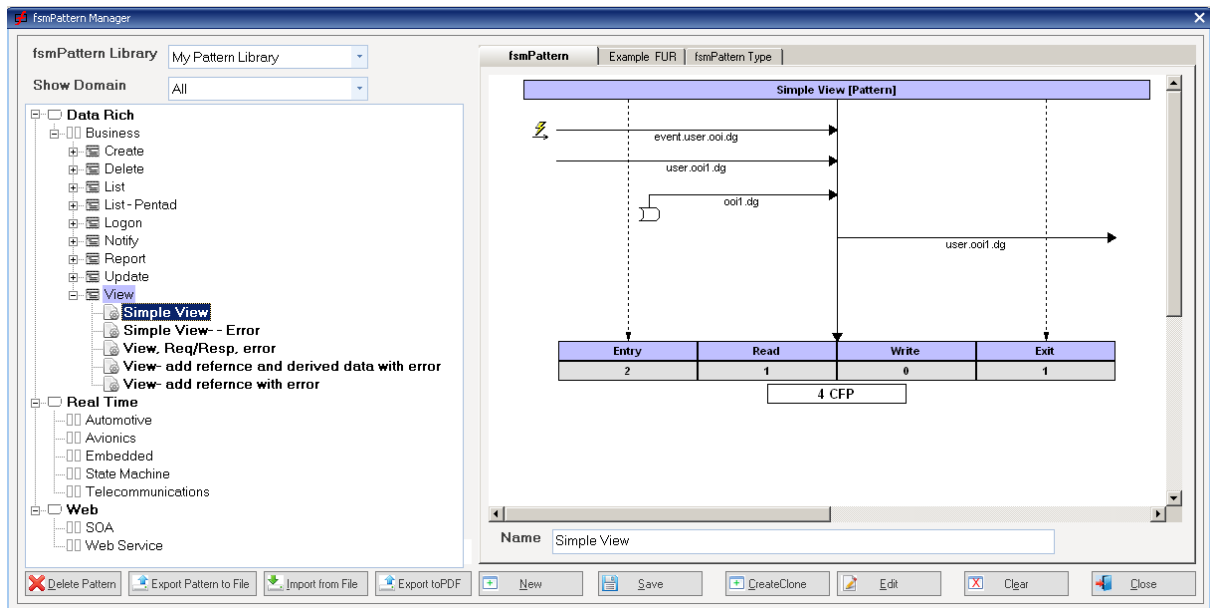
... or it can be saved and transported using or physical media.



The process is the same in either case.

In this demonstration we will export the Simple View fsmPattern, delete it then import it

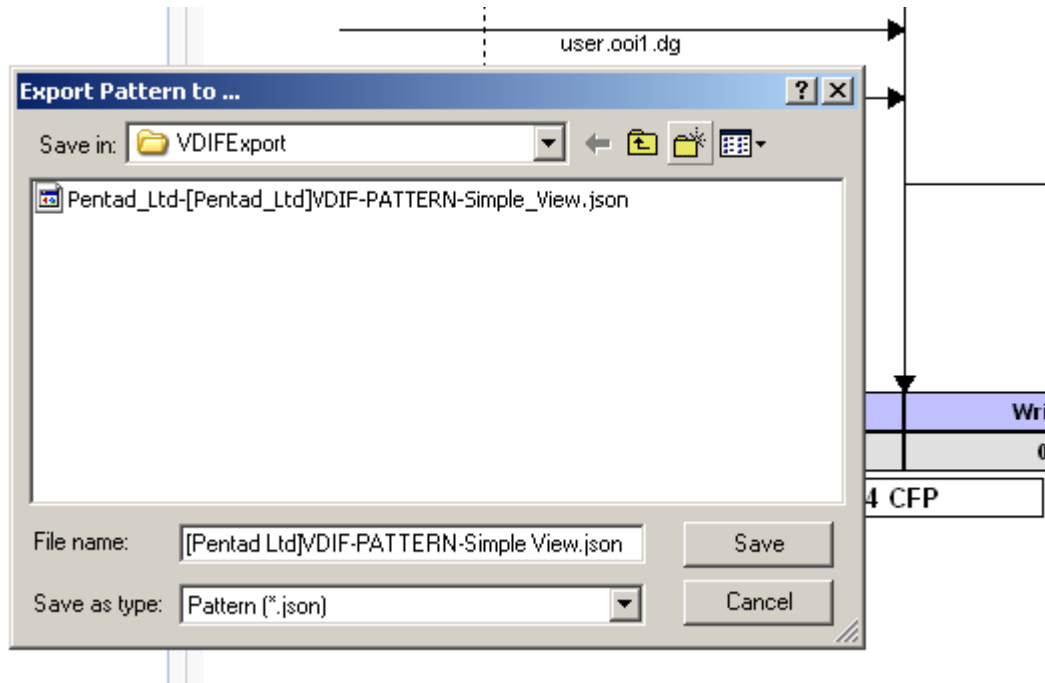
1. Start the *fsmPattern Manager* as previously described
2. navigate to and select the fsmPattern to be exported as shown below



4. Press the *Export to File* button. the standard file save dialog will be displayed,

pointing to the default export directory,

*Note: it can be renamed*



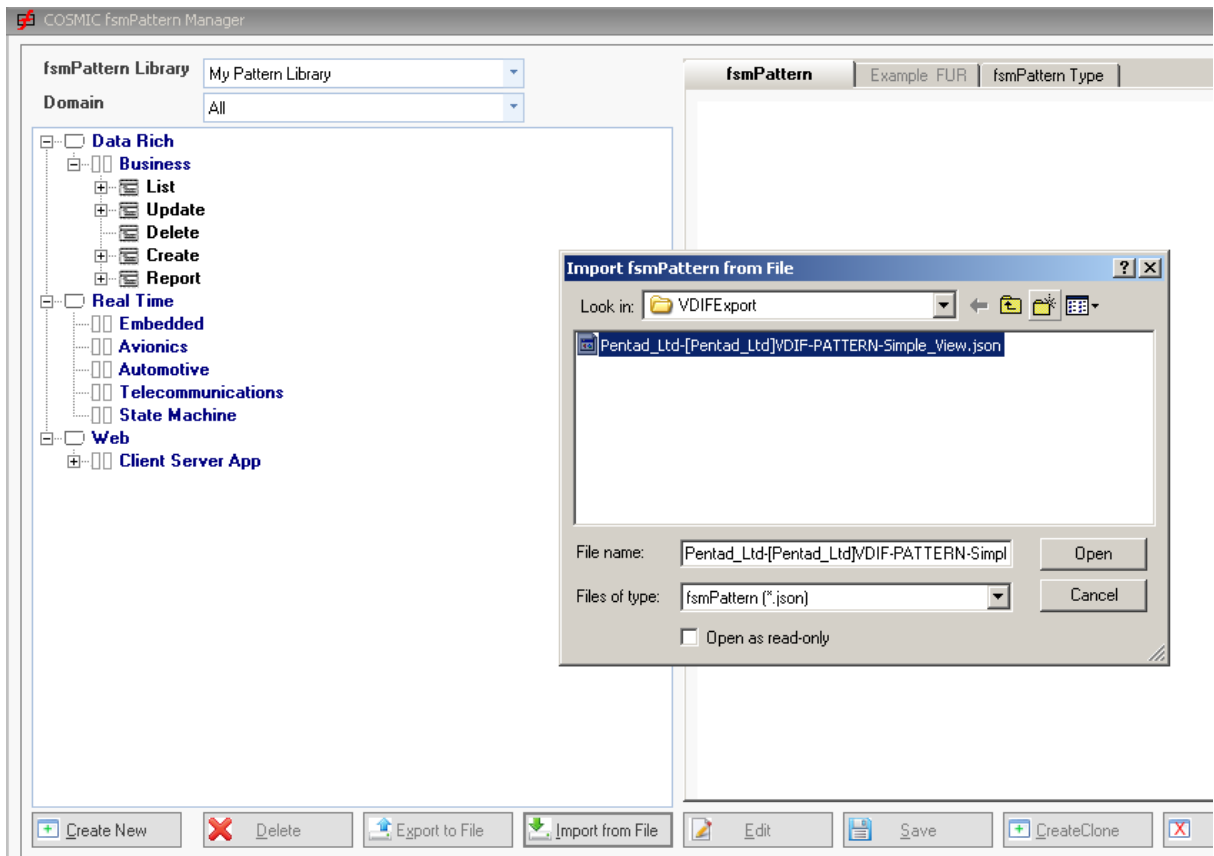
5. Press the **Save** button. .

### 5.3 Import frm VDIF File

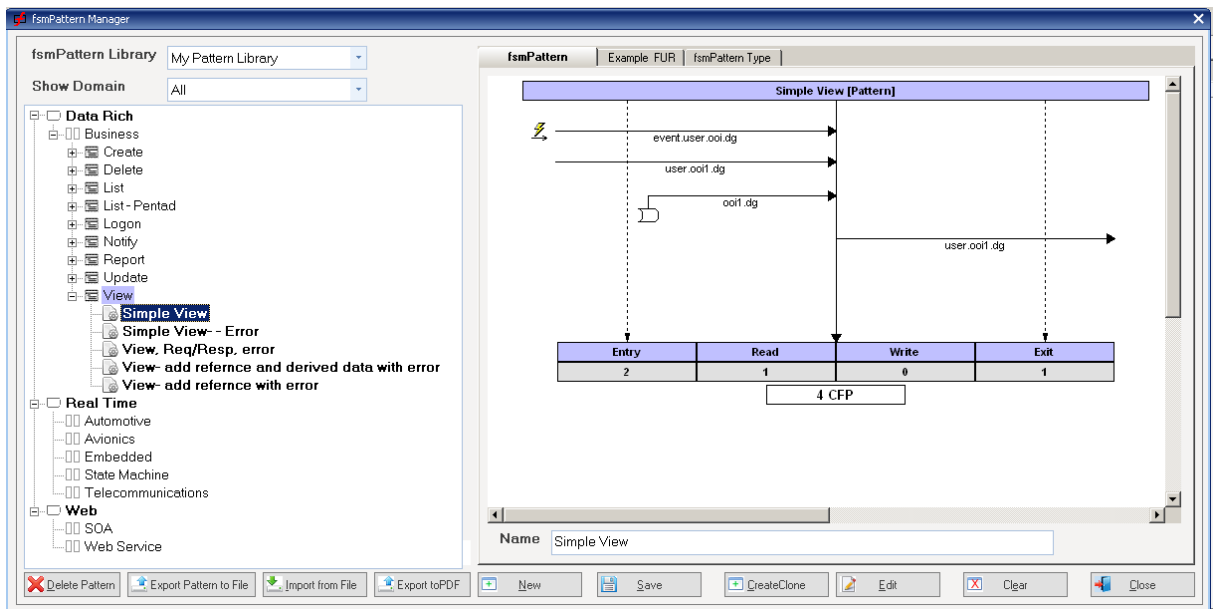
To Import a previously exported file

1. Start the *fsmPattern Manager* as previously described
2. Press the *Import from File* button,
3. the standard load file dialog will be displayed. Navigate to the required file ..





The imported pattern file will be added to My Pattern library





## 6 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 is 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