



European ITS Framework Architecture

FRAME Selection Tool Reference Manual

Version 2

September 2009

Document Control Sheet

Version History:

Version Number	Date	Editor	Summary of Changes
Version1	April 2004	P.H. Jesty	First Version produced by the FRAME-S project
Version 2 – draft 1	January 2009	R.A.P. Bossom	Updated version produced by the FRAME Forum to show changes in display layout and addition of modification forms – first draft for PHJ comment
Version 2 – draft 2	May 2009	R.A.P. Bossom	Second draft for wider comment
Version 2 – draft 3	September 2009	R.A.P. Bossom	Name changed to "Reference Manual" in view of large size

Approval:

	Name	Date
Prepared	R.A.P. Bossom	January 2009
Reviewed	P.H. Jesty	March/April 2009
Authorised		

This Reference Manual has been prepared for users of the FRAME Selection Tool as part of the FRAME methodology for creating ITS architectures. It should not be relied upon or used for any other purpose without an independent check being carried out as to its suitability and prior written authority of the FRAME Forum being obtained. The FRAME Forum does not accept any responsibility or liability for the consequence of this Reference Manual being used for a purpose other than the purposes for which it was commissioned. Any person using or relying on the Reference Manual for such other purpose agrees, and will by such use or reliance be taken to confirm his agreement to indemnify the FRAME Forum for all loss or damage resulting there from. The FRAME Forum accepts no responsibility or liability for this Reference Manual to any party other than users of the FRAME Selection Tool.

Table of Contents

1. Introduction	1
1.1 Document Purpose	1
1.2 Scope	1
1.3 Intended Audience	1
1.4 Terminology	1
1.5 Pre-requisites	1
1.6 Structure of this Reference Manual	2
1.7 Technical Support	2
1.8 Abbreviations	2
Part 1 Background	4
2. Using the FRAME Architecture	5
2.1 Introduction	5
2.2 The FRAME Browsing Tool	5
2.3 The FRAME Selection Tool	5
2.4 Maintenance and management of the FRAME Browsing and Selection Tools	6
2.5 Reporting problems and issues	6
3. The FRAME methodology for creating an ITS Architecture	7
3.1 Background	7
3.2 The Architecture Creation Process	7
3.2.1 Introduction	7
3.2.2 Creating the Functional Viewpoint	8
3.2.3 Creating the Physical Viewpoint	10
3.3 Advantages and facilities provided by the FRAME Selection Tool	11
3.4 Using the FRAME Browsing Tool to support the FRAME Selection Tool	11
Part 2 Using the FRAME Selection Tool	13
4. Obtaining the FRAME Tools	14
4.1 Introduction	14
4.2 Obtaining the FRAME Selection Tool	14
4.3 Obtaining the FRAME Browsing Tool	14
5. Using the FRAME Selection Tool – initial start-up	15
5.1 Introduction	15
5.2 Running the Tool	15
5.3 Selecting the Database	15
5.4 Starting point for the use of the Selection Tool	16
5.5 Exiting from the Selection Tool	18
6. Using the FRAME Selection Tool – the Creation of a Functional Viewpoint	19
6.1 Introduction	19

6.2	Step 1 – naming the Functional Viewpoint	19
6.3	Step 2 – selecting the User Needs	20
6.4	Step 3 - selecting the Functions	21
6.5	Step 4 – selecting the Data Flows for the Functions	23
6.6	Step 5 – selecting the Data Stores used by the Functions	25
6.7	Step 6 – selecting the Data Flows used by the Data Stores	27
6.8	Step 7 – selecting the Terminators and Actors	28
6.9	Step 8 – correcting the errors and warnings resulting from the selections	29
6.9.1	Introduction	29
6.9.2	Dealing with errors from the selection process	29
6.9.3	Fixing errors in the selection of Functions, Data Stores and Terminators	30
6.9.4	Fixing errors in the selection of Data Flows	31
6.9.5	Completing the error fixing process	32
6.9.6	Fixing warnings in the selection of Data Flows for Data Stores and/or Terminators/Actors	33
6.9.7	Using the FRAME Browsing Tool to fix faults	34
6.10	The completed Functional Viewpoint	35
6.11	Using the "Quit" button when creating the Functional Viewpoint	36
7.	Using the FRAME Selection Tool – the Creation of a Physical Viewpoint	37
7.1	Introduction	37
7.2	Step 1 – naming the Physical Viewpoint and associating it with a Functional Viewpoint	37
7.3	Step 2 – defining the Sub-systems	38
7.4	Step 2 – allocating Functions to Sub-systems	39
7.5	Step 3 – creation of Modules	40
7.6	Step 4 – allocating Functions and Data Stores to Modules	42
7.7	Step 5 – creation of Physical Data Flows	43
7.8	Using the "Quit" button when creating the Physical Viewpoint	44
8.	Creation of Reports showing the results from the creation process	45
8.1	Introduction	45
8.2	Selecting the type of "report" for the Functional Viewpoint	45
8.3	Selecting the type of "report" for the Physical Viewpoint	46
8.4	Previewing and saving reports	46
8.5	Making reports easier to read	47
8.6	Other "reports" for the Physical Viewpoint	48
8.7	Producing Diagrams for the Physical Viewpoint	50
9.	Modifying existing Viewpoints	51
9.1	Introduction	51
9.2	Modifying an unused Functional Viewpoint	51
9.3	Modifying a used Functional Viewpoint	52
9.4	Modifying the Physical Viewpoint	53
10.	Hints and Tips for using the FRAME Selection Tool	55
10.1	Introduction	55

10.2	When no errors or warnings are reported, does this mean that the Functional Viewpoint is correct?	55
10.3	Can a Functional Viewpoint be modified after it has been used to create a Physical Viewpoint?	55
10.4	Can a Physical Viewpoint be modified after its creation has been finished	55
10.5	Can different Physical Viewpoints be produced from the same Functional Viewpoint?	55
10.6	Locations for use in the Physical Viewpoint	56
10.7	Locating Data Stores	56
10.8	Pre-preparation – Functional Viewpoint	57
10.9	Pre-preparation – Physical Viewpoint	57
10.10	Which Functional Viewpoint "reports" to produce?	57
10.11	Which Physical Viewpoint "reports" to produce?	58
10.12	Creating Physical Data Flow Names	58
10.13	Using the same Function in different Sub-systems or Modules	58
10.14	Allocating Functions and Data Stores to physical locations	59
10.15	How many Viewpoints can be produced?	59
10.16	Are there any constraints on the scope of a Functional Viewpoint	59
Part 3 Adding extra functionality into the FRAME Database		60
11.	Methodology for making additions to the FRAME Database	61
11.1	Introduction	61
11.2	Before starting to make additions	61
11.3	The process for making additions to the FRAME Database	62
11.4	Limitations of the additions process	62
12.	General principles for modifying the FRAME Database	63
12.1	Introduction	63
12.2	Opening the FRAME Database	63
12.3	Selecting the "Forms" option	64
13.	Entering new User Needs	66
13.1	Introduction	66
13.2	Opening the new User Need Form to add extra User Needs	66
13.3	User Needs numbering system	67
13.4	Structure of the User Needs in the FRAME Database	67
13.5	Selecting the identifier for each new User Need	68
13.6	Entering the new User Needs	68
14.	Entering new Functions	70
14.1	Introduction	70
14.2	Opening the new Function Form	70
14.3	Function numbering and naming system	70
14.4	Selecting the number for each new Function	71
14.5	Entering the new Functions	71

15. Entering new Functional Areas and Sub-Areas	73
15.1 Introduction	73
15.2 Opening the new Functional Area Form	73
15.3 Opening the new Functional Sub-Area Form	74
16. Entering new Data Stores	76
16.1 Introduction	76
16.2 Opening the new Data Store Form	76
17. Entering new Actors and Terminators	78
17.1 Introduction	78
17.2 Opening the new Actors Form	78
17.3 Creating new Terminators	79
18. Entering new User Needs to Function relationships	81
18.1 Introduction	81
18.2 Opening the new User Needs to Function relationships Form	81
19. Entering new Data Flows	84
19.1 Introduction	84
19.2 Opening the new Data Flows Form	84
20. Actions when all additions to the FRAME Database are complete	86
20.1 Introduction	86
20.2 Checking for errors in the modified FRAME Database	86
20.3 Making the changes to the FRAME Database permanent	86
Appendix 1 FRAME Database Structure and Relationships	87
A 1.1 Introduction	87
A 1.2 FRAME Database Tables	87
A 1.2.1 Introduction	87
A 1.2.2 Table: Actor	87
A 1.2.3 Table: Data Flow	87
A 1.2.4 Table: Data Store	88
A 1.2.5 Table: Function	89
A 1.2.6 Table: Functional Area	89
A 1.2.7 Table: Functional Sub-Area	89
A 1.2.8 Table: RelationUserNeedFunction	89
A 1.2.9 Table: Terminator	89
A 1.2.10 Table: UserNeed	89
A 1.2.11 Table: UserNeedGroup	89
A 1.2.12 Table: UserNeedService	89
A 1.2.13 Table: UserNeedTopic	89
A 1.3 Relationships between the tables	90
Appendix 2 FRAME User Needs structure	91
Appendix 3 Functional Areas included in the FRAME Architecture	96
Appendix 4 A template for creating new Functions	98

Table of Figures

Figure 1 – FRAME Tool Structure	5
Figure 2 - ITS Architecture creation process	7
Figure 3 – Creation of the Functional Viewpoint	9
Figure 4 – Creation of the Physical Viewpoint	10
Figure 5 – Selection Tool Opening screen	15
Figure 6 – Database Selection screen with result	16
Figure 7 – Viewpoint drop down menu	17
Figure 8 – Naming the Functional Viewpoint	19
Figure 9 – Selecting User Needs	20
Figure 10 – Selecting Low-level Functions	21
Figure 11 – Adding the selected low-level Functions to the Functional Viewpoint	22
Figure 12 – Selecting Data Flows	23
Figure 13 – Adding the selected Data Flows to the Functional Viewpoint	24
Figure 14 – Selecting Data Stores	25
Figure 15 – Adding the selected Data Stores to the Functional Viewpoint	26
Figure 16 – Adding the Data Flows for the selected Data Stores into the Functional Viewpoint	27
Figure 17 – Selecting Terminators and Actors	28
Figure 18 – Example of the error messages produced from the initial creation of a Functional Viewpoint	30
Figure 19 – Example of the select Functions and error messages windows	31
Figure 20 – Selection and de-selection of Data Flows	32
Figure 21 – Screen shot showing the display of warning messages	33
Figure 22 – Illustration of the use of the FRAME Browsing Tool	34
Figure 23 – Completed Functional Viewpoint screen	35
Figure 24 – The "Quit" option in the Functional Viewpoint	36
Figure 25 – Naming the Physical Viewpoint and associating it with a Functional Viewpoint	37
Figure 26 – Screen from which Sub-systems are defined	38
Figure 27 – Setting locations for Sub-systems	39
Figure 28 – Allocation of Functions to Sub-systems	40
Figure 29 – Defining Modules	41
Figure 30 – Allocation of Functions and Data Stores to Modules	42
Figure 31 – List of Physical Data Flows	43
Figure 32 – Options for the creation of Functional Viewpoint "reports"	45
Figure 33 – Options for the creation of Physical Viewpoint "reports"	46

Figure 34 – An example of a preview of a Functional Viewpoint "report"	47
Figure 35 – An example of a Physical Data Flows "report"	48
Figure 36 – An example of a Function allocation "report"	49
Figure 37 – An example of a Data Store allocation "report"	49
Figure 38 – An example of a Sub-system and Module diagram produced for a Physical Viewpoint	50
Figure 39 – Selecting a previously completed Functional Viewpoint for modification	51
Figure 40 – Creating a "clone" Functional Viewpoint	52
Figure 41 – naming a cloned Functional Viewpoint	53
Figure 42 – Selecting a previously completed Physical Viewpoint for modification	54
Figure 43 – Complete process for adding to the FRAME Database	62
Figure 44 – Microsoft® Access® Opening Screen Shot	63
Figure 45 – Microsoft® Access® "Forms" Facility Screen Shot	64
Figure 46 – Adding new User Need screen	66
Figure 47 – Entry of information for a new User Need	68
Figure 48 – Entry of information a new User Need Topic	69
Figure 49 – Adding new Function screen	70
Figure 50 – Entry of information for a new Function	72
Figure 51 - Adding new Functional Area screen	73
Figure 52 – Adding new Functional Sub-Area screen	74
Figure 53 – Adding new Data Store screen	76
Figure 54 – Adding new Actor screen	78
Figure 55 – Entry of information for a new Actor	79
Figure 56 - Adding new Terminator screen	80
Figure 57 – Entry of information for a new Terminator	80
Figure 58 – Adding new User Needs to Function relationships screen	81
Figure 59 – New User Need selection drop down list	82
Figure 60 – Function selection drop down list	82
Figure 61 – Adding new Data Flows screen	84
Figure 62 – Microsoft® Access® Opening Screen Shot for FRAME Database	88
Figure 63 – FRAME Database Table Relationships	90
Figure 64 – Use of Function Data Entry Template Data	98

Table of Tables

Table 1 – Groups, Services and Topics in the FRAME User Needs	91
Table 2 – a list of the Functional Areas within the FRAME Architecture	96

1. Introduction

1.1 Document Purpose

The purpose of this Reference Manual is to provide a comprehensive "user guide" for the FRAME Selection Tool so that users can be guided through the processes that are supported by the Tool. These processes enable Functional and Physical Viewpoints to be created from the European ITS Framework Architecture (usually called the FRAME Architecture). The Physical Viewpoint is used as the starting point for the creation of specifications for the components required by the ITS implementation for which the Viewpoint has been created. It is also used as the starting point for an analysis of the communications that will be needed to support these components.

1.2 Scope

This Reference Manual is designed for use with Version 3.2 (or later) of the FRAME Architecture and its Database. The database will be in Microsoft® Access® 2000 or a later format. The Manual can also only be used with Version 1.2.2 or later of the Selection Tool.

The list of User Needs and Functions contained in the Appendices to this Manual applies to the indicated version of the FRAME Architecture and its Database. The contents will be updated to be compatible with later versions of the FRAME Architecture and its Database.

1.3 Intended Audience

The intended audience for this Reference Manual is current and prospective users of the FRAME Selection Tool. They are expected to be ITS Architecture Developers who want to base the plans for their new ITS implementations on the FRAME Architecture.

1.4 Terminology

It is unfortunate that the terminology applicable to this field of engineering is not used in a consistent manner. In the following list, the first word (underlined) is the term used in this document, those after the "/" are other alternatives.

Stakeholder Aspiration / User Requirement / Stakeholder Requirement

User Need / User Requirement

Functional Viewpoint /Logical Architecture / Functional Architecture

Physical Viewpoint / Physical Architecture

Communications Viewpoint / Communications Architecture

1.5 Pre-requisites

In its description of how to use the FRAME Selection Tool this User Reference Manual assumes that the user of the Tool understands the basic process of creating an ITS Architecture. For those who do not, it is recommended that they consult the FRAME website (<http://www.frame-online.net>) from which it is possible to download

the presentations that are used in the FRAME Workshop. Alternatively the website can be consulted to see if any Workshops have been planned for the future.

Any FRAME Selection Tool user wanting to add extra User Needs and Functionality to the FRAME Database used by the Tool, should be comfortable with the use of Microsoft® Access® (Versions 2000 to 2003 inclusive). It should not be necessary to create any new forms, or reports, but users will find it useful to have a basic working knowledge of how they work.

One of the consequences of having attended a FRAME Workshop (or have worked through the Workshop materials) is that attendees will be familiar with the organisation and structure of the FRAME Architecture and the methodology behind it. The structure of the FRAME Architecture is an integral part of the way that the FRAME Database tables and their entries are organised. This needs to be clearly understood before adding any new entries to the Database.

1.6 *Structure of this Reference Manual*

This Reference Manual is divided into three parts plus a set of Appendices. The Appendices contain detailed descriptions of the tables in the FRAME Database, plus the structure of the User Needs, a list of Functional Areas with their descriptions and a suggested template for use in the creation of new Functions.

Part 1 contains background information. This consists of an overview of the development of the FRAME Selection Tool and a description on the architecture creation methodology that it supports. It also includes brief notes about the FRAME Browsing Tool, which can be used in conjunction with the FRAME Selection Tool.

Part 2 is a guide to the use of the FRAME Selection Tool. The Chapters in this part cover everything from downloading the Tool and its Database from the FRAME website, through the Viewpoint creation process, to producing the reports showing the contents of the Functional and Physical Viewpoints that the Tool produces. The involvement of the FRAME Browsing Tool in the use of the Selection Tool is also covered.

Part 3 contains a guide to the addition of extra functionality into the FRAME Architecture as it is represented in the FRAME Database. This enables the Tool to be used to create super-sets of the FRAME Architecture for particular ITS implementations. However it will not be possible to see the additional functionality using the FRAME Browsing Tool or to make the changes permanent for all other users of the FRAME Architecture.

1.7 *Technical Support*

Any errors in this Reference Manual, or problems with using the FRAME Selection and Browsing Tools, adding new entities to the FRAME Database, or obtaining the reports directly from the Database should be reported. This can be done by sending an e-mail to: info@frame-online.net with a document attached containing a full problem description.

1.8 *Abbreviations*

The following abbreviations have been used in this Reference Manual;

CSV	Comma Separated Variables: this is a special file in which every entity is separated by a comma. When read by Microsoft® Excel® it can be converted into a spreadsheet in its own format.
D	Data Store – used as the type in the number part of the Data Store identifier, e.g. D4.1.
EC	European Commission
F	Function – used as the type in the number part of the Function identifier, e.g. F2.1.2.1.
FFM	Freight and Fleet Management
FRAME	The generic name of the EC funded projects that created version 3 of the European ITS Framework Architecture, plus the FRAME Selection and Browsing Tools. The projects also promoted the use of the Architecture which has given rise to the use of "FRAME Architecture" as its alternative name.
HTML	Hyper Text Mark-up Language - it provides a means to describe the structure of text-based information in a document – for further information see http://en.wikipedia.org/wiki/HTML .
ID	Identity – used as part of some FRAME Database field names
PT	Public Transport
T	Terminator – used as the type identifier for Terminators and their components, Actors.
TCC	Traffic Control Centre
TIC	Travel Information Centre
TMC	Traffic Management Centre
UN	User Need

Part 1 Background

This part of the Reference Manual provides background information about the creation and contents of the FRAME Selection and Browsing Tools. It consists of an overview of the development of the both Tools and a description on the architecture creation methodology that they support.

It contains the following two Chapters:

Chapter 2: Using the FRAME Architecture

Chapter 3: The FRAME methodology for creating an ITS Architecture

This part of the Manual provides information for those who wish to understand the methodology behind the FRAME Browsing and Selection Tools. It also provides an insight into how the Tools are produced and maintained.

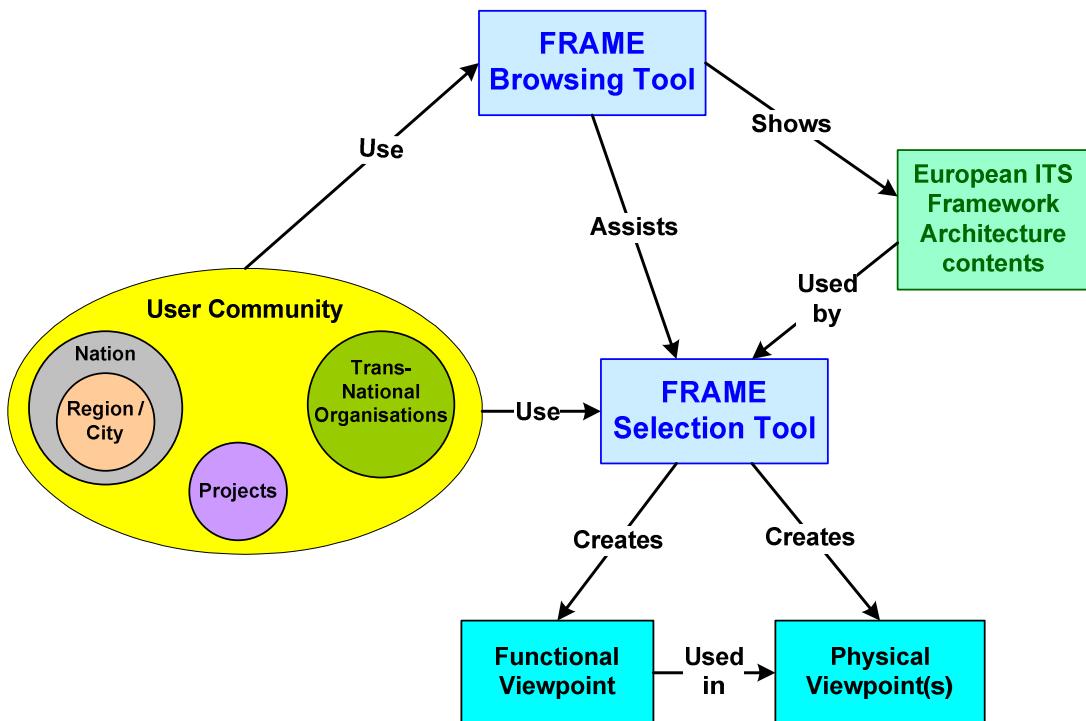
2. Using the FRAME Architecture

2.1 *Introduction*

Any facility that enables the use of the FRAME Architecture needs to provide its users with two principal features (see Figure 1):

- The ability to **browse** through the FRAME Architecture, following related elements and being provided with their definitions.
- The ability to **select** a sub-set of the Functional Viewpoint in the FRAME Architecture that satisfies a sub-set of the European ITS User Needs, and then to create a corresponding Physical Viewpoint of that sub-set.

Figure 1 – FRAME Tool Structure



2.2 *The FRAME Browsing Tool*

The FRAME Browsing Tool enables users to view the contents of the FRAME Architecture. It consists of HTML pages created from the contents of the Architecture. These pages can be viewed using a standard Internet browser such as Microsoft® Internet Explorer, Netscape® Navigator, or Mozilla FireFox. The HTML pages that make up the FRAME Browsing Tool can be freely downloaded from the FRAME website.

2.3 *The FRAME Selection Tool*

The FRAME Selection Tool provides a facility that enables some or all FRAME Architecture to be selected and used to create a particular ITS architecture. This architecture will initially comprise a set of User Needs and the corresponding Functional Viewpoint. The Functional Viewpoint is used by the Tool to create one or

more Physical Viewpoints. Together the User Needs and the Viewpoints make up the particular ITS architecture.

The Tool uses a Microsoft® Access® database. This together with the Tool itself can also be downloaded from the FRAME website. Initially the Database only contains the contents of the FRAME Architecture. However the Tool will expand it to include the contents of all other ITS architectures that the user creates. If necessary the Database can be copied and different ITS architectures created from the copies.

2.4 *Maintenance and management of the FRAME Browsing and Selection Tools*

The FRAME Browsing and Selection Tools are maintained and managed by the support facility provided through the FRAME website. At the moment this operates as part of the E-FRAME project, but once this has finished (May 2011) it will come under the direction of the FRAME Forum. The FRAME Forum is the "user group" for the FRAME Architecture and information about it can be obtained from the FRAME website at: <http://www.frame-online.net>.

The FRAME Browsing Tool and Microsoft® Access® database used by the FRAME Section Tool are produced directly from the proprietary tool used by the FRAME Team to maintain the FRAME Architecture. This means that whenever a new version of the Architecture is produced, a new version of the FRAME Browsing Tool and the database will also be produced and made available from the FRAME website. The version numbers of the FRAME Browsing Tool and the Microsoft® Access® database will be kept in step with that of the FRAME Architecture.

Changes to the FRAME Selection Tool are made independently of those to the FRAME Architecture and hence the FRAME Browsing Tool plus the Microsoft® Access® database. This means that the version number of the FRAME Selection Tool will not be kept in step with that for the FRAME Architecture.

2.5 *Reporting problems and issues*

Any problems and issues that users have with the FRAME Browsing and Selection Tools (including this Reference Manual) must be sent to the support facility provided through the FRAME website. This can be accessed by sending an e-mail to: info@frame-online.net to which is attached a document containing the full description of the problems and issues. A response with a description of the action to be taken will be provided to the originator.

3. The FRAME methodology for creating an ITS Architecture

3.1 *Background*

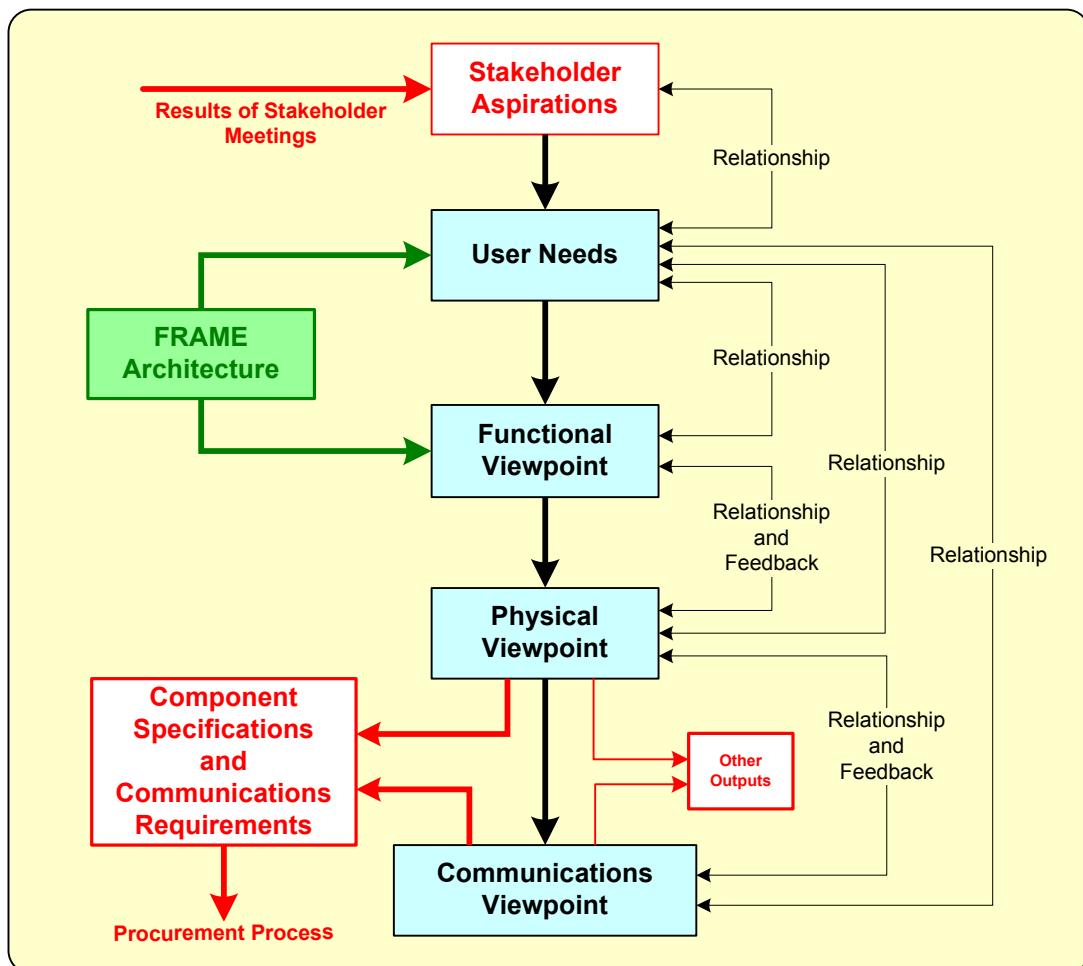
Although the FRAME Architecture is only made up of two components, it is very large. This because it has to provide support for an increasingly wide range of services that are available for inclusion in any ITS implementation. These services include such things as trip planning, journey assistance, road network management, fleet management (for both Public Transport (PT) and goods vehicles), determining the need for payment for both road and PT use as well as other services, and support for enforcement. This means that the process of going from a set of User Needs to the identification of physical entities needed to implement them needs to be carefully managed.

3.2 *The Architecture Creation Process*

3.2.1 *Introduction*

The process used to create an ITS architecture is divided into several parts. These are shown in simplified form by the diagram in Figure 2.

Figure 2 - ITS Architecture creation process



The FRAME Selection Tool provides a user friendly mechanism for the creation of the Functional and Physical Viewpoint. The details of how this is done and how the Stakeholder Aspirations are mapped to the User Needs and the Communications Viewpoint is created are provided in the following sections.

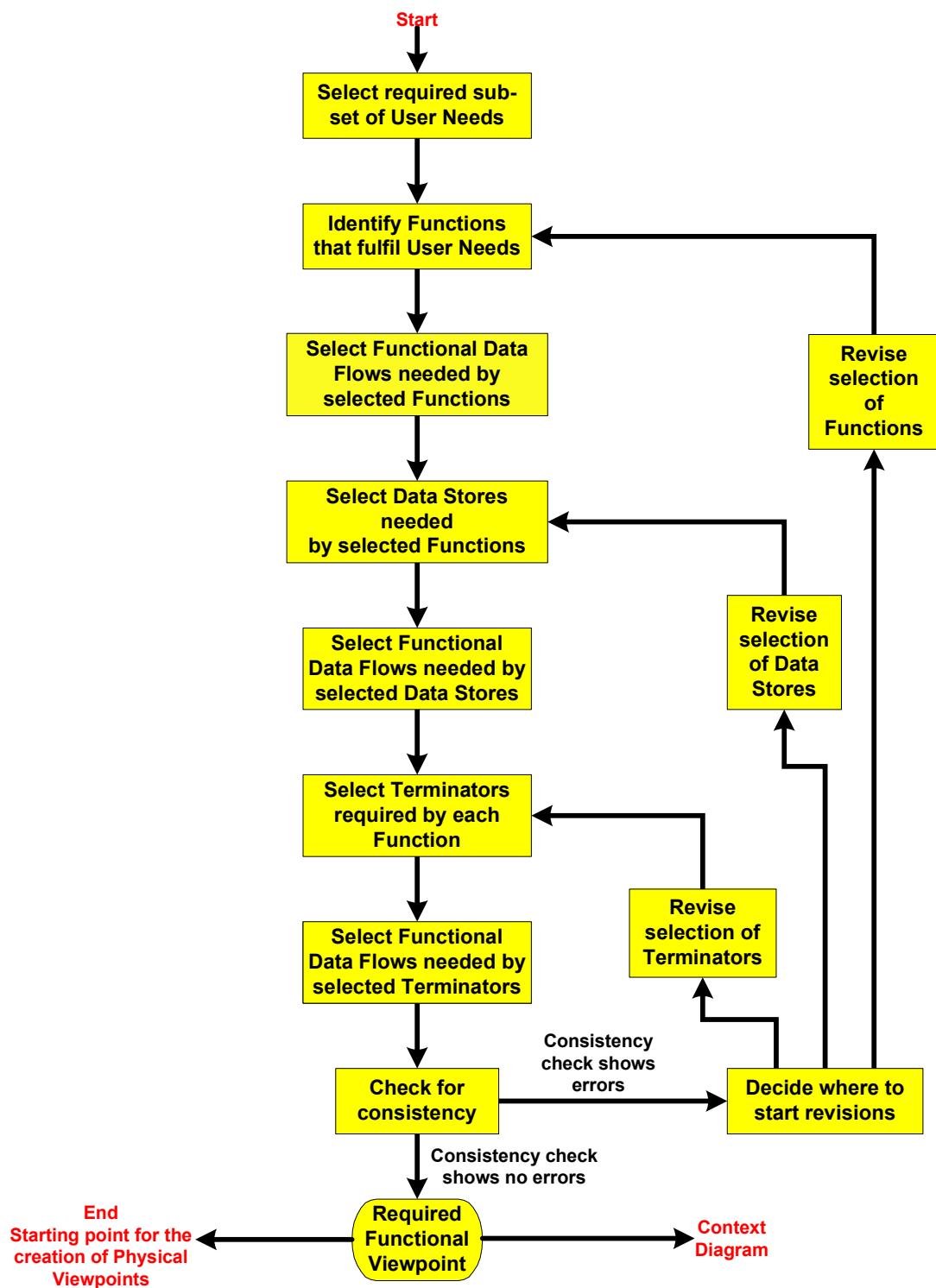
3.2.2 Creating the Functional Viewpoint

The basic steps involved in the process of creating a Functional Viewpoint can be summarised as the following:

- (a) Prepare the Stakeholder Aspirations that will express in the Stakeholders' own words the services that the ITS implementation is expected to support. This should be done through consultation and meetings with Stakeholders so that their aspirations for the future ITS implementation can be obtained.
- (b) Select a name for the ITS Architecture that will define the functionality needed to implement the services that the Stakeholders have described.
- (c) Identify and select the User Needs that define the service(s) described in the Stakeholder Aspirations. This step is often described as "mapping the Stakeholder Aspirations to the User Needs".
- (d) Select all of the Functions that support the User Needs.
- (e) Confirm that the selected Functions are reasonable, removing any from the selection that it is considered are not wanted as part of the ITS implementation. Their functionality will have to be provided by other means outside the ITS Architecture.
- (f) Select the Data Flows needed by the selected Functions.
- (g) Select the Data Stores needed by the selected Data Flows.
- (h) Select the additional Data Flows needed by the selected Data Stores.
- (i) Identify the Terminators associated with all these Data Flows.
- (j) Select the Data Flows needed by the Terminators
- (k) Check that the selection of Functions, Data Stores, Terminators and their Data Flows is consistent. If not revise the selections previously made until consistency is achieved.
- (l) Produce outputs from which the Context Diagram can be created.

With the exception of steps (a) and (c) all of the other steps can be carried out using the FRAME Selection Tool. A diagram of the way in which these steps relate to each other is shown in Figure 3.

Inconsistencies will usually appear in the form of Data Flows that have no source or target, or Functions that have only one Data Flow. The latter can be accepted, but the former have to be resolved by revising the selection of Data Flows. This can only be done by going back to the selection of Functions, Data Stores or Terminators whose Data Flows need to be revised. So when inconsistencies are found, the first thing to do is to decide which of the three entities are affected. If necessary the actual selection can remain unaltered and only their Data Flows revised.

Figure 3 – Creation of the Functional Viewpoint

Note that the "Required Functional Viewpoint" will usually contain a sub-set of the functionality that is in the Functional Viewpoint in the FRAME Architecture. However it is also possible to create a "super-set", i.e. to add extra functionality. This is discussed later in this Reference Manual.

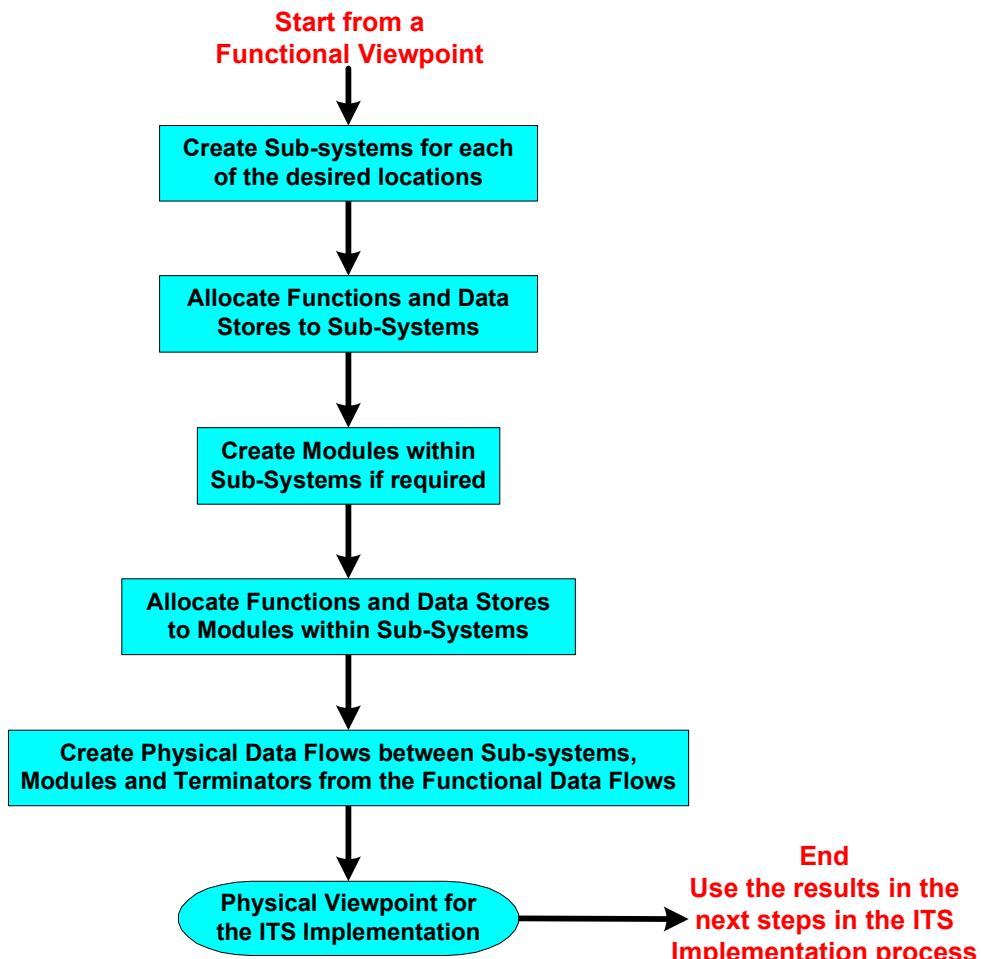
3.2.3 Creating the Physical Viewpoint

The starting point from which to create a Physical Viewpoint is a Functional Viewpoint created using the method described in the previous section. Its creation follows the following steps:

- (a) If there is a choice, select the Functional Viewpoint to use as the starting point for the creation process.
- (b) Provide a name for the Physical Viewpoint.
- (c) Identify each of the Sub-systems and their locations.
- (d) Allocate Functions and Data Stores to the Sub-Systems.
- (e) Where desirable, create Modules within Sub-systems.
- (f) Allocate Functions and Data Stores to Modules within the Sub-Systems.
- (g) Produce reports on the contents of the Physical Viewpoint.

All of the above steps can be carried out using the FRAME Selection Tool. A diagram of the way in which these steps relate to each other is shown in Figure 4.

Figure 4 – Creation of the Physical Viewpoint



The creation of the Physical Data Flows is done automatically within the FRAME Selection Tool once all the Sub-systems and Modules have been created and

Functions assigned to them. However the Tool gives them numbers that may need to be changed to more meaningful names.

The Physical Viewpoint that is produced at the end of these steps provides the starting point for the stage in the ITS implementation process. It also provides one of the starting points for the creation of the other outputs that can be produced from an ITS architecture. These are discussed further in a later part of this Chapter.

3.3 *Advantages and facilities provided by the FRAME Selection Tool*

The process of creating an ITS architecture just described, is quite demanding and time-consuming, especially if there is only paper documentation to rely on. The task is made a little easier using the FRAME Browsing Tool, which provides hyper-linked access to the components of the European ITS Functional Viewpoint via an Internet Browser. But the situation is completely different if it is possible to exploit the support of an electronic assistant in the form of the FRAME Selection Tool.

However the FRAME Selection Tool does not do all the work. The design choices are always left to the ITS architect, who is released from most of the routine activities such as, finding the description of a specific Function, Data Flow, Data Store, Terminator or Actor to decide whether it has to be included or not, and determining which Data Flows are associated with a specific set of Functions, etc. Thus the Selection tool will assist in the creation of a logically consistent ITS architecture, but not a semantically consistent one.

The FRAME Selection Tool provides facilities for the creation of tables that show the contents of both the Functional and Physical Viewpoints. These tables can be produced by the Tool in Comma Separated Variable (CSV) format that can be read by Microsoft® Excel® for further processing. Using a suitable tool, such as Microsoft® VISIO®, it is then possible to draw such things as the Context Diagram plus diagrams that will illustrate the Physical Viewpoint. (Note that the Selection Tool does not have any drawing facilities of its own at the moment.)

Once the FRAME Selection Tool has been used to create a Functional Viewpoint, it is expected that some users may wish to add additional elements (e.g. User Needs, Functions, Data Flows, etc.) in order to create a full national, regional or service architecture. These facilities are available by making additions directly to the Microsoft® Access® database (the FRAME Database) that is used by the FRAME Selection Tool. The modified Database can then be used by the FRAME Selection Tool to create the required ITS architecture in exactly the same way as for the original Database. The use of these facilities is described in Part 3 of this Reference Manual.

3.4 *Using the FRAME Browsing Tool to support the FRAME Selection Tool*

As noted at the start of this Chapter, although it consists of only two components, the FRAME Architecture is very large. It contains User Needs and the functionality to fulfil them for most of the services that can be supported by a road-based ITS implementation. This can make it difficult to decide exactly which parts of the FRAME Architecture to include in any ITS architecture that is created from it.

The FRAME Browsing Tool can provide a solution to this difficulty. It enables any particular part of the FRAME Architecture to be studied in detail in order to see whether or not its User Needs and functionality will be relevant to the services that are to be provided by the ITS implementation for which an ITS architecture is to be created with the FRAME Selection Tool.

The FRAME Browsing Tool can be particularly useful when fixing errors and warnings that have resulted from the selection of User Needs and functionality using the FRAME Selection Tool. This is illustrated in section 6.9.7 of this Reference Manual.

Part 2 Using the FRAME Selection Tool

This part of the Reference Manual provides a guide to the use of the FRAME Selection Tool. It part covers everything from downloading the Tool and the FRAME Database from the FRAME website, through the Viewpoint creation process, to producing the reports showing the contents of the Functional and Physical Viewpoints that the Tool produces. The involvement of the FRAME Browsing Tool in the use of the Selection Tool is also covered.

It contains the following seven Chapters:

- Chapter 4: Obtaining the FRAME Tools
- Chapter 5: Using the FRAME Selection Tool – initial start-up
- Chapter 6: Using the FRAME Selection Tool – the Creation of a Functional Viewpoint
- Chapter 7: Using the FRAME Selection Tool – the Creation of a Physical Viewpoint
- Chapter 8: Creation of Reports showing the results from the creation process
- Chapter 9: Modifying existing Viewpoints
- Chapter 10: Hints and Tips for using the FRAME Selection Tool

4. Obtaining the FRAME Tools

4.1 *Introduction*

This Chapter describes how to obtain a copy of the FRAME Selection Tool for use in creating ITS architectures. A section showing how to obtain the FRAME Browsing Tool is also included is also included in case it is needed.

4.2 *Obtaining the FRAME Selection Tool*

The FRAME Selection Tool is available for downloading from the FRAME website, which can be found at: <http://www.frame-online.net/>. There is a part of the website that contains the files that need to be downloaded. These comprise the following:

- (1) FRAME Selection Tool: this contains the actual tool itself;
- (2) FRAME Database: this contains the Microsoft® Access® database that will be used by the Selection Tool;
- (3) Patch: this contains a set of Microsoft® Windows® operating system files that are used by the Tool and which are not always present on every computer.
- (4) User Reference Manual: additional and updated copies of this Manual.

Information about downloading these files and for installing the Tool, the Database and the patch files are described in website page. Note that once the files have been downloaded for the first time, it is not necessary to download all of them again. It should only be necessary to download those that have changed.

The choice of folders into which these files are downloaded is entirely at the user's discretion. However it recommended that the Database is put into its own folder and a copy made of the Database file. The copy should then be chosen when the Selection Tool asks for the Database to be specified. This will preserve a copy of the original Database so that it does not need to be downloaded again.

4.3 *Obtaining the FRAME Browsing Tool*

The FRAME Browsing Tool is also available for downloading from its own part of the FRAME website. Again it is downloaded as a zipped file, from which the files that comprise the actual Tool need to be extracted. Advice on how to set up the Tool is also available on the website.

There is also a brief guide on how to use the FRAME Browsing Tool on the website. This is very straight forward as the Tool uses a "standard" Internet browser such Internet Explorer or Netscape Navigator.

5. Using the FRAME Selection Tool – initial start-up

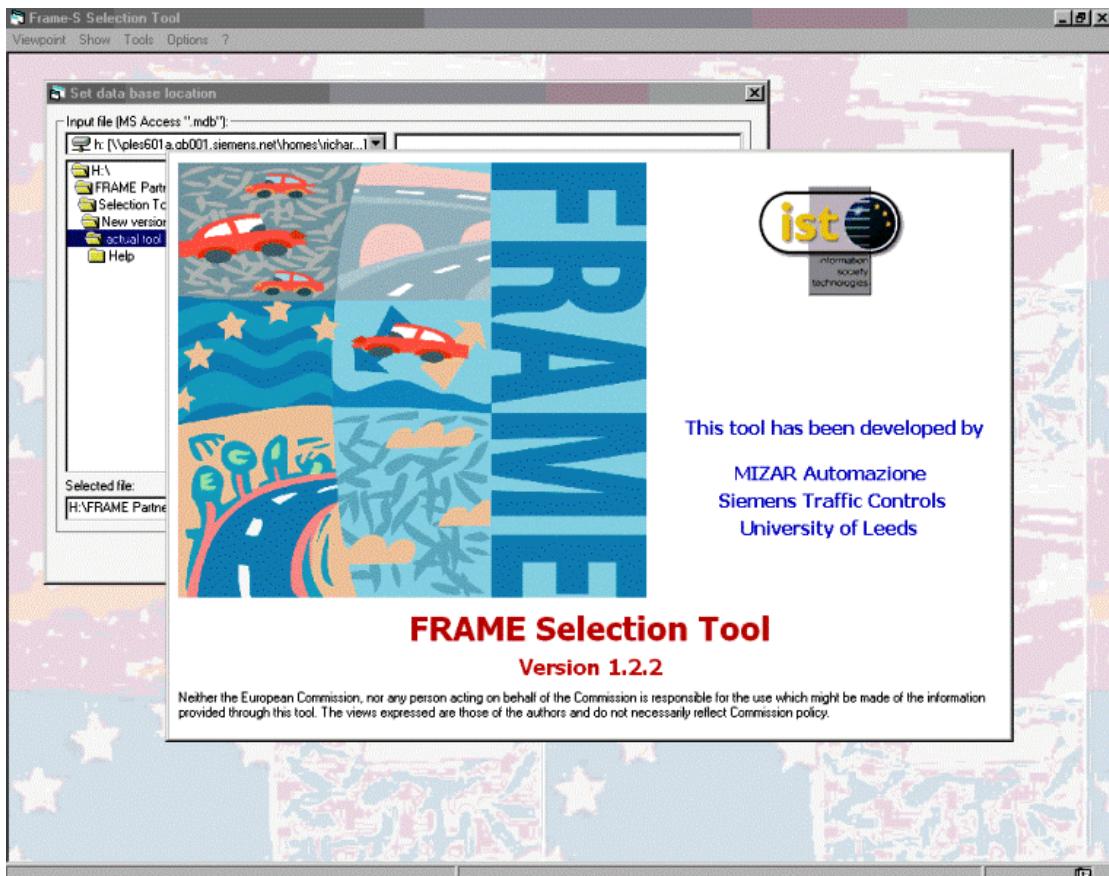
5.1 *Introduction*

This Chapter describes how to start-up the FRAME Selection Tool and select the Database that contains the FRAME Architecture, or a previously created Viewpoint. This is the same for work on either a Functional or Physical Viewpoint.

5.2 *Running the Tool*

When the FRAME Selection Tool is run the screen shown in Figure 5 appears. The "splash" in the middle of the screen will disappear after a few seconds and leave the Database selection screen.

Figure 5 – Selection Tool Opening screen

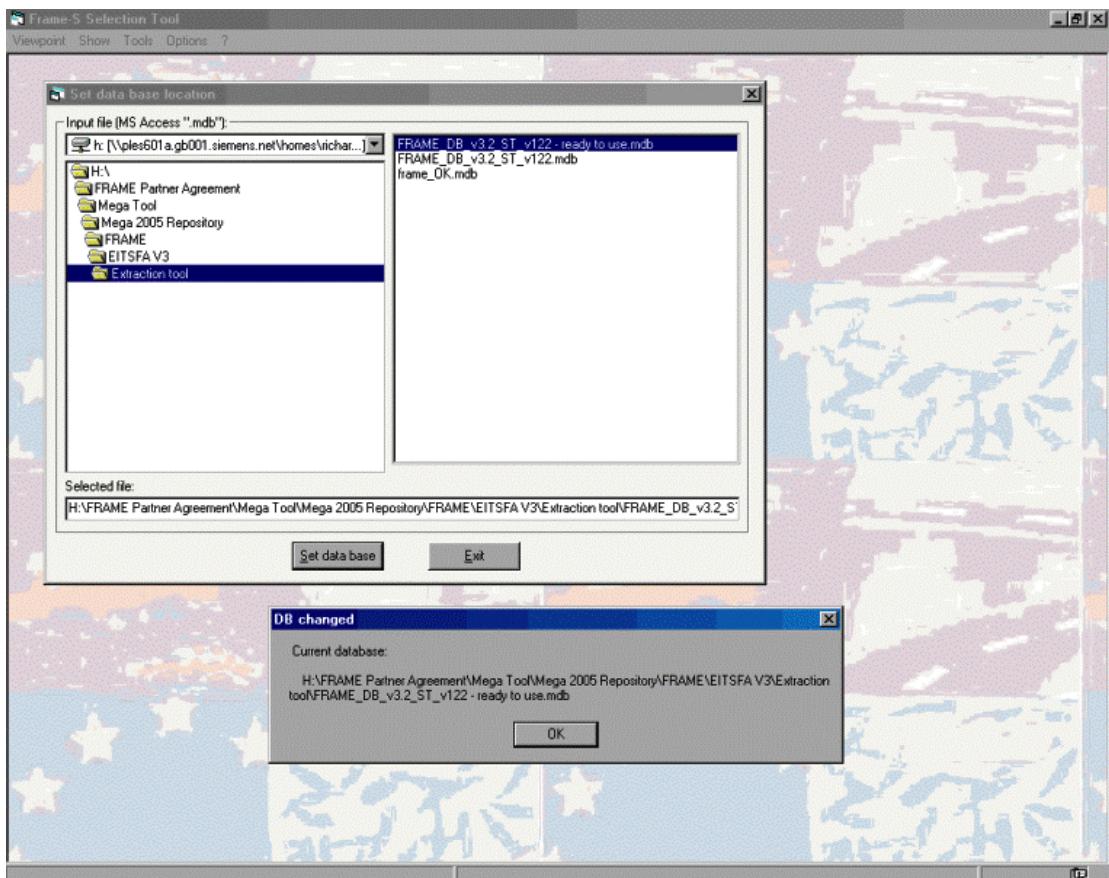


5.3 *Selecting the Database*

The first proper screen (see Figure 6) enables the user to select the FRAME Database that the Selection Tool is to use. The panel on the left of the "Set Database Location" window is used to select the folder in which the Database is located. Initially it will point to the folder in which the Tool was stored when it was downloaded from the FRAME website – see section 4.2. After that it will point to the last database used by the Tool.

Changing to a different folder is achieved by double clicking on a higher level folder to display its sub-folders. If necessary the double clicking is repeated for one of the sub-folders until a folder containing the required Database is found. The right hand panel of the window will remain blank until the folder that has been selected contains one or more Databases (MDB files). When this happens, the right hand panel will display a list of any Databases that are in the folder. Selecting one by clicking on it will cause its name to be displayed in the "Selected file" panel and produce the confirmation window shown at the bottom of Figure 6.

Figure 6 – Database Selection screen with result



Once the "OK" button is pressed, the "Select Database Location" window will disappear. It is then possible to select the starting point for the work to be done to creation the particular ITS architecture.

5.4 Starting point for the use of the Selection Tool

The starting point for the use of the Selection Tool will depend on the state of the Database that has just been opened and the work that has to be done. There are two options: work on a Functional Viewpoint or on a Physical Viewpoint. These two options are revealed by clicking on "Viewpoint" drop down menu at the top left of the screen. It is also possible to quit ("Exit") from the Tool at this point or to use this same option at a later time. The option to quit is also provided explicitly on many other user screens, the use of which is summarised in section 5.5.

The options displayed when either "Functional" or "Physical" is selected from the "Viewpoint" are the same. They are shown in Figure 7 and their selection has the following results:

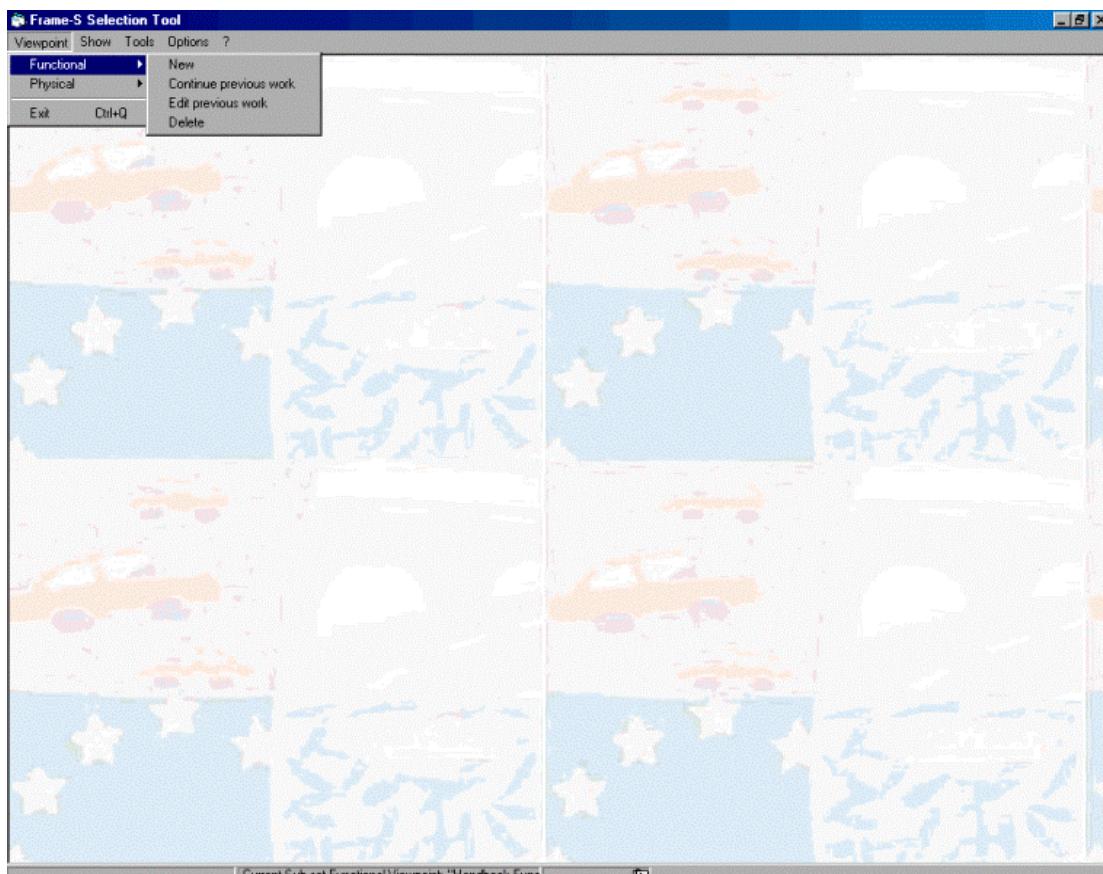
"New": means start the creation of a new Functional or Physical Viewpoint. This is discussed further in section 6 (Functional Viewpoint) and section Physical Viewpoint). The work of creating an ITS architecture begins with the creation of the Functional Viewpoint, and when that is complete and consistent, it may be used as the basis for one or more Physical Viewpoints. This follows the process that was described in Chapter 3. Once a Functional Viewpoint has been used to create a Physical Viewpoint it is no longer possible to change that Functional Viewpoint. Instead, any attempt to do so will result in a window that asks whether a "clone" (i.e. a copy) should be made of the Functional Viewpoint.

"Continue previous work": enables work to continue on the creation of a Functional or Physical Viewpoint that is only part completed. Selection of this option will automatically take the user to the point in the creation process at which the work was previously stopped – see also section 5.5. Note that this only applies to the first pass through the creation process.

"Edit previous work": enables a previously completed Functional or Physical Viewpoint to be changed. However it is not possible to change a Functional Viewpoint if it has already been used as the starting point for a Physical Viewpoint – see section 9.3.

"Delete": enables the deletion of a previously created Functional or Physical Viewpoint, regardless of whether it has been completed or not.

Figure 7 – Viewpoint drop down menu



From this point onwards, the way that the Tool is used depends on which type of Viewpoint is to be created, or worked on. The following Chapters describe how the Tool is used for work on either Functional or Physical Viewpoints.

5.5 *Exiting from the Selection Tool*

It is possible to exit from the Selection Tool at any point in either the Functional or Physical Viewpoint creation processes. The "exit" option appears in the drop down menu under "Viewpoint" at the top of the screen – see Figure 7.

If the "Exit" option is "clicked" the Selection Tool will produce a screen asking the user to confirm that they want to exit by clicking on the "Yes" button. Clicking on the "No" button will cause the Tool to continue with the part of the creation process that it had reached.

If the "Yes" button is "clicked" for the "Exit" option, the Selection Tool will ask if the current work is to be saved. Again a "Yes" button is provided to confirm this – clicking on "No" will mean that all the work will be lost on the currently open Functional or Physical Viewpoint up to the last point at which the "OK" button was "clicked". Clicking on the "Yes" button will cause the Selection Tool to save the work up to the completion of the previous step in the Functional or Physical Viewpoint creation processes. When the Tool is run again, selecting "Continue with previous work" will cause it to go to the screen from which "Exit" was selected. Thus the Tool will behave in a similar way to when the "Quit" button is "clicked", as described in sections 6.11 and 7.8.

Note that if the current work is to be completely discarded the only way to do this is to select the "Edit Previous Work" option from the "Viewpoint" drop down menu. This will allow everything that has been selected to be deleted or revised. Alternatively the Functional or Physical Viewpoint containing the work should be deleted and the creation process started again using a new Functional or Physical Viewpoint.

6. Using the FRAME Selection Tool – the Creation of a Functional Viewpoint

6.1 *Introduction*

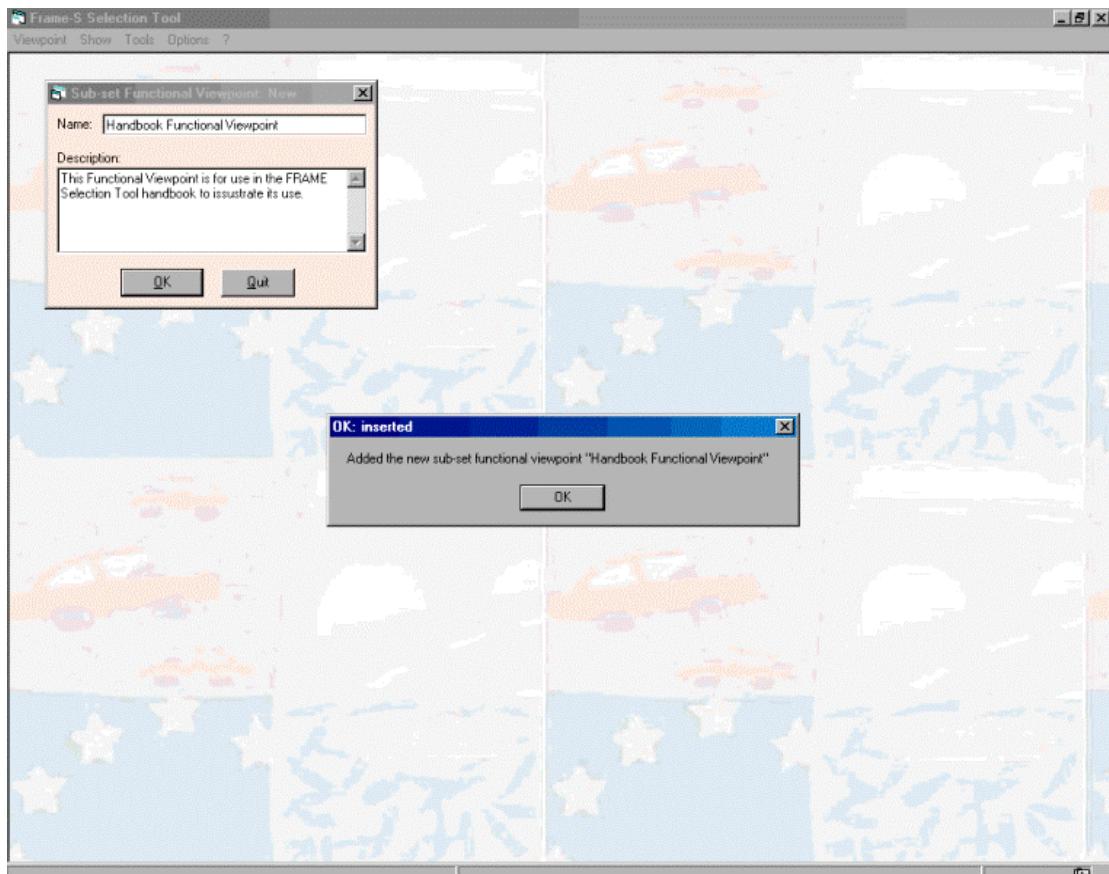
This section describes the way that the Selection Tool is used to create a new Functional Viewpoint or modify an existing one. It follows the process described in section 3.2.2.

6.2 *Step 1 – naming the Functional Viewpoint*

The first step in the process of creating a new Functional Viewpoint is to give it a name. This is done using the screen shown in Figure 8 which appears automatically when the "New" option is selected for a Functional Viewpoint using the menu shown in Figure 7. Although a name has to be created, its actual form and content is entirely up to the user. However providing a description is entirely optional.

Once the name (and the optional description) is complete the "OK" button should be pressed. The confirmation screen shown in Figure 8 will then be produced, unless the chosen name has already been used. In this situation, the user will be asked to choose another name.

Figure 8 – Naming the Functional Viewpoint

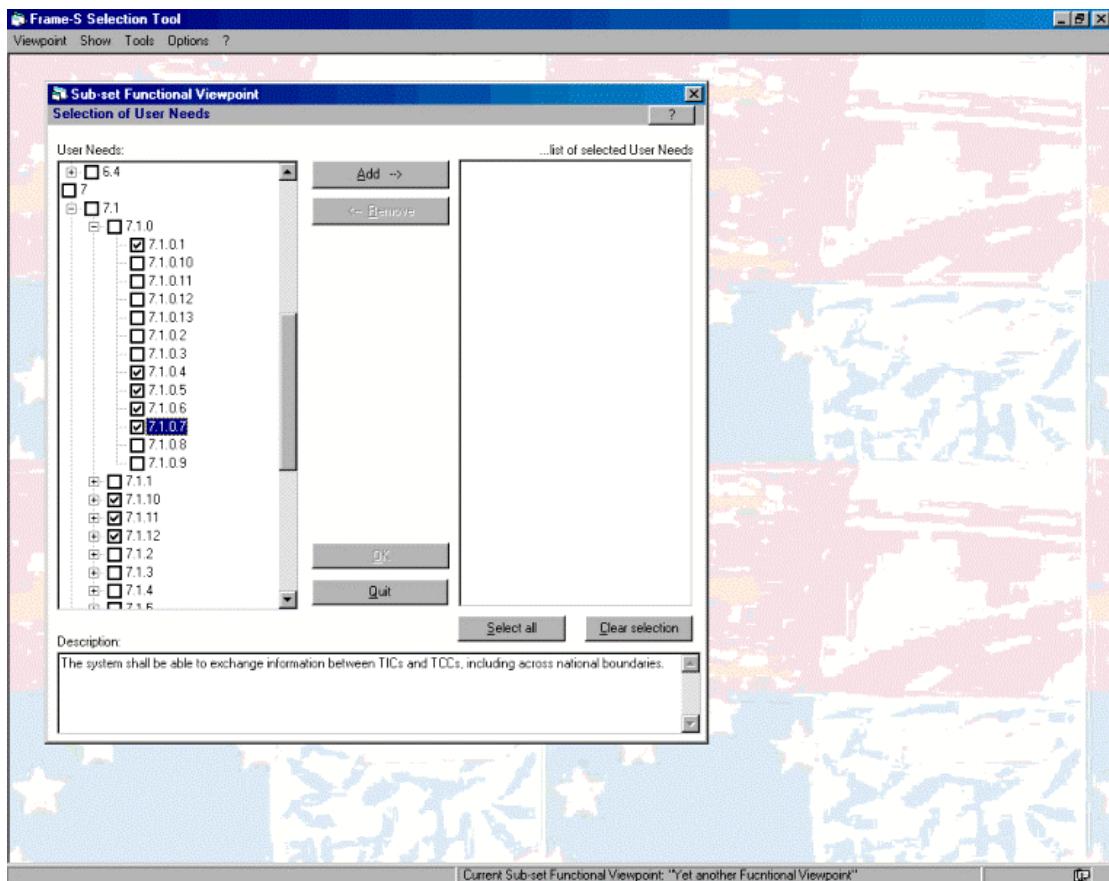


6.3 Step 2 – selecting the User Needs

Once the name of the new Functional Viewpoint has been defined, the next step in its creation is to select that User Needs that it will serve. This is achieved using the screen shown in Figure 9.

The way in which the structure of the User Needs is portrayed by the Selection Tool is shown in the left-hand part of the "Selection of User Needs" window in Figure 9. Clicking on a "+" symbol within any User Need "Group" will cause the expansion of its "Service" or "Topic", as has been shown for "Group" 7. Note that the "+" symbol becomes a "-" symbol where the expansion has been made. The description of a "Group", "Service", "Topic" or User Need is automatically displayed in the "Description" window part of the "Selection of User Needs" window by clicking on its number. Particular User Needs can be selected by clicking each of the "square" boxes adjacent to their numbers, which will cause a "✓" to appear in them, as shown for the User Needs in Topic 7.1.0. If all the User Needs in a particular "Group", "Service" or "Topic" are to be selected, its "square" box should be clicked as shown for "Topics" 7.1.10, 7.1.11 and 7.1.12.

Figure 9 – Selecting User Needs



When all the required User Needs have been selected, they are included in the Viewpoint by clicking on the "Add" button. This will cause them to appear in the right-hand part of the "Selection of User Needs" window in Figure 9 and to disappear from the left-hand window.

If at this stage the user decides that the wrong User Needs have been selected, then they can be de-selected by clicking the "square" boxes adjacent to their numbers in

the right-hand part of the "Selection of User Needs" window. They are then actually de-selected by clicking on the "Remove" button. All of the selected User Needs can be de-selected by clicking on the "Select all" button and then the "Remove" button. Alternatively the actual User Needs to be de-selected can be cleared by clicking on the "Clear selection" button so that a new selection can be made.

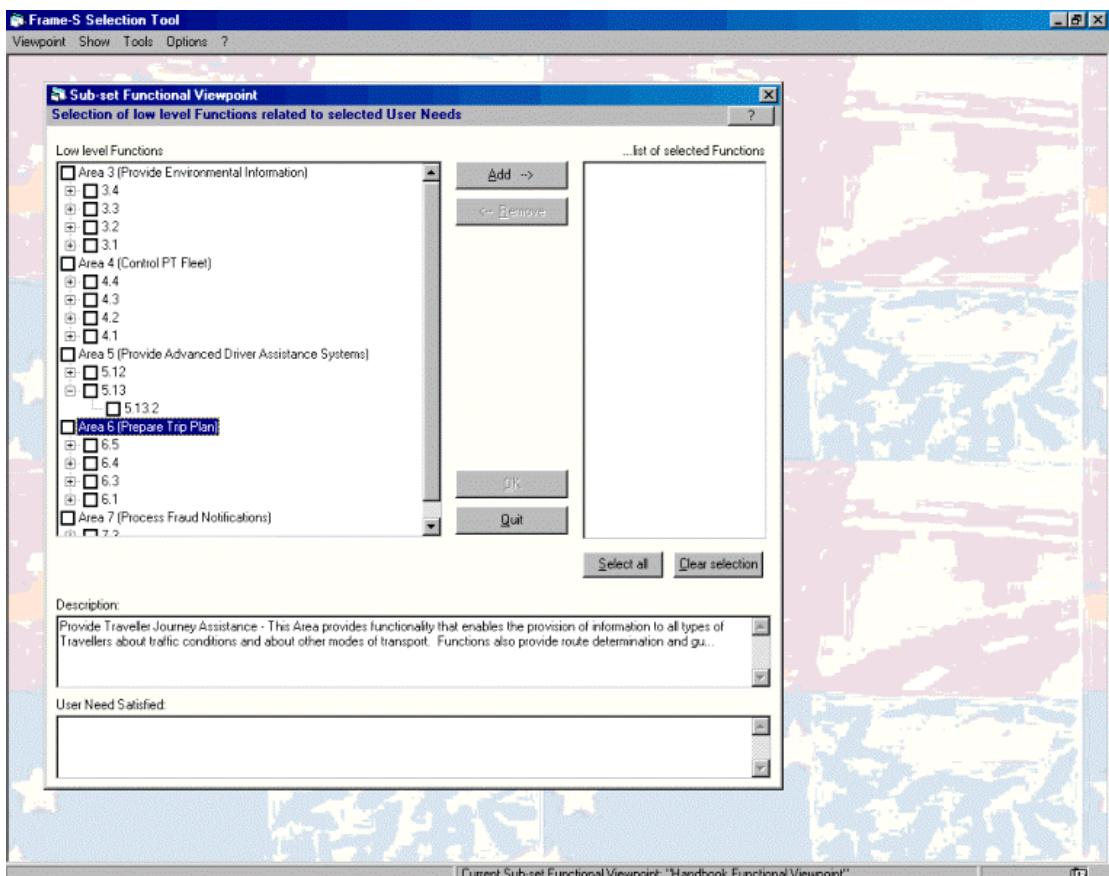
When the selection is complete it must be confirmed by clicking on the "OK" button at the bottom of the middle of the screen. This will also move the Selection Tool on to the next step in the Functional Viewpoint creation process.

6.4 Step 3 - selecting the Functions

Once the "OK" button is "clicked" in Figure 9 the screen shown in Figure 10 will appear automatically. This will take the user to the next step in the Functional Viewpoint creation process in which the Functions are selected.

In the left-hand part of the "Selection of low level Functions..." window all of the Functions that are served by the previously selected User Needs are shown. Initially the corresponding Functional Areas and the names of the top high-level Functions are also shown. The actual low-level Functions can be revealed by clicking on the "+" symbol next to the Functional Area or top high-level Function number as illustrated for Function 5.13.2.

Figure 10 – Selecting Low-level Functions



Note that this window behaves differently the first time it is shown when creating a Functional Viewpoint to all the other occasions that it is requested. On the first occasion **only** those Functions that have been designated as directly satisfying the

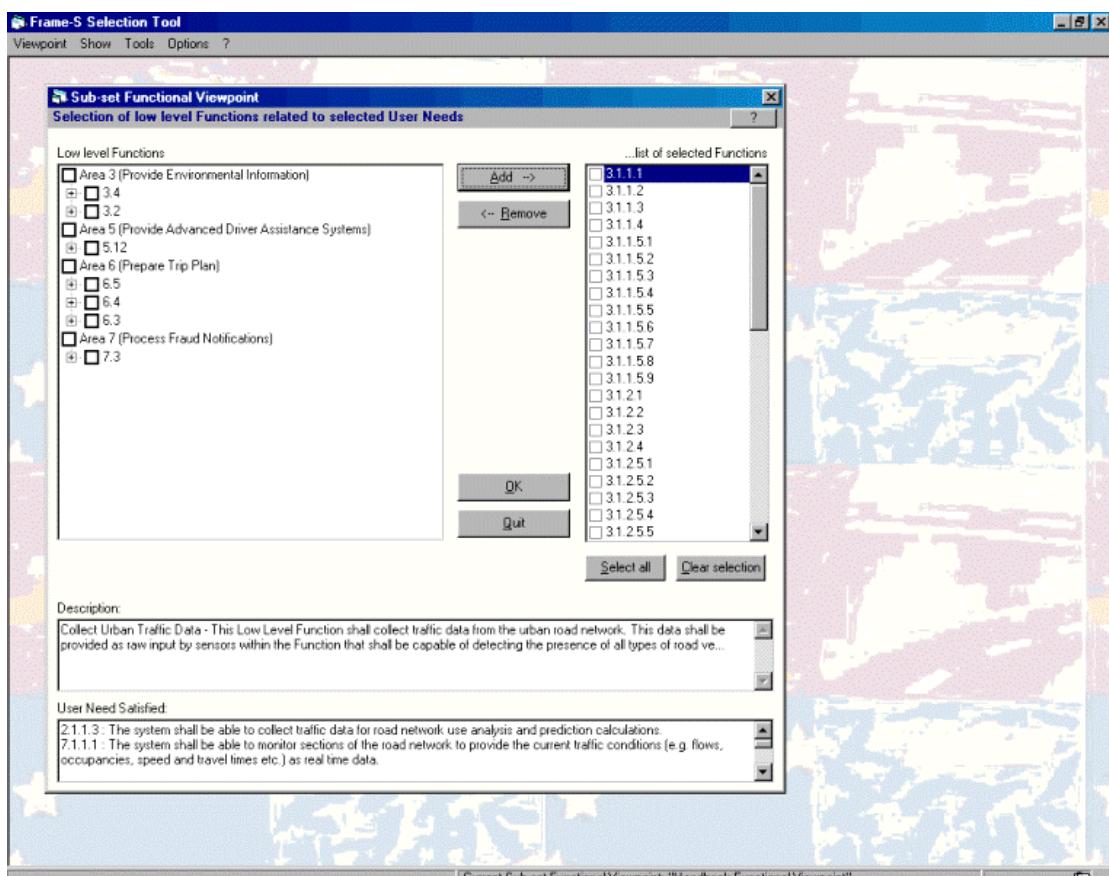
selected User Needs are available – the primary Functions. Even these Functions need to be selected carefully because many User Needs can have a number of interpretations and/or features.

On all other occasions that this window is displayed the full set of Functions is available for selection. Some of these will be needed to support the initial selection of primary Functions, e.g. to provide them with their required data, and to produce a full consistent Functional Viewpoint – see section 6.9.

The Overview description of a Functional Area, top high-level Function, or low-level Function can be automatically displayed in the "Description" window part of the "Selection of low level Functions..." window by clicking on its number. Only when a low-level Function number is "clicked", will the User Needs that are partially or completely satisfied by it be also displayed in the "User Need Satisfied" window.

Selecting particular Functions for inclusion in the Viewpoint is achieved by clicking on the "square" boxes adjacent to their numbers so that a "✓" symbol appears in each box. When all of the required Functions have been selected they must be included in the Viewpoint by clicking on the "Add" button. This will cause them to move to the right-hand part of the "Selection of low level Functions..." window in Figure 11. At the same time the selected Functions will completely disappear from the left-hand window, leaving only those Functions that have not been selected.

Figure 11 – Adding the selected low-level Functions to the Functional Viewpoint



At this point the list of selected Functions is not final, and additional Functions can be added and/or deleted at a later stage. Alternatively the selection of Functions can be

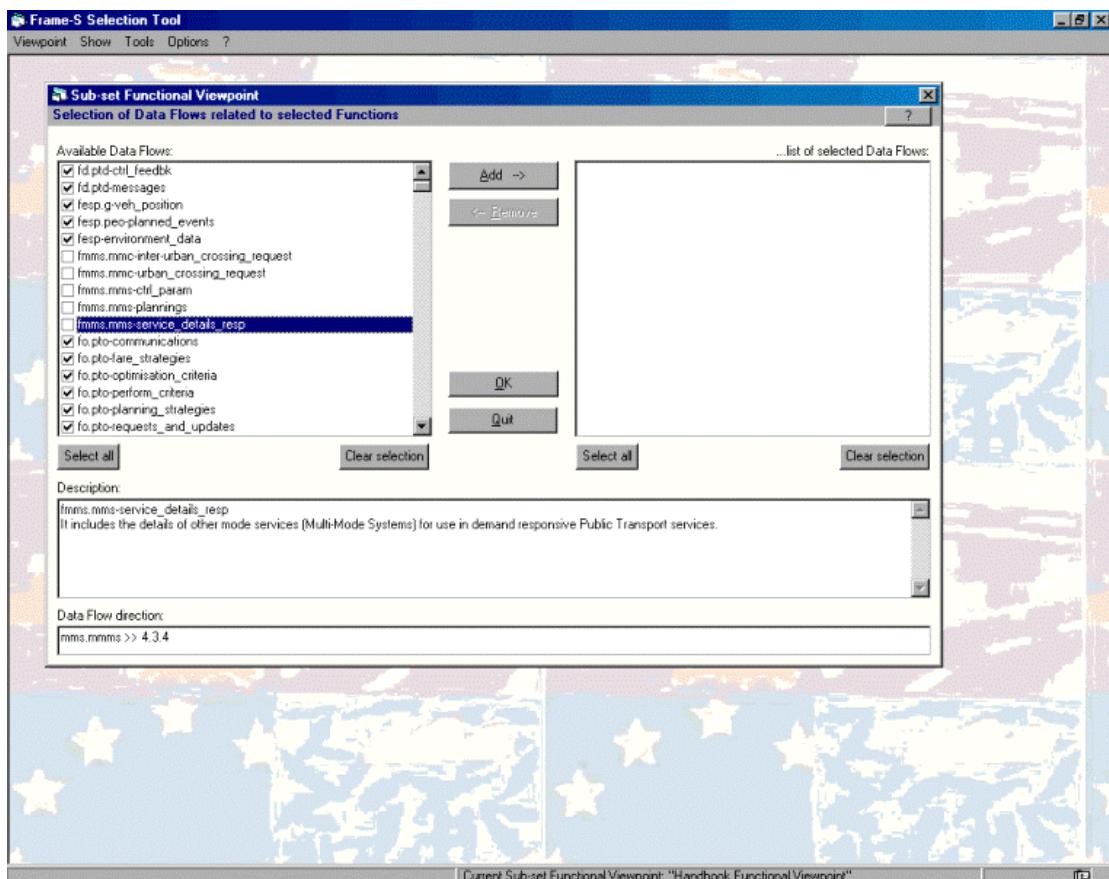
changed at this stage. Clicking the "square" boxes adjacent to the Function numbers in the right-hand part of the "Selection of low level Functions..." window will mark them for de-selection. To de-select all of the selected Functions click on the "Select all" button. De-selection is then achieved by clicking on the "Remove" button. If a Function is incorrectly identified for de-selection, this can be cleared by clicking on its "square" box again. All of the identified Functions can be removed from the de-selection list by clicking on the "Clear selection" button.

When the selection of Functions is complete it has to be confirmed by clicking on the "OK" button. This will also move the Selection Tool on to the next step in the Functional Viewpoint creation process.

6.5 Step 4 – selecting the Data Flows for the Functions

Once the "OK" button is "clicked" in Figure 11 the screen shown in Figure 12 will appear automatically. This is the screen through which the Data Flows to be used by the Functions are selected.

Figure 12 – Selecting Data Flows



In the left-hand part of the "Selection of Data Flows..." window **all** the Functional Data Flows that provide either inputs or outputs to the selected Functions are displayed. The display will therefore include not just Data Flows between the selected Functions, but those between the selected Functions and the remaining unselected Functions, plus those between the selected Functions and all of the Terminators and Data Stores with which they exchange data;

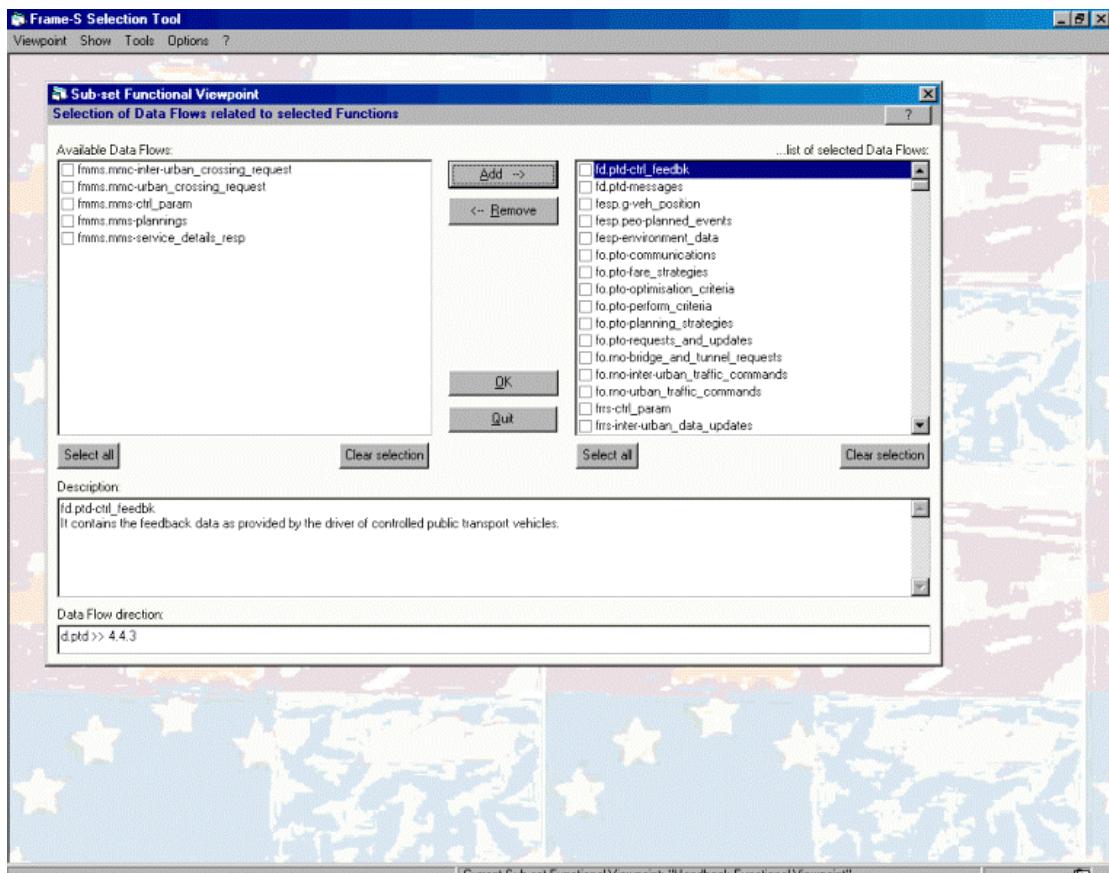
The Data Flows are listed by their names in alphabetical order. Thus those Data Flows from Terminators appear first (their names always begin with the letter "f") and those to Terminators appear last (their names always begin with the letter "t"). All the other Data Flows will be shown in between, with only the mnemonic at the start of the name to show if they are between Functions in the same Functional Area, or between Functional Areas.

The name and description of a Data Flow is automatically displayed in the "Description" window part of the "Selection of Data Flows..." window by clicking on its name. This will also display the identities of the source and destination Functions, Terminators or Data Stores in the "Data Flow direction" window – see Figure 12.

Selecting the Data Flows to be included in the Viewpoint is achieved by clicking on the "square" boxes next to their names causing a "✓" symbol to appear in each box, as shown in Figure 12. Alternatively, all of the Data Flows can be selected by clicking on the left-hand "Select All" button. The selection can be removed by clicking on the left-hand "Clear All" button enabling a new selection to be made.

When all of the required Data Flows have been selected they must be included in the Viewpoint by clicking on the "Add" button. This will cause them to move to the right-hand part of the "Selection of Data Flows..." window as shown in Figure 13. At the same time the selected Data Flows will disappear from the left-hand window, leaving it displaying only those Data Flows that have not been selected.

Figure 13 – Adding the selected Data Flows to the Functional Viewpoint



If the user is unhappy with the selected Data Flows they can be de-selected by clicking the "square" boxes next to their names in the right-hand part of the "Selection

of Data Flows..." window. All of the selected Data Flows can be de-selected by clicking on the right-hand "Select all" button. If a Data Flow is incorrectly identified for de-selection, this can be cleared by clicking on its "square" box again. All of the identified Data Flows can be removed from the de-selection list by clicking on the "Clear selection" button. Data Flows are actually de-selected by clicking on the right-hand "Remove" button.

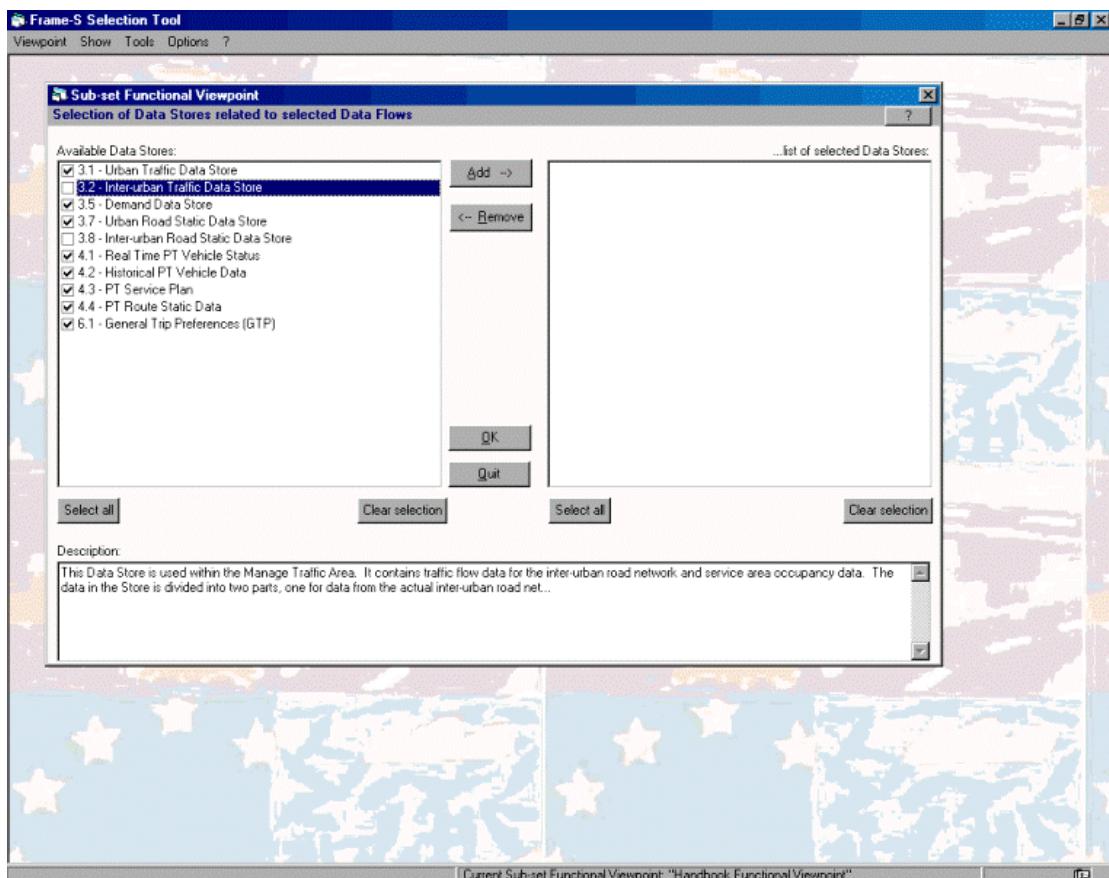
In order to make a reasonably (but not perfect) selection of the required Data Flows, it is recommended that the FRAME Browsing Tool is activated. The Tool can be used to ascertain how each Function behaves, i.e. to identify the Data Flows needed to provide its inputs and those used to provide its outputs. The required information can be found in the FRAME Browsing Tool by looking at a combination of the Data Flow Diagrams, the Function descriptions and the Data Flow descriptions. The appropriate Data Flows can then be selected based on the Functions that have been selected for a particular Functional Viewpoint. An illustration of a typical output from the Browsing Tool is shown in Figure 22 with details in the surrounding section 6.9.7.

When the user is happy with the selection of Data Flows it has to be confirmed by clicking on the "OK" button. This will also move the Selection Tool on to the next step in the Functional Viewpoint creation process.

6.6 Step 5 – selecting the Data Stores used by the Functions

Once the "OK" button is "clicked" in Figure 13 the screen shown in Figure 14 will appear automatically. This is the screen through which the Data Stores to be used by the selected Functions are identified.

Figure 14 – Selecting Data Stores

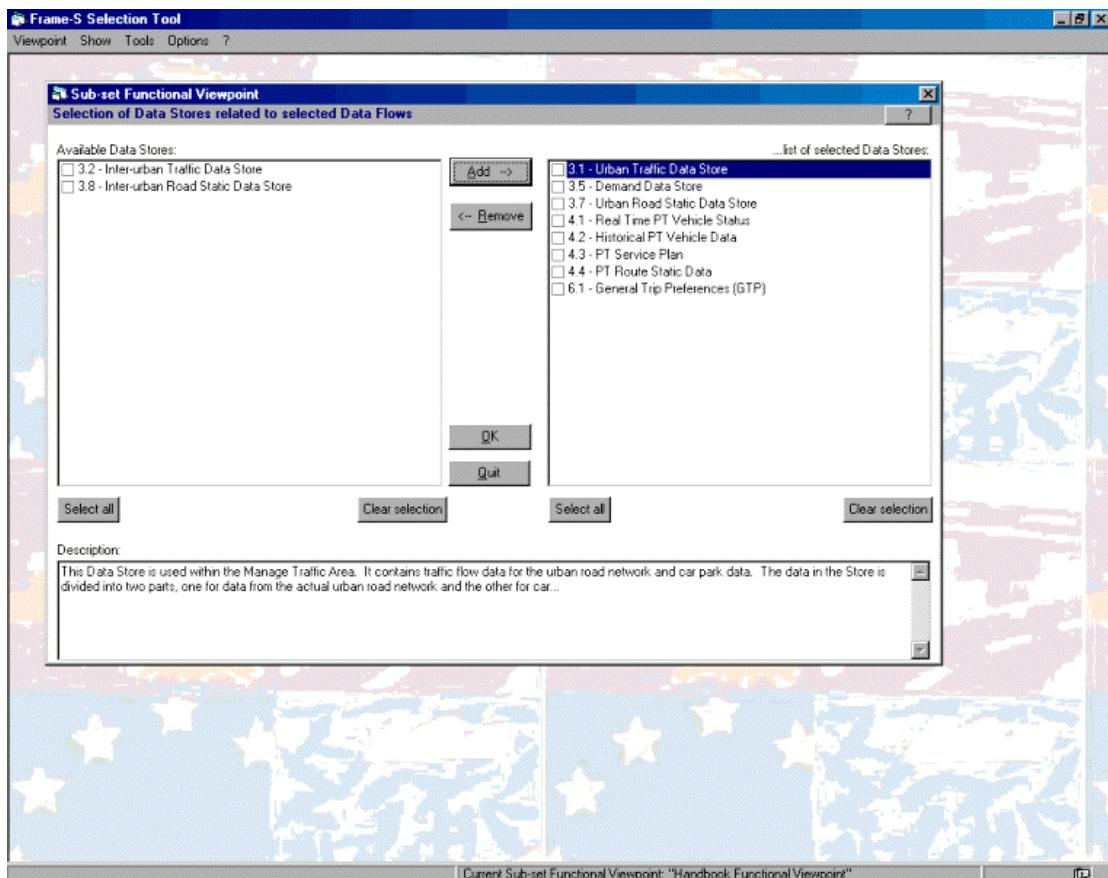


In the left-hand part of the "Selection of Data Stores..." window all of the Data Stores that are used by the previously selected Functions are displayed. The Data Stores are listed in numerical order. The description of a Data Store is automatically displayed in the "Description" window part of the "Selection of Data Stores..." window by clicking on its name, as shown in Figure 14 for Data Store 3.2.

Selecting particular Data Stores for inclusion in the Viewpoint is achieved by clicking on the "square" boxes adjacent to their numbers so that a "✓" symbol appears in each box, as shown in Figure 14. Alternatively, all of the Data Flows can be selected by clicking on the left-hand "Select All" button. At this stage the selection can be amended by clicking on the left-hand "Clear All" button and making a new selection.

When all of the required Data Stores have been selected they must be included in the Viewpoint by clicking on the "Add" button. This will cause them to move to the right-hand part of the "Selection of Data Stores..." window as shown in Figure 15. At the same time the selected Data Stores will disappear from the left-hand window, leaving it to display only those Data Stores that have not been selected.

Figure 15 – Adding the selected Data Stores to the Functional Viewpoint



If at this stage the user decides that the wrong Data Stores have been selected, then they can be de-selected by clicking the "square" boxes adjacent to their numbers in the right-hand part of the "Selection of Data Stores..." window. All of the selected Data Stores can be de-selected by clicking on the right-hand "Select all" button. If a Data Store is incorrectly identified for de-selection, this can be cleared by clicking on its "square" box again. All of the identified Data Stores can be removed from the de-selection list by clicking on the "Clear selection" button. Data Stores are then actually de-selected by clicking on the right-hand "Remove" button.

When the selection of Data Stores is complete it has to be confirmed by clicking on the "OK" button. This will also move the Selection Tool on to the next step in the Functional Viewpoint creation process.

6.7 Step 6 – selecting the Data Flows used by the Data Stores

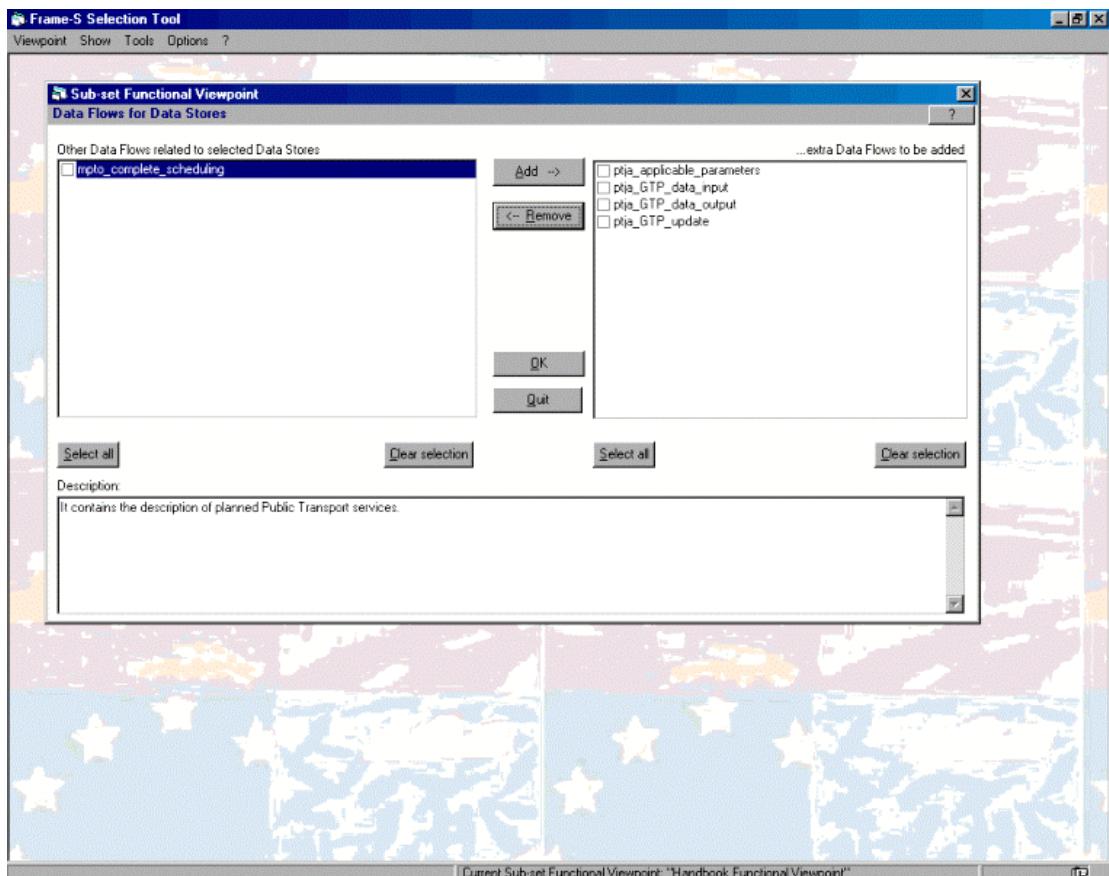
Once the "OK" button is "clicked" in Figure 15 the screen shown in Figure 16 will appear automatically. This is the screen through which the Data Flows to be used by the Data Stores are selected.

In the left-hand part of the "Selection of Data Flows..." window all of the Data Flows that are used by the previously selected Data Stores are displayed. The layout and use of this Selection Tool window is the same as that already described for the Functional Data Flows in section 6.5.

Once the Data Flows have been selected they are moved to the right-hand window using the "Add" button, as shown in Figure 16. The Data Flow selection can then be modified in the same way as already described in section 6.5.

When the selection of the Data Flows needed by the Data Stores is complete it has to be confirmed by clicking on the "OK" button. This will also move the Selection Tool on to the next step in the Functional Viewpoint creation process.

Figure 16 – Adding the Data Flows for the selected Data Stores into the Functional Viewpoint



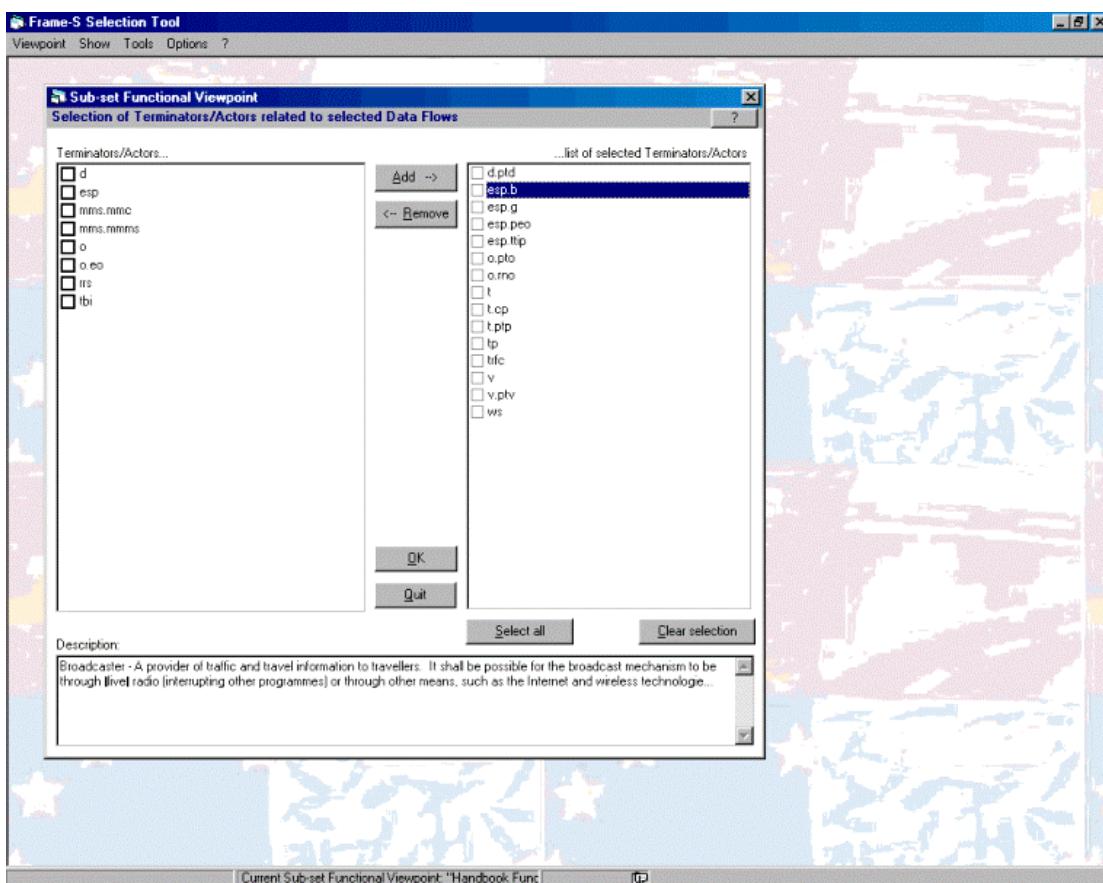
6.8 Step 7 – selecting the Terminators and Actors

Once the "OK" button is "clicked" in Figure 16 the screen shown in Figure 17 will appear automatically. This is the screen through which the Terminators and Actors that are to be used by the Functions are selected.

The left-hand part of the "Selection of Terminators/Actors..." window shows all of the Terminators and Actors that are at the sources or destinations of the previously selected Data Flows. They are listed in alphabetical order, with Actors being identified by their different style of mnemonics. The description of a Terminator or Actor is automatically displayed in the "Description" window part of the "Selection of Terminators/Actors..." window by clicking on its name, as shown in Figure 17 for the "Broadcaster" Actor within the "External Service Provider" Terminator.

Selecting particular Terminators and Actors for inclusion in the Viewpoint is achieved by clicking on the "square" boxes adjacent to their names so that a "√" symbol appears in each box. Note that selecting a Terminator will automatically select all of its Actors, so there may be some instances when some of these Actors will have to be de-selected because they are not needed. When all of the required Terminators and Actors have been selected they must be included in the Viewpoint by clicking on the "Add" button. This will cause them to move to the right-hand part of the "Selection of Data Stores..." window as shown in Figure 17. At the same time the selected Terminators and Actors will disappear from the left-hand window, leaving it to display only those that have not been selected.

Figure 17 – Selecting Terminators and Actors



If at this stage the user decides that any Terminators and/or Actors should not have been selected, then they can be de-selected by clicking the "square" boxes adjacent to their names in the right-hand part of the "Selection of Terminators/Actors..." window. All of the selected Terminators and/or Actors can be identified for de-selection by clicking on the "Select all" button. If a Terminator/Actor is incorrectly identified for de-selection, this can be cleared by clicking on its "square" box again. All of the identified Terminators/Actors can be removed from the de-selection list by clicking on the "Clear selection" button. Terminators and/or Actors are then actually de-selected by clicking on the "Remove" button.

When the selection of Terminators and Actors is complete it has to be confirmed by clicking on the "OK" button. This will also move the Selection Tool on to the next step in the Functional Viewpoint creation process.

6.9 Step 8 – correcting the errors and warnings resulting from the selections

6.9.1 Introduction

This section describes the methods that should be used to fix the errors and warnings that are usually found when selections are made as described in the previous sections. Errors are covered first as they have to be fixed before the Tool will allow the Viewpoint to be used as the starting point for the creation of a Physical Viewpoint. Warnings appear below the Errors as although strictly speaking they need not be fixed for the Functional Viewpoint to be used, they should be.

Users of the Selection Tool should not be alarmed that error and/or warning messages appear at the end of the selection process. As noted in Chapter 3, the FRAME Architecture is large and complex and it would perhaps be surprising if a complete set of User Needs, Functions, Data Flows, Data Stores and Terminators can be selected from it, without there being at least one error, or a warning of something that is wrong.

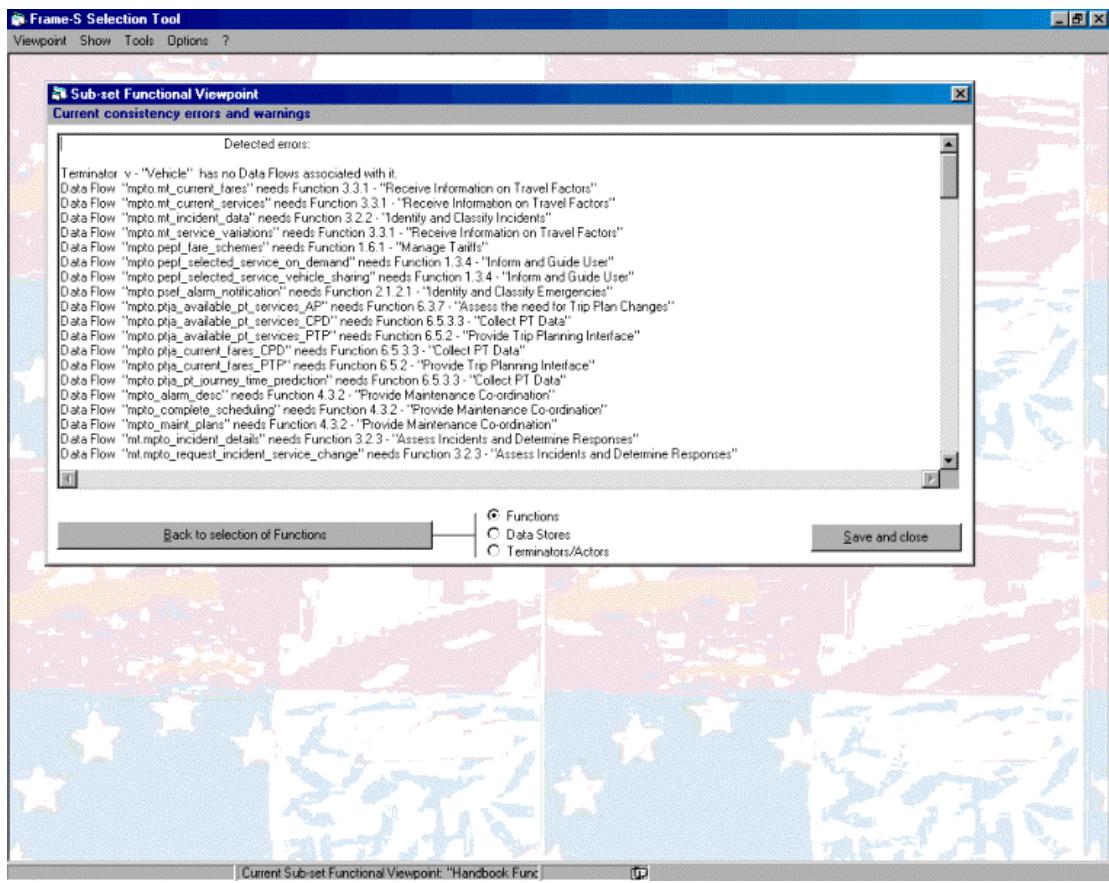
6.9.2 Dealing with errors from the selection process

Once the "OK" button is "clicked" in Figure 17 the selection of the contents of the Functional Viewpoint has been completed on this first pass through the selection process. There will almost certainly be some errors resulting from the selections made in this first pass. They will most likely have occurred for one of the following reasons:

- (1) No Data Flows have been associated with a Terminator, Function or Data Store;
- (2) A Data Flow does not have a source or destination Terminator, Function or Data Store.

An example of a typical set of error messages is shown by the screen shown in Figure 18. The scroll buttons must be used to enable all of the error messages to be seen plus any "warning" messages that come after the error messages.

Figure 18 – Example of the error messages produced from the initial creation of a Functional Viewpoint



6.9.3 Fixing errors in the selection of Functions, Data Stores and Terminators

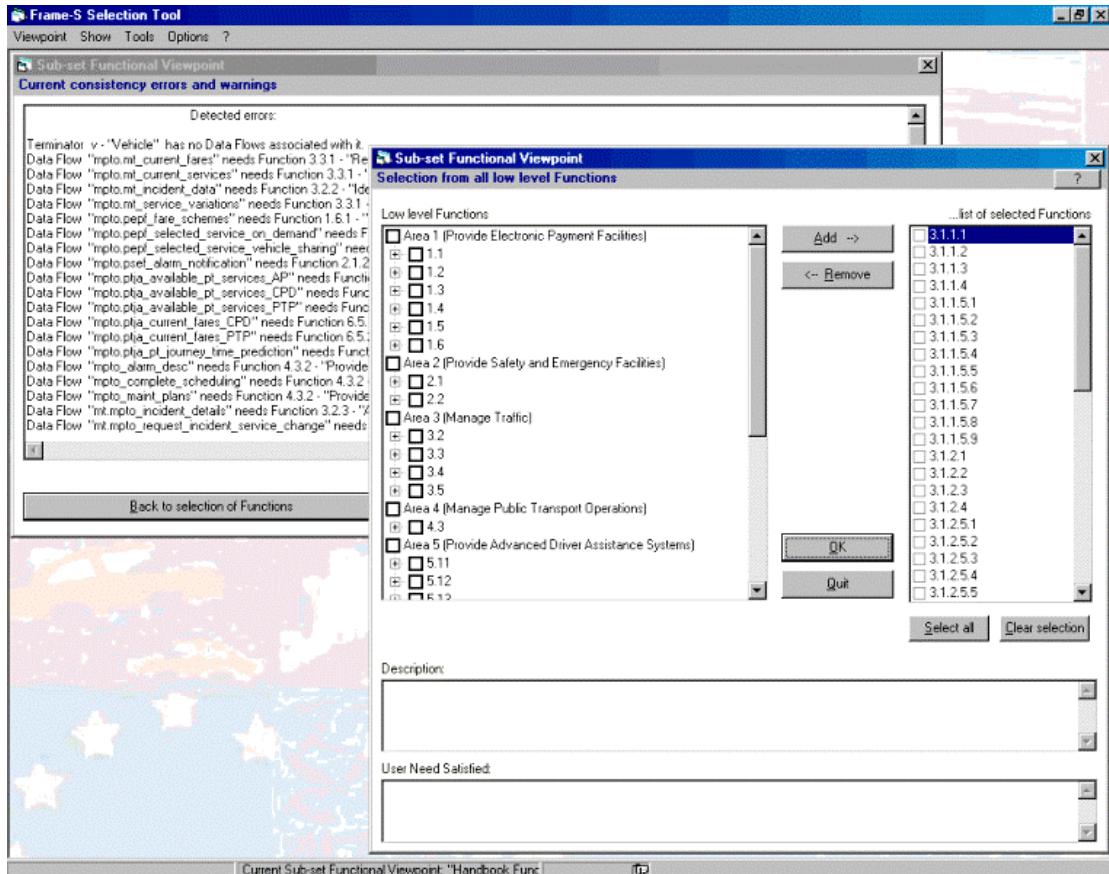
The causes of the error messages must now be removed by amending the selections of Functions, Data Flows, Data Stores and Terminators that have been made in Steps 3 to 7. The choice of Data Flows is dependent on the Functions, Data Stores and Terminators that have been selected, so the user is given the option to return one of these, as can be seen at the bottom of the window in Figure 18. Selecting either Functions, or Data Stores or Terminators changes the contents of the left hand button which is "clicked" to make the actual return.

When the return is made the window for the selection of Functions, Data Stores and Terminators is displayed. These have already been shown in Figure 11, Figure 15 and Figure 17 respectively. In each case the Functions, Data Stores and Terminators that have already been selected will appear in the right-hand part of the window and **all** of those that were not selected will appear in the left-hand part of the window. Thus for example, all of the Functions will appear, not just those that are connected with the selected User Needs.

As an aid to the correction of the errors, each of these windows will appear on top of the window with the list of errors shown in Figure 18. This is illustrated by Figure 19, which shows the window for selecting Functions has been re-displayed. Note that the error message and select Function (Data Store, Terminator or Data Flow) window can be moved to improve their visibility and that each window can be brought to the

front by selecting it. The window with the list of errors can be closed by the user, or left to be updated with the errors resulting from the next pass through steps 3 to 7.

Figure 19 – Example of the select Functions and error messages windows



The user must now decide how to amend the selection of Functions, Data Stores and/or Terminators/Actors in order to "fix" the errors. The processes for selection and de-selection have already been described in steps 3 to 7 (see sections 6.4 to 6.8) and should be followed again.

It is possible to follow an almost "mechanical" process and do whatever is necessary to "fix" the errors. However it is recommended that a more logical approach is followed in which the actual cause of the error is investigated using the FRAME Browsing Tool. This involves using the Tool to examine the cause of each error so that the appropriate solution can be found. Users should not be worried if as a result of an investigation it is found that "fixing" some errors requires the selection of Functions, Data Stores or Terminators/Actors not included in the first pass. Including them in the Functional Viewpoint must enable previously unavailable data to be provided, either directly or through some form of processing. In some cases, extra services may become available as a result, which again should not be a cause for concern, although Stakeholders should be notified in case they want the availability of these services to remain hidden.

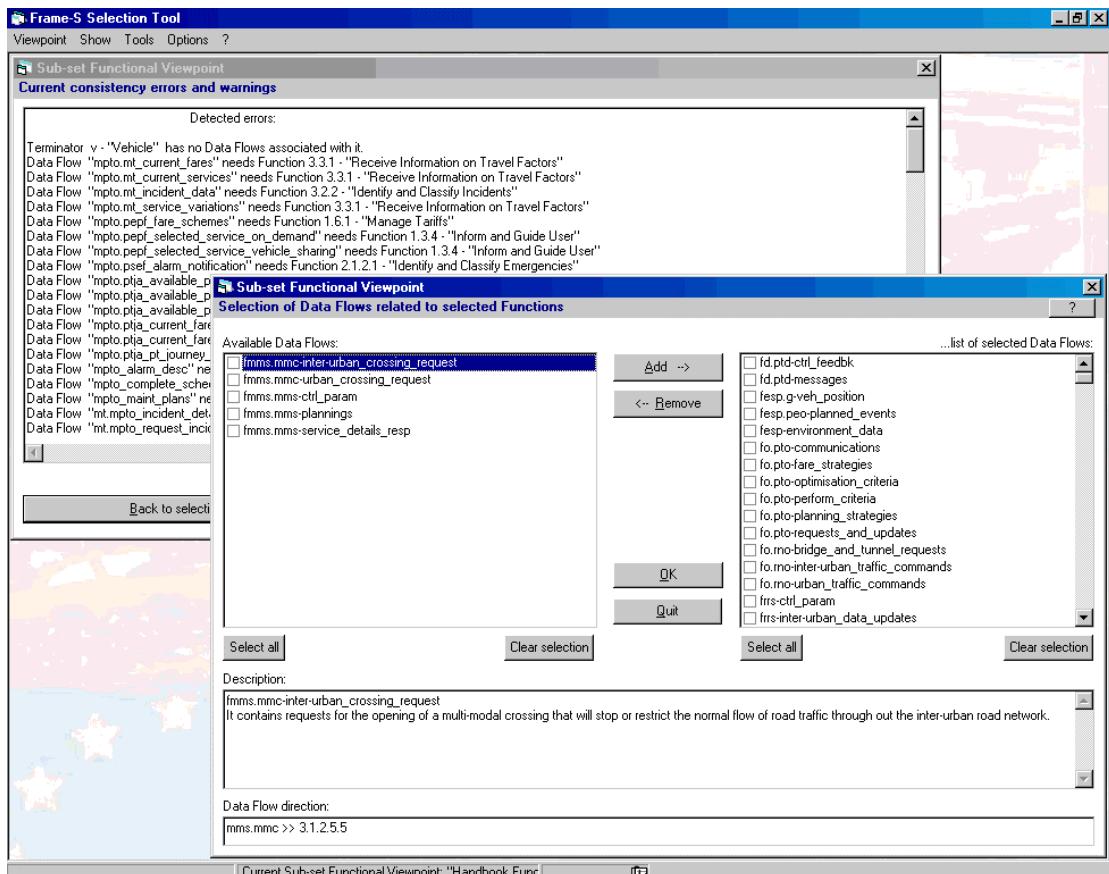
6.9.4 Fixing errors in the selection of Data Flows

If the errors concern missing Data Flows then they can only be "fixed" by returning to the selection of Functions or Data Stores screens shown in Figure 11 and Figure 15 respectively. As described in the previous section, this is achieved by selecting the

"Functions" or "Data Stores" options in Figure 18 and then clicking on the "Back to selection of ..." option.

The process for selecting and de-selecting Data Flows has already been described in steps 3 to 7 (see sections 6.4 to 6.8) and should be followed once more. Again it is recommended that the FRAME Browsing Tool is used as an aid to the investigation of which Data Flows need to be selected or de-selected. An example of the window through which Data Flows for Functions can be selected or de-selected is shown in Figure 20.

Figure 20 – Selection and de-selection of Data Flows



6.9.5 Completing the error fixing process

Once the "Back to selection of ..." option has been selected the Tool will pick up the process at the appropriate point in steps 3 to 7 (see sections 6.4 to 6.8). None of the following steps will be skipped, so for example returning to step 3 will mean that all of steps 4 to 7 will have to be repeated. For those steps where no changes are to be made then clicking on the "OK" button will retain the current selection.

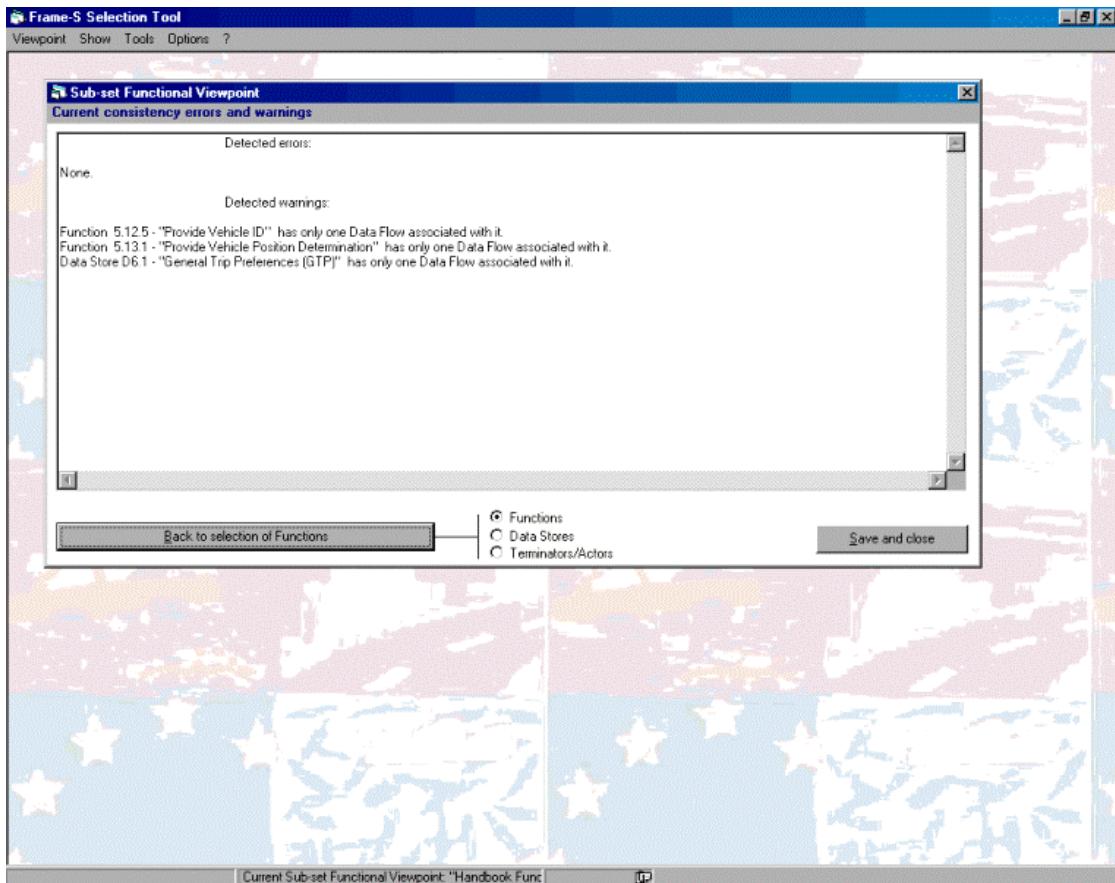
The process for "fixing" errors described in the previous sections can be repeated as many times as necessary until all the errors are "fixed". If it is decided that all the errors cannot be "fixed" without changing the selection of User Needs, then the "Save and close" button at the bottom right of the window in Figure 18 must be "clicked". The Functional Viewpoint should then be re-opened through the "Edit previous work" option, as described in section 5.4. This will cause the screen in Figure 9 to be displayed enabling the selection of User Needs to be changed. As a

consequence steps 2 to 7 (see sections 6.3 to 6.8) will have to be repeated in order to fully re-create the Functional Viewpoint.

6.9.6 Fixing warnings in the selection of Data Flows for Data Stores and/or Terminators/Actors

Once all of the errors have been "fixed" the screen shown in Figure 21 will be displayed. Now the warnings shown at the bottom of the window must also be fixed.

Figure 21 – Screen shot showing the display of warning messages



Sometimes the action taken to fix a warning message will result in the generation of further errors and warnings. For example, adding an extra Data Flow to remove a warning that a particular Function has only one Data Flow may result in the need for an additional Function or Terminator in order to provide the extra Data Flow with a source. It is not possible to move backwards through the screens for steps 3 to 7 – see sections 6.4 to 6.8. So for example, the only way to add a Function to support the addition of an extra Data Flow is to click "OK" for each of the screens in steps 4 to 7 – see sections 6.5 to 6.8 and then to click on "Back to selection of Functions" in Figure 21.

Any error messages produced by the additional selection of extra Functions, Data Stores, Data Flows and Terminators/Actors to fix the warning messages will have to be resolved in a similar way that described in the previous parts of this section.

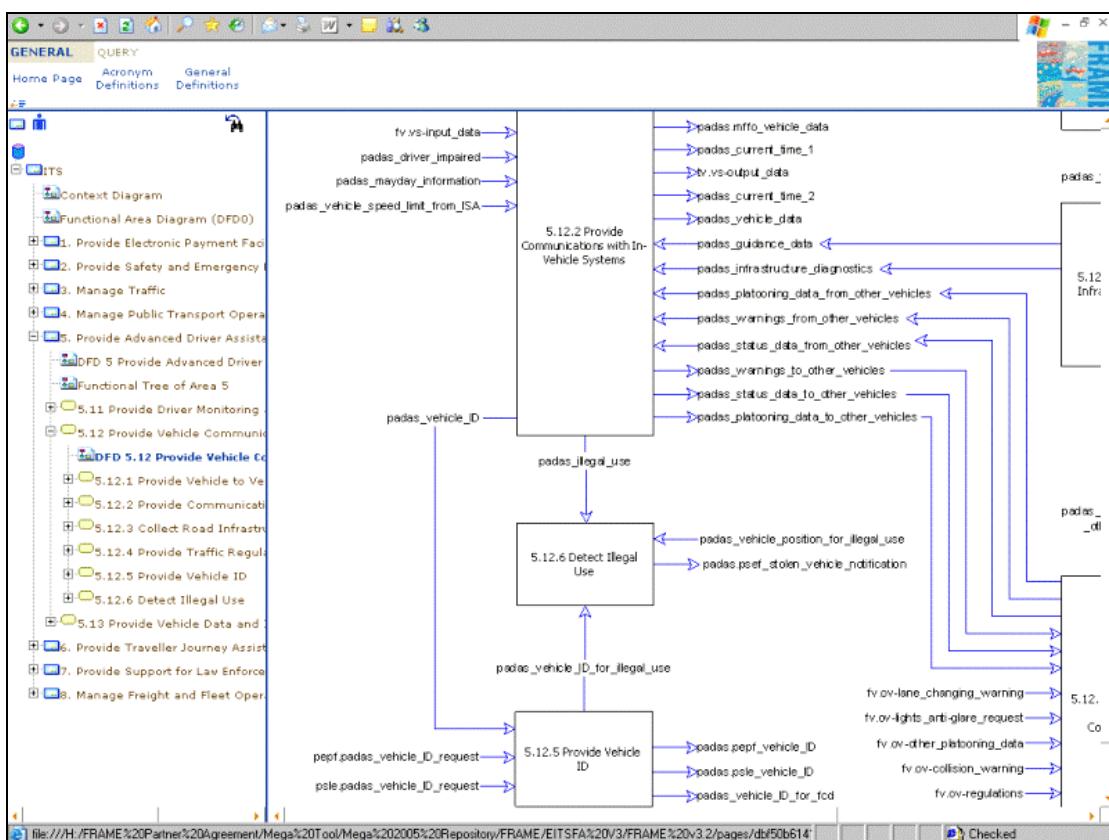
6.9.7 Using the FRAME Browsing Tool to fix faults

Fixing the warnings shown in Figure 21 provides an example of how the FRAME Browsing Tool can be used to support the use of the FRAME Selection Tool. Each of the warnings shown in Figure 21 needs knowledge of exactly what is in the FRAME Architecture for it to be fixed. For example it will be very difficult to work out if the fact that Function 5.12.5 has only one Data Flow will be a problem, and if it is, how this can be best resolved.

Figure 22 illustrates a page from the FRAME Browsing Tool showing the part of the FRAME Architecture containing the Function (5.12.5) mentioned in the first warning message in Figure 21. (Note that the "full screen mode" option of the Internet browser has been used to display the maximum amount of information.) The page has been found and displayed by expanding the structure for Functional Area 5 in the left-hand side of the browsing window.

As can be seen, Function 5.12.5 has a number of Data Flows that are its inputs and outputs, of which only one (padas_vehicle_ID_for_fcd) has currently been included in the Functional Viewpoint. From Figure 22 it can be seen that the Data Flow "padas_vehicle_ID" is needed to provide this data to Function 5.12.5. As a result of this Function 5.12.2 will be needed and the Terminator Data Flow fv.vs-input_data will also be needed to provide the source of the vehicle identity.

Figure 22 – Illustration of the use of the FRAME Browsing Tool



The FRAME Browsing Tool can also be used to display any part of the FRAME Architecture, as well as the descriptions of individual parts such as Functions, Data Stores, Data Flows, Terminators, Actors and User Needs. Thus the FRAME

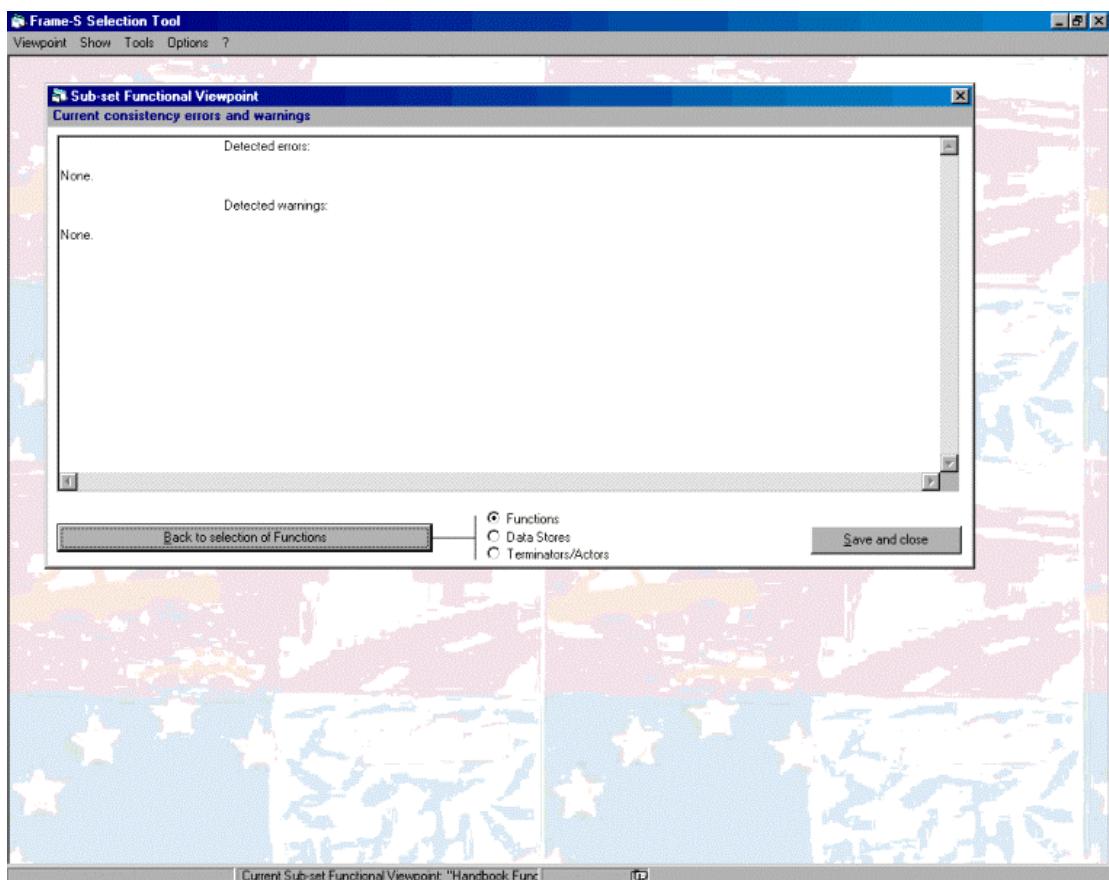
Browsing Tool can be used as an aid to the fixing of the other two warnings that are shown in Figure 21.

Note that it is possible to obtain a printout of the entire web-page that is displayed by the FRAME Browsing Tool. For displays of DFD's this can be done by right clicking on the diagram and selecting the "Print Picture...." option. Displays of the text from pages can only be printed by copying them into a WORD document, which is then printed in the usual way.

6.10 *The completed Functional Viewpoint*

The end result of all the work described in the previous sections of this Chapter is a Functional Viewpoint for which no error or warning messages are generated by the FRAME Selection Tool. The window produced by the Tool when this happens is shown in Figure 23.

Figure 23 – Completed Functional Viewpoint screen



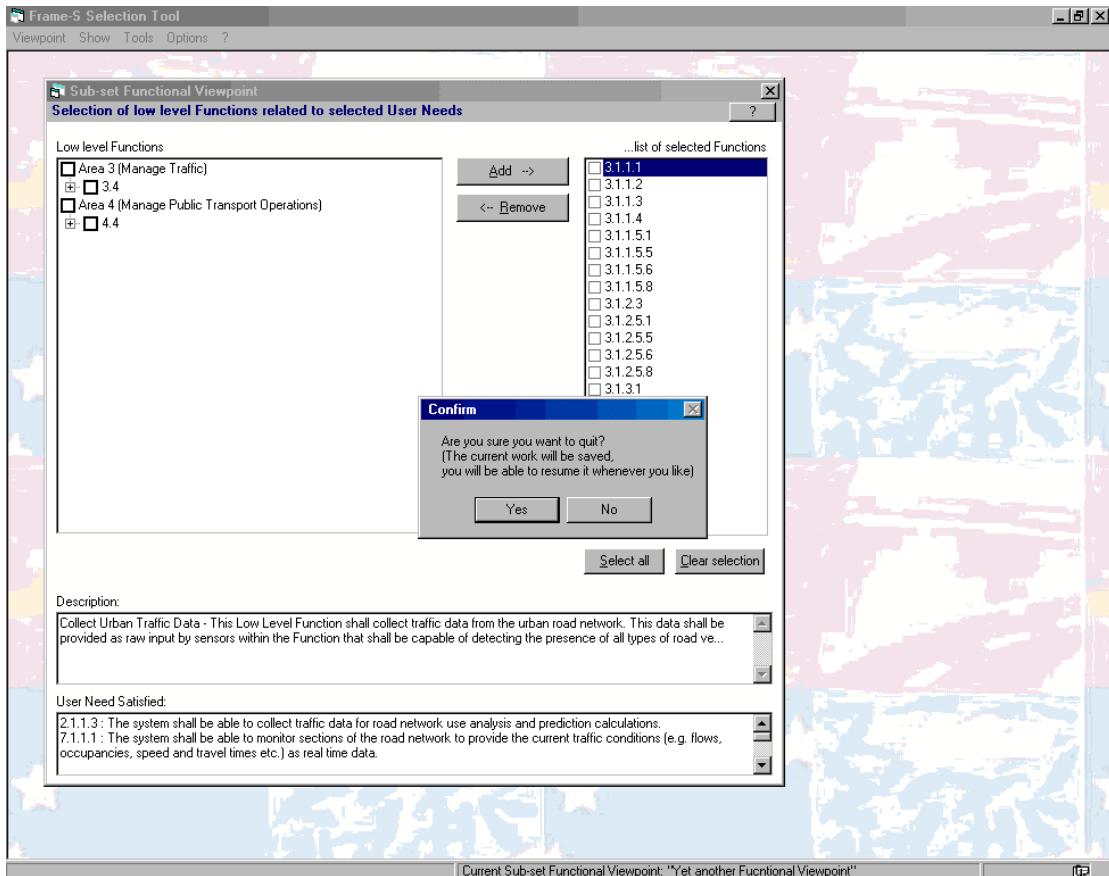
In order to complete the Functional Viewpoint creation process, it is necessary to click on the "Save and Close" button at the bottom of the window on the right-hand side. When this is done, the "Sub-set Functional Viewpoint" window will disappear and the Tool is now ready to start again with the selection of a Viewpoint to create – see Figure 7.

Note that the name of the Functional Viewpoint that has just been created will no longer appear in the "Continue with previous work" option. This is because the Selection Tool regards it as completed. It will however appear in the "Edit previous work" option enabling it to be changed.

6.11 Using the "Quit" button when creating the Functional Viewpoint

Most of the user interface screens provided by the Selection Tool include a button called "Quit" – see previous Figures. An example of the display that appears when the "Quit" button is "clicked" is shown in Figure 24. As can be seen, the user needs to confirm that they want to "Quit" the Functional Viewpoint creation process at this point. Clicking on the "No" option will return the user to the current screen and enable the continuation of the selection process.

Figure 24 – The "Quit" option in the Functional Viewpoint



Clicking on the "Quit" button enables the Selection Tool user to stop the selection process at the previous step. So, for example, if the "Quit" button is "clicked" (and the subsequent "OK" is also "clicked" – see Figure 24 above) when the screen shown in Figure 11 is being used, the selection of User Needs made on the previous screen will be retained, but any selection of Functions that has been made on this screen will be lost.

7. Using the FRAME Selection Tool – the Creation of a Physical Viewpoint

7.1 *Introduction*

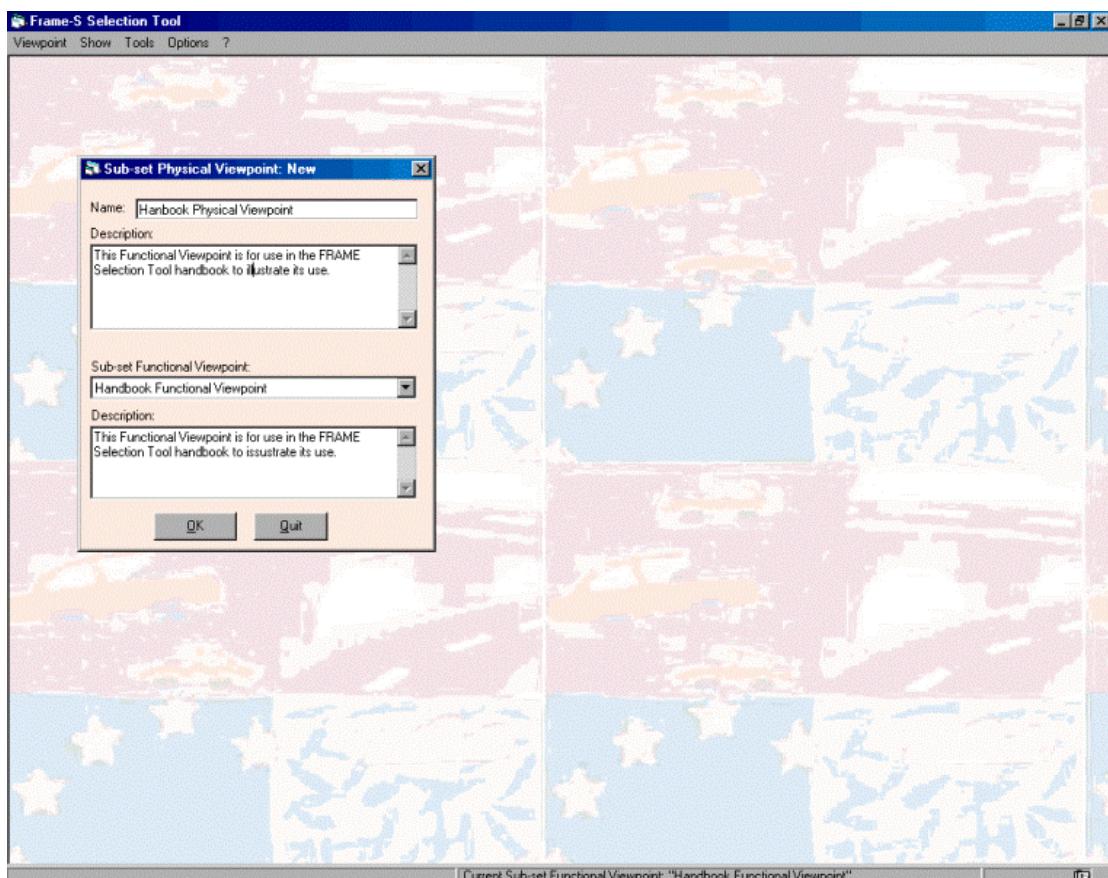
This section describes the way that the FRAME Selection Tool is used to create a new Physical Viewpoint. It follows the process described in section 3.2.3.

7.2 *Step 1 – naming the Physical Viewpoint and associating it with a Functional Viewpoint*

The first step in the process of creating a new Physical Viewpoint is to give it a name and to associate it with a previously created Functional Viewpoint to act as the starting point for its creation. This is done using the screen shown in Figure 25 which appears automatically when the "New" option is selected for a Physical Viewpoint using the menu shown in Figure 7. A name has to be created, but its actual content and the addition of a description is entirely up to the user.

The Functional Viewpoint that will act as the starting point for the creation of the new Physical Viewpoint is selected by clicking on the drop down arrow at the end of the "box" labelled "Sub-set Functional Viewpoint". This will produce a list of available Functional Viewpoints from which one can be selected. When this is done, its description will then be displayed for reference purposes only – it cannot be changed.

Figure 25 – Naming the Physical Viewpoint and associating it with a Functional Viewpoint

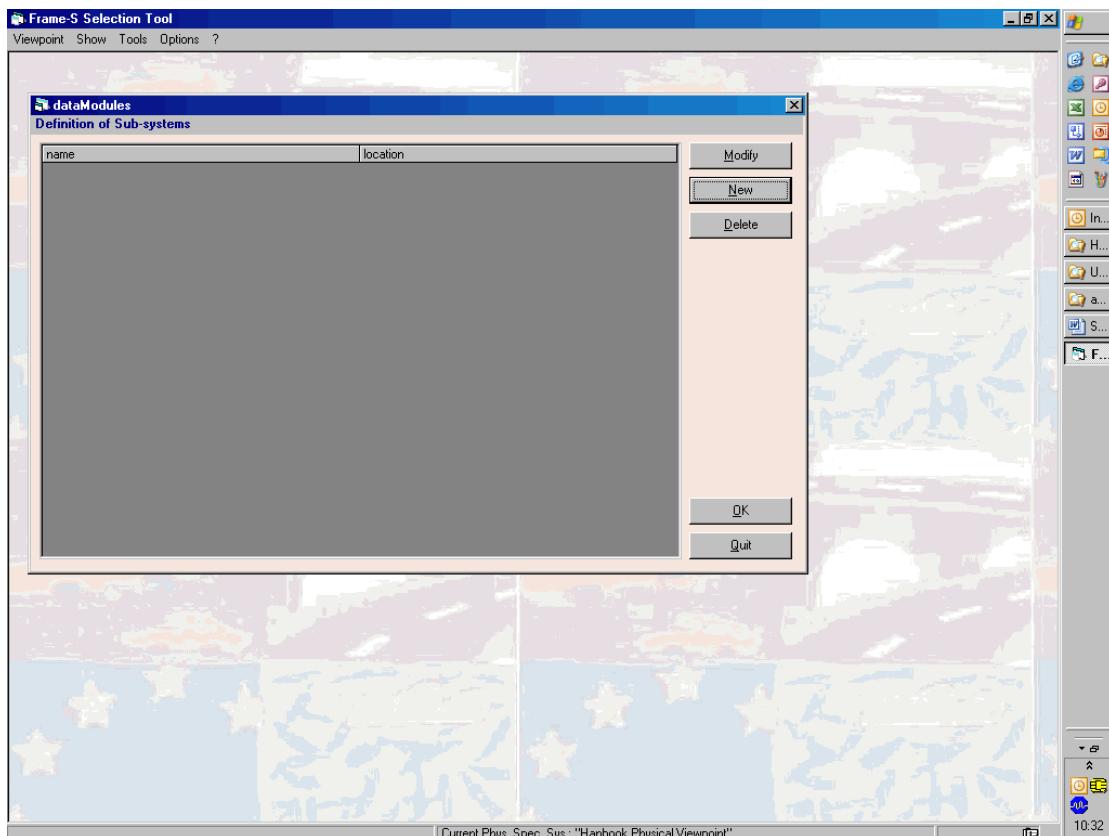


Once the name (and optionally the description) of the new Physical Viewpoint is complete and the Functional Viewpoint that is to be the starting point for its creation has been selected, the "OK" button should be pressed. A confirmation screen similar to that shown in Figure 8 will then be produced, unless the chosen Physical Viewpoint name has already been used. In this situation, the user will be asked to choose another name. Clicking on the "OK" button in the confirmation screen will cause the Selection Tool to move onto the next step.

7.3 Step 2 – defining the Sub-systems

The next step in its creation is to define the Sub-systems that it will include. This is achieved clicking on the "New" option in the screen shown in Figure 26.

Figure 26 – Screen from which Sub-systems are defined

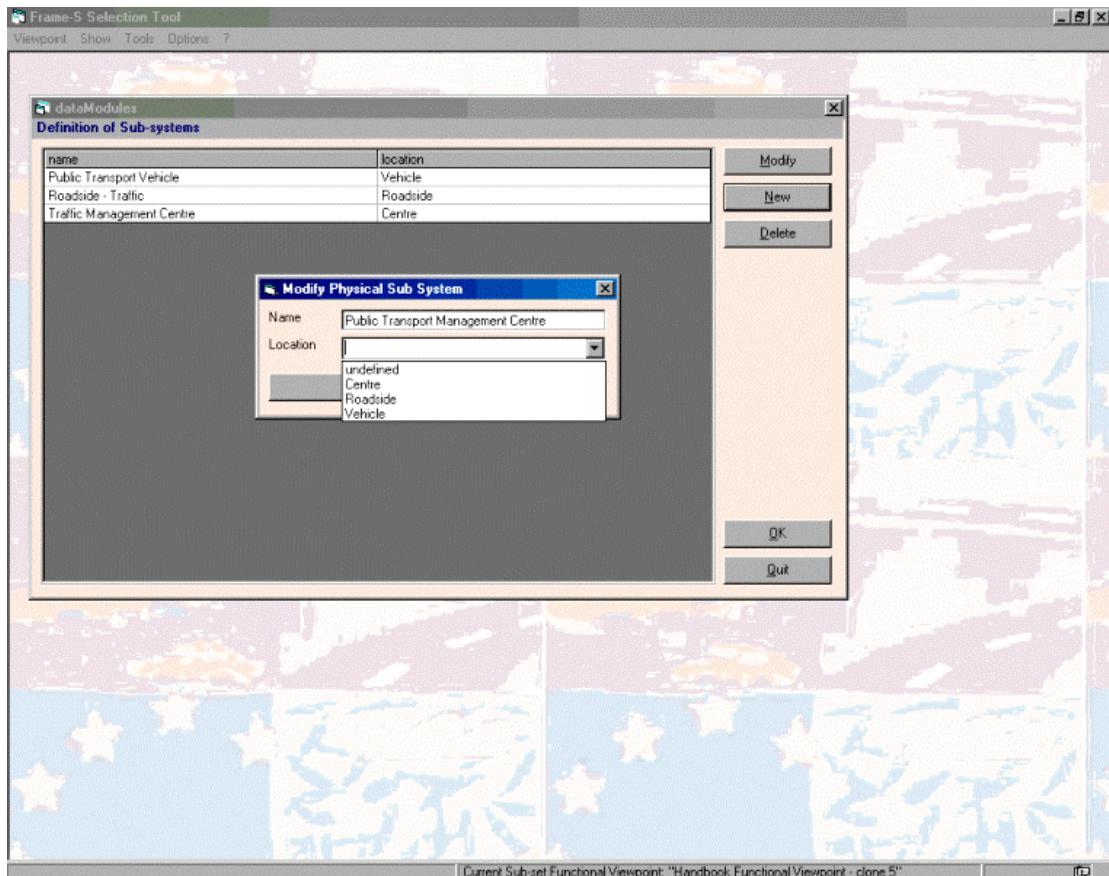


The result of clicking on the "New" button in Figure 26 is the appearance of an additional window (Modify Physical Sub System) into which the name and location of each Sub-system are entered, as shown in Figure 27. When the first Sub-system name and location have been entered, the "OK" button in the additional window (hidden by the drop down list) is "clicked" and it will close. The Sub-system name and location will then appear at the top of the "Definition of Sub-systems" window. The "New" button should then be "clicked" again so that the name and location of the second Sub-system can be entered. This process is repeated until information about all of the Sub-systems has been entered. Note that use of <arrow>, <tab> and <return> keys will achieve the same effect.

Figure 27 actually illustrates the situation after three Sub-systems have been created and with the fourth in the process of being created. The user is given the option of allocating the new Sub-system to an existing location (see the drop down list) or

specifying a new location. In this particular case, the "Public Transport Management Centre" Sub-system will be allocated to the "Centre" location, illustrating that it is possible to have more than one Sub-system in the same physical location.

Figure 27 – Setting locations for Sub-systems



If at any point it is decided that the name or location of one of the Sub-systems needs to be changed, then it is selected from the list on the "Definition of Sub-systems" window and the "Modify" button clicked. This will make the "Modify Physical Sub-System" window re-appear so that the name and/or location can be changed.

When all the Sub-systems have been defined, the Selection Tool can be moved on to the next step by clicking on the "OK" button.

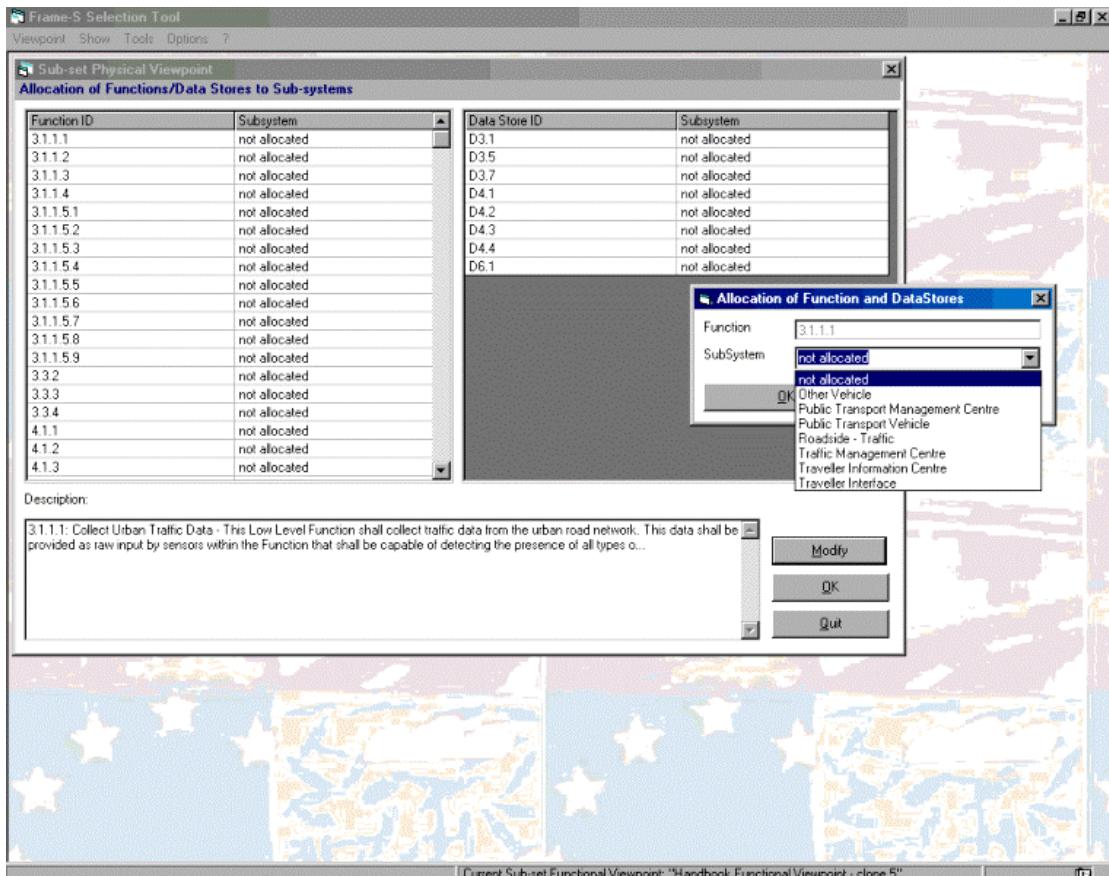
7.4 Step 2 – allocating Functions to Sub-systems

Once the definition of the Sub-systems has been completed, clicking on the "OK" button in Figure 27 will cause the screen shown in Figure 28 to appear. It enables each of the Functions and Data Stores in the Functional Viewpoint to be allocated to one of the Sub-systems.

To make this allocation, first click a Function or Data Store number. This will cause the number and first part of the description to appear in the "Description" window at the bottom of the screen. Then click on the "Modify" button, which will cause the "Allocation of Function and Data Stores" window to appear with the Function or Data Store number displayed next to "Function". Clicking on the drop down arrow next to "SubSystem" will cause the list of Sub-systems that have just been created to appear, as shown in Figure 28. The Sub-system to which the Function or Data Store

is to be allocated is then selected and the "OK" button (hidden by the drop down list) is clicked. This will close the "Allocation of Function and Data Stores" window and cause the selected Sub-system to appear in the "Subsystem" column next to the "Function ID" or "Data Store ID" for which the allocation has just been made.

Figure 28 – Allocation of Functions to Sub-systems



This process is repeated until all of the Functions and Data Stores have each been allocated to one of the Sub-systems. Note that use of <arrow>, <tab> and <return> keys will achieve the same effect. The "OK" button is then "clicked" which will cause the Selection Tool to move on to the next step.

7.5 Step 3 – creation of Modules

Once the allocation of Functions and Data Stores to Sub-systems has been completed, clicking on the "OK" button in Figure 28 will cause the screen shown in Figure 29 to appear. It enables any Modules that are to be part of one or more of the Sub-systems to be defined.

Initially the panel in the centre of the window is blank (all grey) with none of the small windows shown in Figure 29 being shown. The panel on the left of the screen provides a list (in alphabetical order) of all the Sub-systems with the Functions and Data Stores (in numerical order) that have been allocated to them.

The creation of Modules begins by clicking on the "New" option shown on the right of the screen in Figure 29. This will cause an additional window to appear (it has the same title as the main window – "Definition of Modules") into which the Module name is entered. The identity of its parent Sub-system is selected by clicking on the drop

down arrow and selecting the appropriate one from the list that is displayed, as shown in Figure 29. When the first Module name and Sub-system identity have been entered, the "OK" button in the additional window (hidden by the drop down list) is "clicked" and it will close. The Sub-system name and location will then appear in the table that is part of the larger "Definition of Sub-systems" window. The "New" button should then be "clicked" again so that the name and identity of its parent Sub-system for the second Module can be entered. This process is repeated until information about all of the Modules has been entered. Note that use of <arrow>, <tab> and <return> keys will achieve the same effect.

The list of all the Sub-systems with the Functions and Data Stores that have been allocated to them in the panel on the left of the screen in Figure 29 is useful for checking that all of the Functions are in the expected Sub-systems. If any are Functions or Data Stores are found to be missing, or their allocation to a particular Sub-system needs to be changed,

Figure 29 – Defining Modules

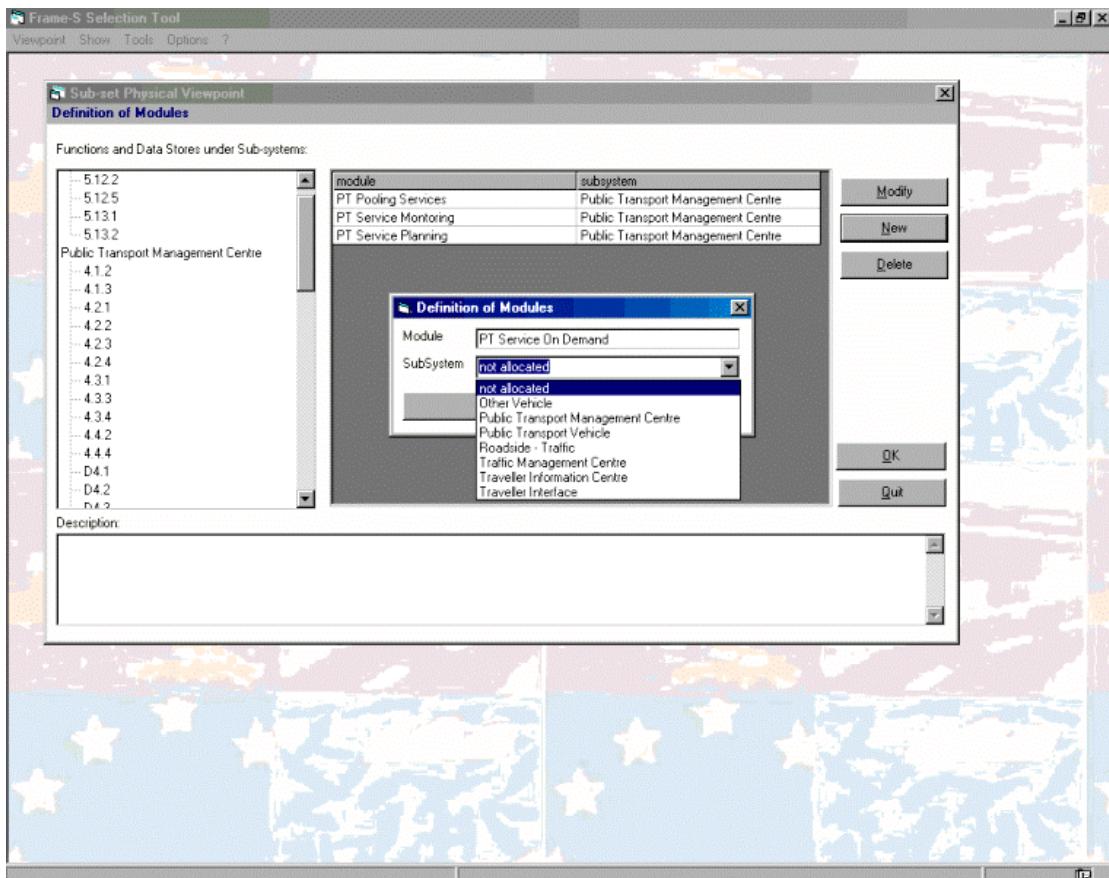


Figure 29 actually illustrates the situation after three Modules have been created and with the fourth in the process of being created. The user is given the option of allocating the new Module to any of the existing Sub-systems. In this particular case, the "PT Service On Demand" Module will be allocated to the "Public Transport Management Centre" Sub-system. Note that the Selection Tool does not allow two Modules to have the same name, even if they are in different Sub-systems.

If at any point it is decided that the name or identity of the parent Sub-system for one of the Modules needs to be changed, then it is selected from the list of those that have been created and the "Modify" button clicked. This will make the smaller

"Definition of Modules" window re-appear so that the name and/or parent Sub-system identity can be changed.

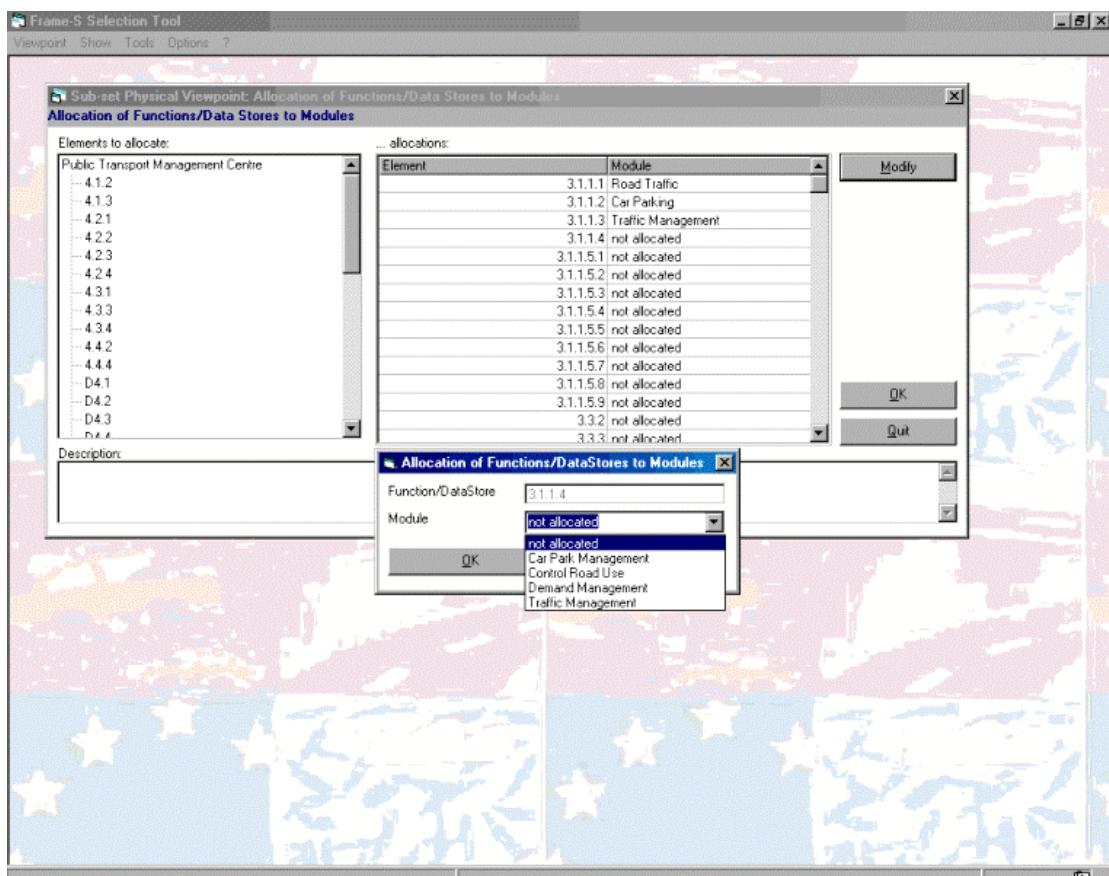
When all the Modules have been defined, the Selection Tool can be moved on to the next step by clicking on the "OK" button.

7.6 Step 4 – *allocating Functions and Data Stores to Modules*

Once the Modules have been defined, clicking on the "OK" button in Figure 29 will cause the screen shown in Figure 30 to appear. It enables each of the Functions and Data Stores in the Functional Viewpoint to be allocated to a Module within the Sub-system to which they were allocated in the previous step.

To make this allocation, first click a Function or Data Store number in the "Element" table of the "...allocation:" window, then click on the "Modify" button, which will cause the "Allocation of Function and Data Stores to Modules" window to appear with the Function number displayed next to "Function". Clicking on the drop down arrow next to "Module" will cause the list of Modules in the Sub-system to which the Function or Data Store has been allocated to appear, as shown in Figure 30. The Module to which the Function or Data Store is to be allocated is then selected and the "OK" button (hidden by the drop down list) is clicked. This will close the "Allocation of Function and Data Stores to Modules" window and cause the selected Module name to appear in the "Module" column next to the Function or Data Store number in the "Element" list for which the allocation has just been made. Note that use of <arrow>, <tab> and <return> keys will achieve the same effect.

Figure 30 – Allocation of Functions and Data Stores to Modules



Again the list on the left of the screen in Figure 29 provides a list of all the Sub-systems (in alphabetical order) with the Functions and Data Stores that have been allocated to them. This is useful for checking that all of the Functions are in the expected Sub-systems.

This process is repeated until all of the Functions and Data Stores have each been allocated to one of the Sub-systems. The "OK" button is then "clicked" which will cause the Selection Tool to move on to the next step.

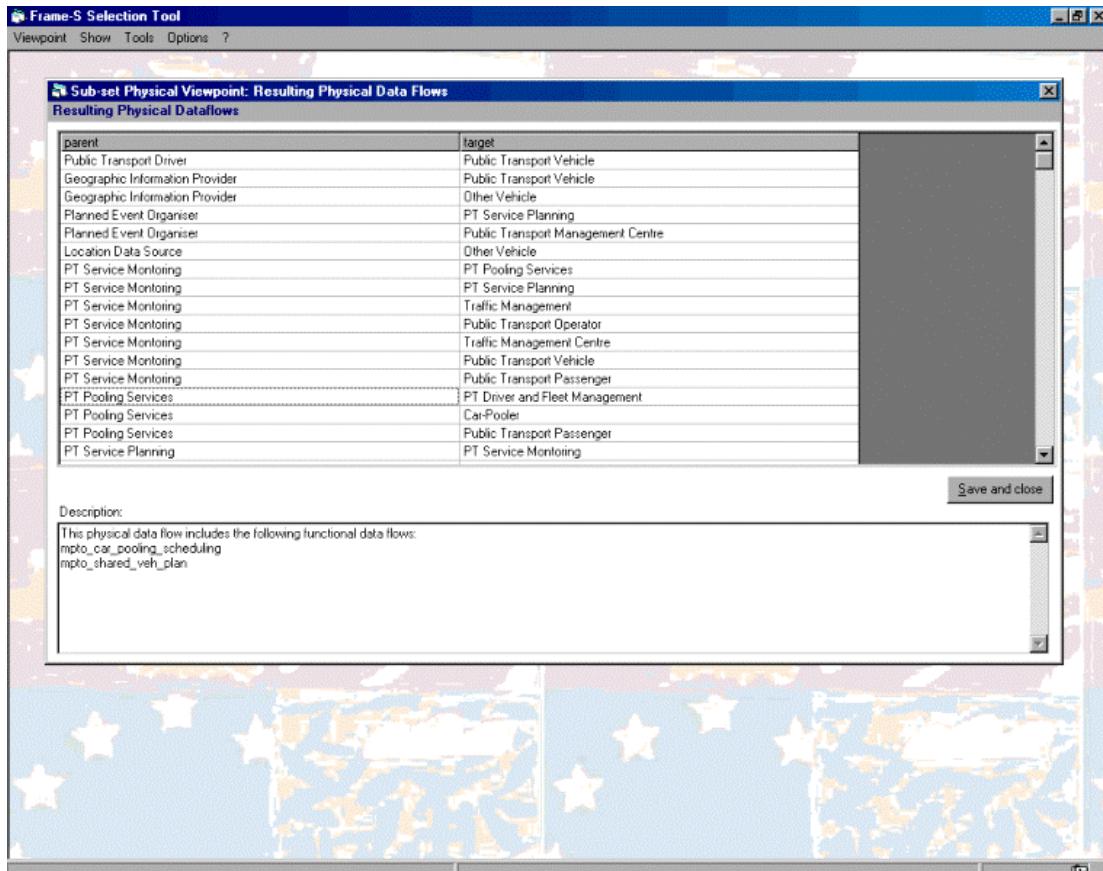
This process is repeated until all of the Functions and Data Stores have each been allocated to one of the Modules (if any) in the Sub-system to which they were previously allocated. Note that if there are no Modules in a Sub-system then the only option provided in the drop down list is "not allocated". The "OK" button is then "clicked" which will cause the Selection Tool to move on to the next step.

7.7 Step 5 – creation of Physical Data Flows

The next step in the process of producing the Physical Viewpoint is the creation of the Physical Data Flows. It is initiated by clicking the "OK" button in Figure 30, which will produce the screen shown in Figure 31.

The creation of the Physical Data Flows is automatic and is created from the Functional Data Flows between the Functions, Data Stores and Terminators in each of the Sub-systems and Modules. Clicking on one of the items in either the "Parent" or "Target" columns will show the identities of the constituent Functional Data Flows in the "Description" window, as illustrated by the example in Figure 31.

Figure 31 – List of Physical Data Flows



The only way to see the Physical Data Flows properly is to create a report, save it and then browse through the information. It may also be helpful to create the other reports that are available for the Physical Viewpoint. The description of how to do this and to produce Microsoft® Excel® Workbooks containing information about the contents of the Physical Viewpoint is provided in Chapter 8.

7.8 Using the "Quit" button when creating the Physical Viewpoint

The appearance and operation of the "Quit" button during the creation of a Physical Viewpoint is much the same as in the creation of a Functional Viewpoint. This is described in section 6.11. Thus for example when the "Quit" button is "clicked" (and it is subsequently confirmed) in Figure 28, the created Sub-systems will be saved. However any allocations of Functions to Sub-systems will be lost.

8. Creation of Reports showing the results from the creation process

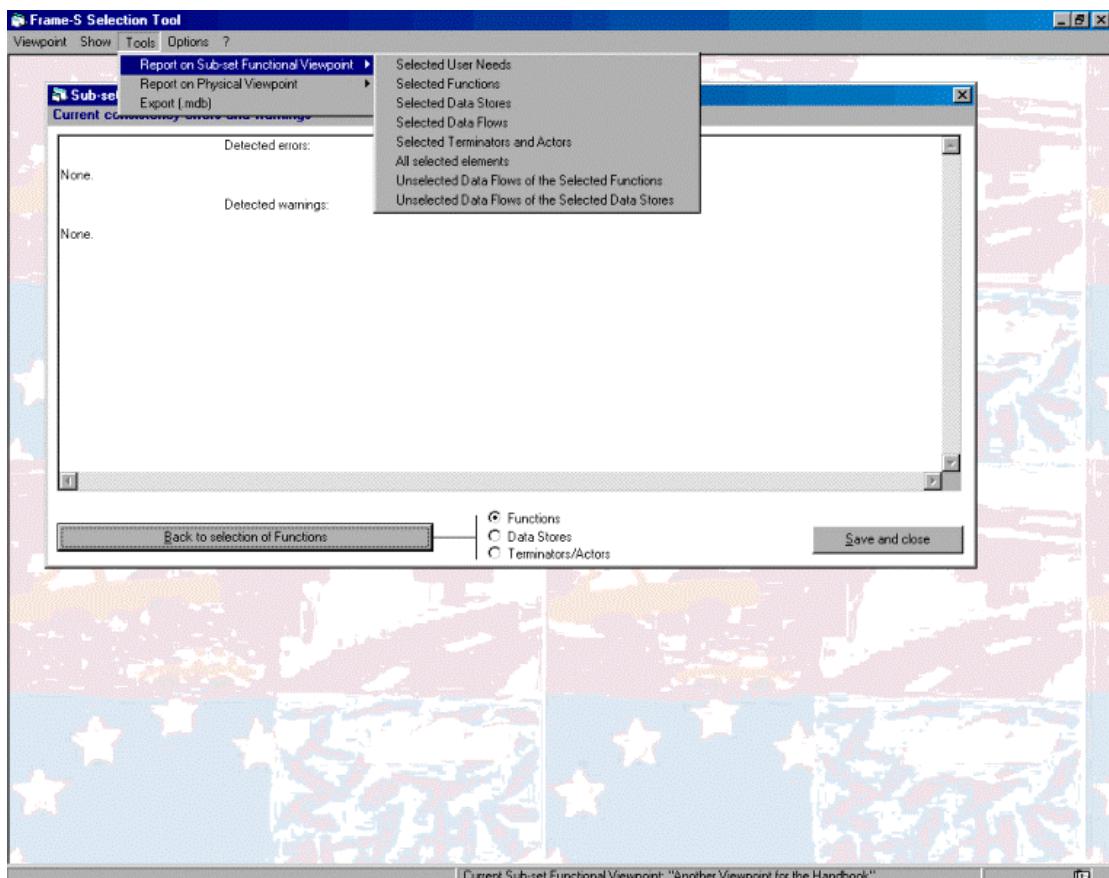
8.1 *Introduction*

The FRAME Selection Tool provides facilities that enable the production of several "reports" from the results of the selection process. The "reports" are available to show the results for both the Functional and Physical Viewpoints and are the main way to get information about their contents, once the selection process has been completed.

8.2 *Selecting the type of "report" for the Functional Viewpoint*

The type of "report" that is to be produced from the Functional Viewpoint is selected by clicking on "Tools" drop down menu at the top of the screen and then selecting "Report on Sub-set Functional Viewpoint". This will produce the display shown in Figure 32.

Figure 32 – Options for the creation of Functional Viewpoint "reports"

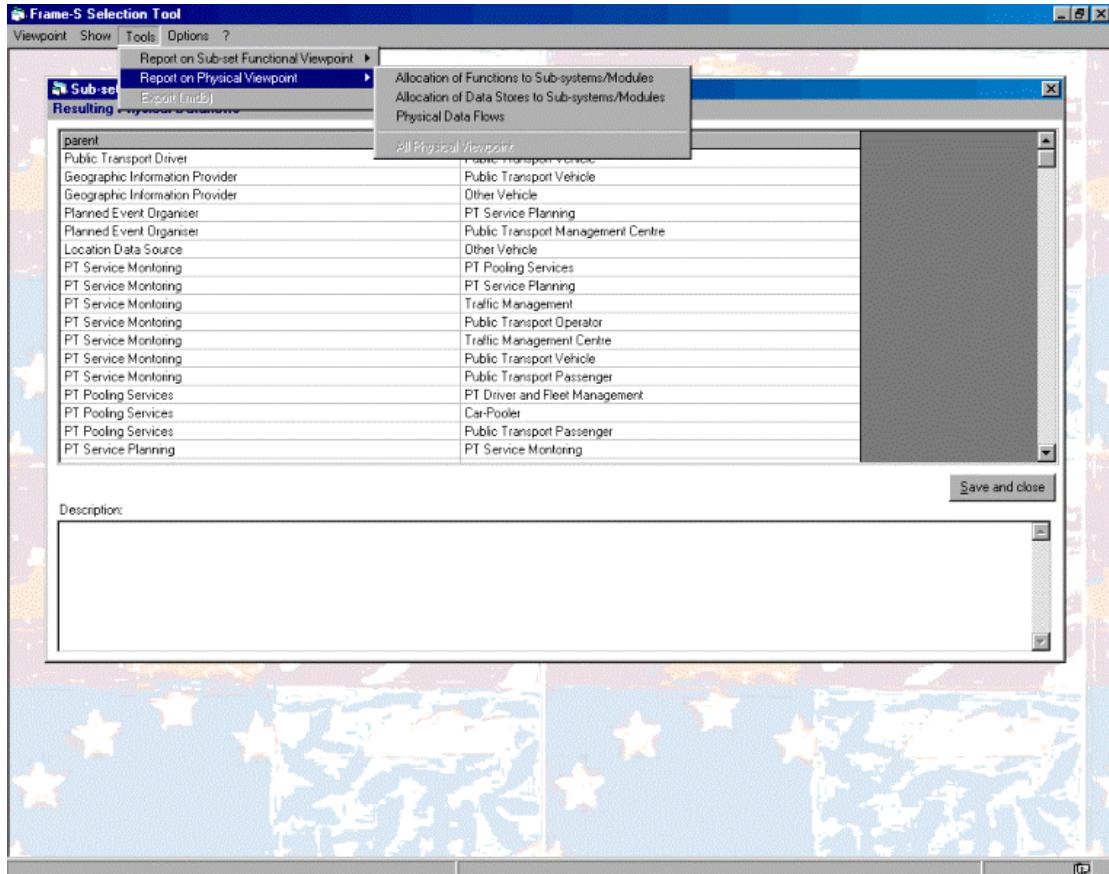


The information contained the "reports" produced by each of the options should be obvious from their names. So for example, the "report" produced by the "All selected elements" options contains all of the information provided by each of the "Selected..." options above it.

8.3 Selecting the type of "report" for the Physical Viewpoint

Again the type of "report" that is to be produced from the Physical Viewpoint is selected by clicking on "Tools" drop down menu and then selecting "Report on Physical Viewpoint". This will produce the display shown in Figure 33.

Figure 33 – Options for the creation of Physical Viewpoint "reports"



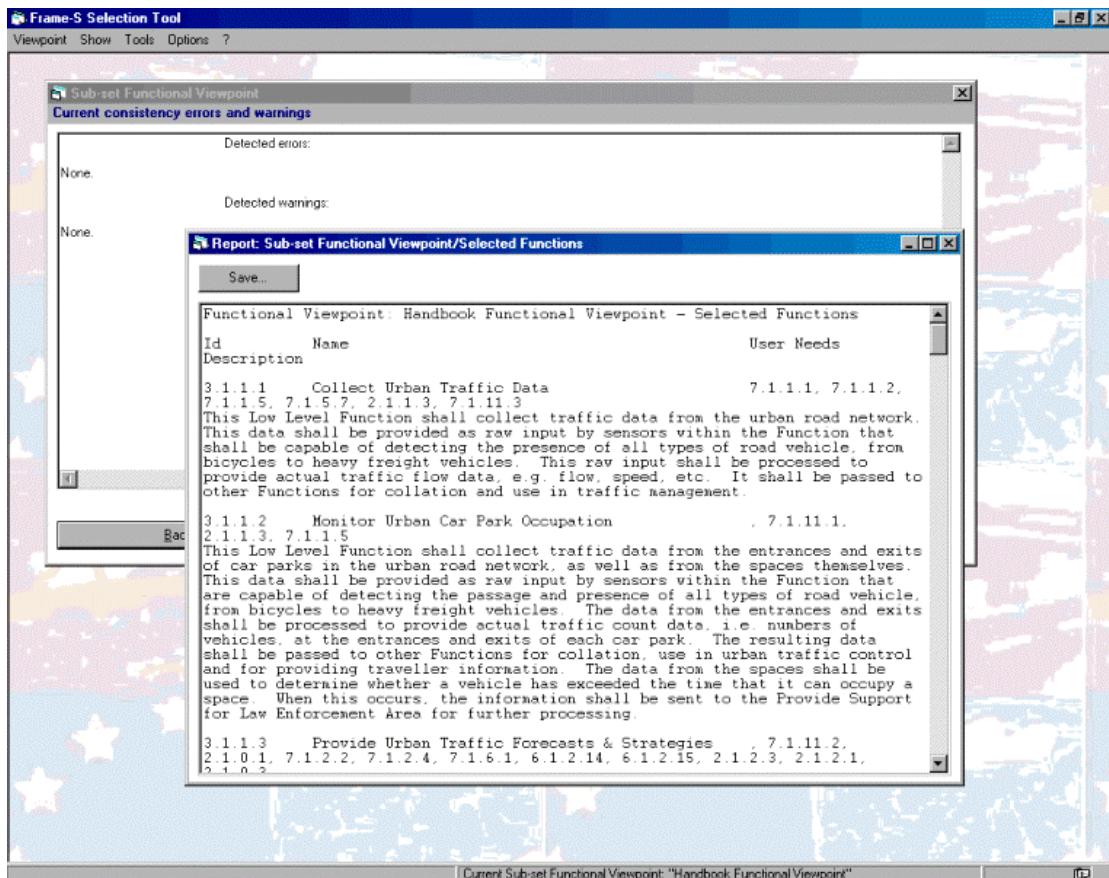
The information contained the "reports" produced by each of the options should be obvious from their names.

8.4 Previewing and saving reports

Once a "report" has been selected, a preview of it appears in a separate window, as shown by the example in Figure 34. This is for a list of the Functions in a Functional Viewpoint, for which the Function number, name, User Needs, and Overview description are provided.

In a file in "CSV format" the information is saved as a series of text strings, one set for each element in the "report". A separate line is used for all of the information about each of the elements, regardless of how much text is involved. So as can be seen from Figure 34, the appearance of the "report" in the preview window can sometimes be difficult to understand and should be used to confirm that the information required has been selected. It should then be "saved".

The solution is to re-format the CSV formatted file, which is easily done by importing it into a Microsoft® Excel® and re-formatting it in the process. This is described in section 8.5.

Figure 34 – An example of a preview of a Functional Viewpoint "report"

When the "Save" button is "clicked" the user must select the folder and the name of the file to be saved in CSV format. Confirmation that the save has taken place is provided by a separate window which itself requires a click on the "OK" button. The resulting CSV file is not easy to read and needs a little processing to make it so, as defined below.

8.5 Making reports easier to read

In order to make the "report" easier to read, it is necessary to convert it into a more acceptable format. This is done using Microsoft® Excel®, or an equivalent "office" type application.

The way to convert the CSV file using Microsoft® Excel® is by using the following steps (Note: other compatible applications may have different procedures):

1. Run Microsoft® Excel® and open a new Workbook.
2. On the "Data" menu for the first spreadsheet, point to "Import External Data", and then click "Import Data".
3. In the "Files of type" box, click "Text Files (*.txt; *.prn; *.csv; *.tab;...)".
4. In the "Look in" list, locate and double-click the CSV file with the "report" that needs to be display
5. The Text Import Wizard should then open. On the first page, select "Delimited" to define the type of data and click "Next". Then specify "Other" for the "Delimiter" and add the "|" (vertical bar – shift \ keys) as the actual delimiting character. The "Data Preview" window should now

show the data arranged in columns. Click on "Next" but make no changes before clicking on "Finish".

6. In the "Import Data" dialog box, do not click on anything except "OK".

The new Workbook should now contain the "report" data as the first spreadsheet with the different parts of the information about the elements arranged in different columns. Any comma "," characters will have been replaced by "^" (shift 6) characters. The actual format and layout of the columns and rows can be set up to suit individual user requirements using the standard facilities in Microsoft® Excel®. An example of a "processed report" for the Physical Data Flows as a Microsoft® Excel® spreadsheet is shown in Figure 35.

Figure 35 – An example of a Physical Data Flows "report"

A	B	C	D	E	F
1	Physical Data Flow Parent	Target	Functional Data Flow	Source	Destin
2	P00473	Public Transport Driver	fd.ptd-ctrl_feedbk	d.ptd	4.4.3
3	P00473	Public Transport Driver	fd.ptd-messages	d.ptd	4.1.4
4	P00474	Geographic Information Provider	fesp.g-veh_position	esp.g	4.1.1
5	P00475	Geographic Information Provider	fesp.g-data	esp.g	5.13.1
6	P00476	Planned Event Organiser	fesp.peo-planned_events	esp.peo	4.2.1
7	P00477	Planned Event Organiser	fesp.peo-planned_events	esp.peo	4.2.1
8	P00478	Location Data Source	flds-vehicle_location	lds	5.13.1
9	P00479	PT Service Monitoring	mpto_current_veh_status	D4.1	4.3.4
10	P00480	PT Service Monitoring	mpto_route_data_for_planning	D4.4	4.2.1
11	P00480	PT Service Monitoring	mpto_route_data_for_reports	D4.4	4.2.4
12	P00481	PT Service Monitoring	mpto_mt_vehicle_priority_request	4.4.2	3.1.1.!
13	P00482	PT Service Monitoring	to.pto-performance_fig	4.1.3	o.pto
14	P00482	PT Service Monitoring	to.pto-pred_pt_data	4.1.2	o.pto
15	P00483	PT Service Monitoring	mpto_mt_vehicle_priority_request	4.4.2	3.1.1.!
16	P00484	PT Service Monitoring	mpto_current_vehicle_data	D4.1	4.4.1
17	P00484	PT Service Monitoring	mpto_observed_performance_fig	4.1.3	4.4.1
18	P00485	PT Service Monitoring	tt.ptp-pred_pt_info	4.1.2	t.ptp
19	P00486	PT Pooling Services	mpto_car_pooling_scheduling	4.2.2	D4.3
20	P00486	PT Pooling Services	mpto_shared_veh_plan	4.3.4	4.3.3
21	P00487	PT Pooling Services	tt.cp-plan	4.2.2	t.cp
22	P00488	PT Pooling Services	tt.ptp-shared_veh_plan	4.3.4	t.ptp
23	P00489	PT Service Planning	mpto_historical_veh_data	D4.2	4.1.2
24	P00489	PT Service Planning	mpto_route_data_updates	4.2.4	D4.4
25	P00490	PT Service Planning	mpto_historical_data	D4.2	4.3.4
26	P00491	PT Service Planning	mpto_up_dated_scheduling	4.2.1	D4.3
27	P00492	PT Service Planning	to.pto-responses	4.2.4	o.pto
28	P00493	PT Service Planning	mpto_recorded_data	D4.2	4.4.1
29	P00494	PT Service On Demand	mpto_on_dmd_services	4.3.1	4.3.3

Only some of the columns are shown in this example. The partially visible column on the right hand side is the "Destination" of each Physical Data Flow. There is also another column to the right that contains the description of each Functional Data Flow that is used in the Physical Data Flows.

As can be seen, each Physical Data Flow has not been given a name "Pxxxxx". Whilst they can be changed to something more meaningful, it should be remembered that each time a Physical Viewpoint is re-created, by going from Step 4 to Step 5, a new set of Pxxxxx numbers will be allocated by the Selection Tool. However if the Data Flow names are to be changed, then the user is free to choose whatever names are most appropriate using the "cut and paste" facilities provided by Microsoft® Excel®.

8.6 Other "reports" for the Physical Viewpoint

Two other "report" for a Physical Viewpoint appear in the drop down list shown in Figure 33: "Allocation of Functions to Sub-systems/Modules" and "Allocation of Data Stores to Sub-systems/Modules". Examples of these reports are shown in Figure 36

and Figure 37. They have been converted from their original CSV files into Microsoft® Excel® spreadsheets and re-formatted to make them easier to read.

Figure 36 – An example of a Function allocation "report"

A	B	C	D	E
1 Sub-system Name	Sub-system Location	Function Id	Function Name	Module
2 Roadside - Traffic	Roadside	3.1.1.1	Collect Urban Traffic Data	Road Traffic
3 Roadside - Traffic	Roadside	3.1.1.2	Monitor Urban Car Park Occupation	Car Parking
4 Traffic Management Centre	Central	3.1.1.3	Provide Urban Traffic Forecasts & Strategies	Traffic Management
5 Traffic Management Centre	Central	3.1.1.4	Manage Urban Traffic Data	Traffic Management
6 Traffic Management Centre	Central	3.1.1.5.1	Provide Urban Traffic Management	Traffic Management
7 Traffic Management Centre	Central	3.1.1.5.2	Provide Planned Urban Traffic Management	Traffic Management
8 Traffic Management Centre	Central	3.1.1.5.3	Provide Urban Car Park States	Car Park Management
9 Traffic Management Centre	Central	3.1.1.5.4	Provide Urban Traffic Speed Management	Control Road Use
10 Roadside - Traffic	Roadside	3.1.1.5.5	Provide Urban Output Actuation	Road Traffic
11 Traffic Management Centre	Central	3.1.1.5.6	Provide Urban Traffic Lane Management	Control Road Use
12 Traffic Management Centre	Central	3.1.1.5.7	Provide Operator Interface for Urban Traffic	Traffic Management
13 Roadside - Traffic	Roadside	3.1.1.5.8	Detect Urban Traffic Violations	Road Traffic
14 Traffic Management Centre	Central	3.1.1.5.9	Manage Urban Static Traffic Data	Traffic Management
15 Traffic Management Centre	Central	3.3.2	Implement Demand Management Strategy	Demand Management
16 Traffic Management Centre	Central	3.3.3	Develop Demand Management Strategy	Demand Management
17 Traffic Management Centre	Central	3.3.4	Manage Demand Data Store	Demand Management
18 Public Transport Vehicle	Vehicle	4.1.1	Estimate Vehicle Indicators	
19 Public Transport Management Centre	Central	4.1.2	Predict Vehicle Indicators	PT Service Monitoring
20 Public Transport Management Centre	Central	4.1.3	Calculate Service Performance	PT Service Monitoring
21 Public Transport Vehicle	Vehicle	4.1.4	Confer to Vehicles	
22 Public Transport Management Centre	Central	4.2.1	Plan & Schedule Services	PT Service Planning
23 Public Transport Management Centre	Central	4.2.2	Plan Vehicle Pooling Services	PT Pooling Services
24 Public Transport Management Centre	Central	4.2.3	Manage Fare Schemes	PT Service Planning
25 Public Transport Management Centre	Central	4.2.4	Manage PT Route Stores and Operator Interface	PT Service Planning
26 Public Transport Management Centre	Central	4.3.1	Provide Service on Demand	PT Service On Demand
27 Public Transport Management Centre	Central	4.3.3	Manage PT Drivers	PT Driver and Fleet Management
28 Public Transport Management Centre	Central	4.3.4	Manage Vehicle Sharing	PT Pooling Services
29 Public Transport Vehicle	Vehicle	4.4.1	Optimise Control Action	

The above Figure shows all of the columns in the spreadsheet. In this example Physical Viewpoint, the scroll bar will have to be used to see all of the Functions.

Figure 37 – An example of a Data Store allocation "report"

A	B	C	D	E
1 Sub-system Name	Sub-system Location	Data Store Id	Data Store Name	Module
2 Traffic Management Centre	Central	D3.1	Urban Traffic Data Store	Traffic Management
3 Traffic Management Centre	Central	D3.5	Demand Data Store	Demand Management
4 Traffic Management Centre	Central	D3.7	Urban Road Static Data Store	Traffic Management
5 Public Transport Management Centre	Central	D4.1	Real Time PT Vehicle Status	PT Service Monitoring
6 Public Transport Management Centre	Central	D4.2	Historical PT Vehicle Data	PT Service Planning
7 Public Transport Management Centre	Central	D4.3	PT Service Plan	PT Driver and Fleet Management
8 Public Transport Management Centre	Central	D4.4	PT Route Static Data	PT Service Monitoring
9 Traveller Interface	Personal Device	D6.1	General Trip Preferences (GTP)	
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

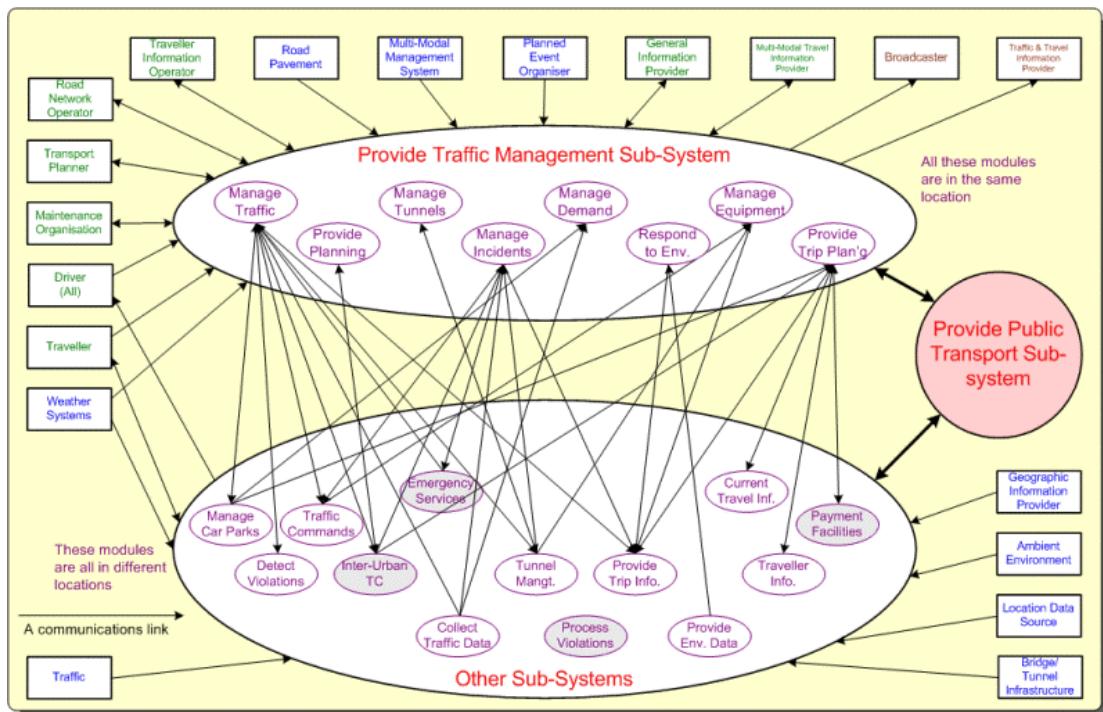
8.7 Producing Diagrams for the Physical Viewpoint

It is often very useful to draw diagrams to illustrate the contents of the Physical Viewpoint. Unfortunately the FRAME Selection Tool does not yet provide the facility to create these drawings. However it is possible to produce them from the "reports".

The most useful ones are those showing the Physical Data Flows between the complete system and the Terminators/Actors (called the "Context Diagram"), between the Sub-systems and between the Modules within the Sub-systems. Sometimes the last two sets of diagrams may show the Physical Data Flows that link the Sub-systems and Modules to the Terminators/Actors.

The easiest way to create these diagrams is with a specialised drawing tool such as Microsoft® Visio®. An example of such a diagram, produced for a "real" ITS Architecture and showing the Physical Data Flows between the Sub-systems, Modules and Terminators/Actors is shown in Figure 38.

Figure 38 – An example of a Sub-system and Module diagram produced for a Physical Viewpoint



(Produced by kind permission of Kent Highway Services, UK)

This particular diagram shows the Modules in two Sub-systems and the Physical Data Flows between them. Links to another Sub-system are shown as single lines. The diagram also shows the Physical Data Flows between the two Sub-systems and the Terminators/Actors. Different colours have been used for the Terminators (blue) and Actors (green).

9. Modifying existing Viewpoints

9.1 *Introduction*

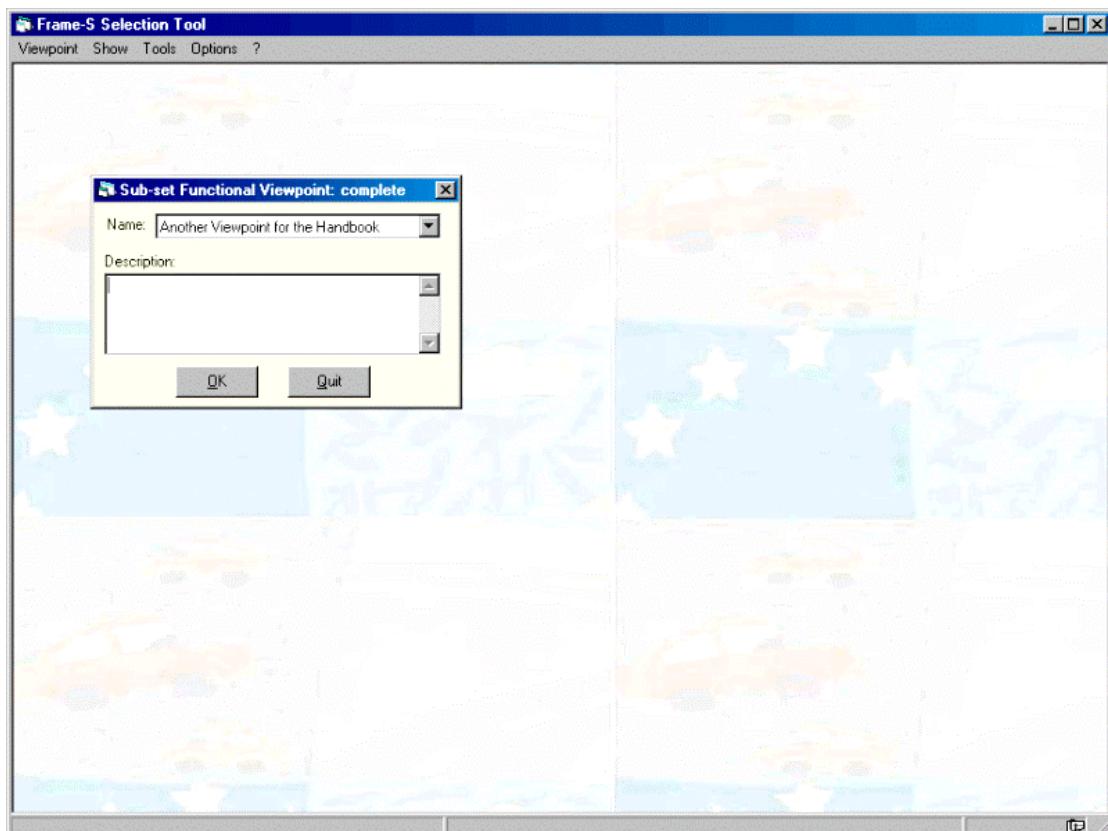
This Chapter describes the ways in which previously created Viewpoints can be modified. The reasons for modifications can be many and varied.

9.2 *Modifying an unused Functional Viewpoint*

To modify a Functional Viewpoint that has not yet been completed, the "OK" button found on all the remaining main screens should be "clicked" until the screen in Figure 23 appears when the "Save and close" option should be "clicked" to return to the "blank" screen. (Note: the screen in Figure 23 will almost certainly show that there are errors. These should be ignored as some if not all of them will be a result of not completing the Functional Viewpoint.)

Clicking on the "Viewpoint" option will produce the display shown in Figure 7 to appear. This time the "Edit previous work" option is selected, which will produce the screen shown in Figure 39 to be produced. (Note: the alternative of using the "Continue previous work" option will return the user to the point at which the Functional Viewpoint creation process was stopped. It does not enable the selections made in the previous steps to be changed.)

Figure 39 – Selecting a previously completed Functional Viewpoint for modification



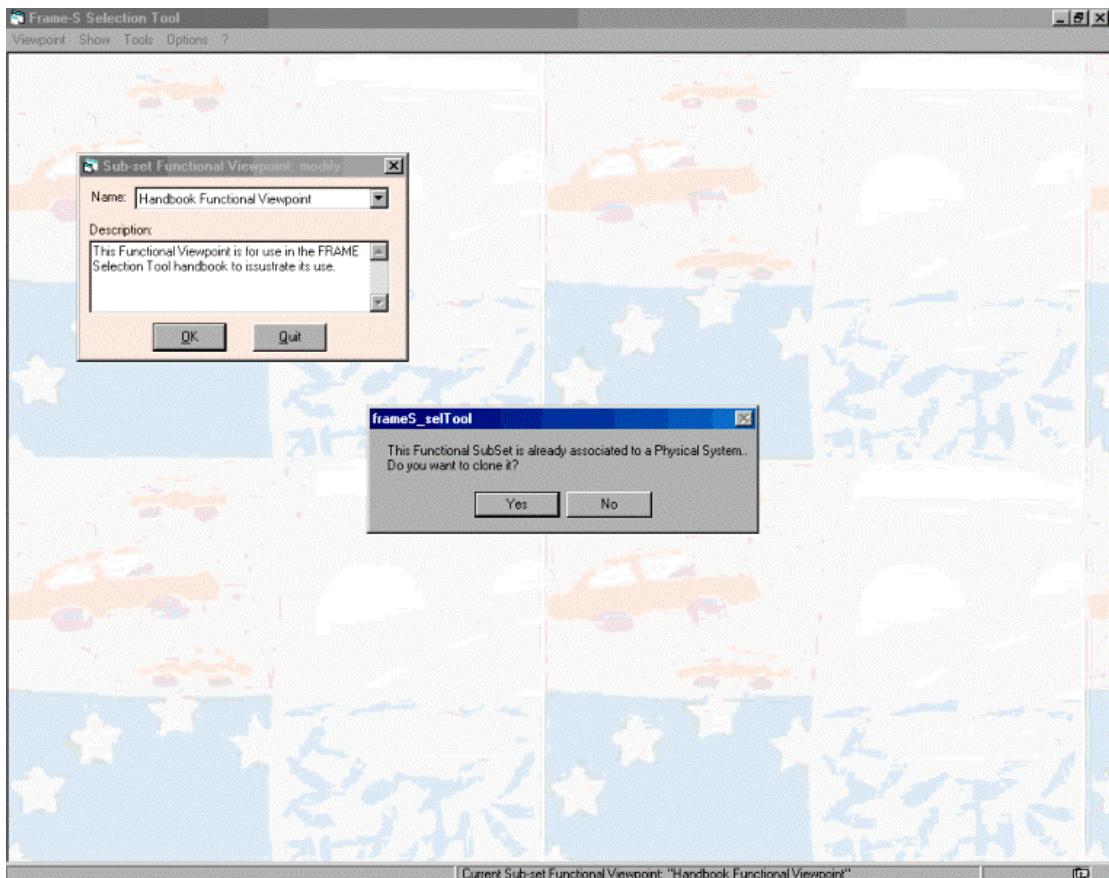
If only one Functional Viewpoint has been created, then it will appear in the "Name" window. Otherwise the down arrow on the right of the "Name" window should be

used to show the Viewpoint that is to be changed. Clicking on the "OK" button makes it possible to restart the creation process for the selected Functional Viewpoint. The steps described in Chapter 6 can then be followed so that any necessary changes can be made to the previous selection of User Needs, Functions, Data Stores, Data Flows and Terminators/Actors.

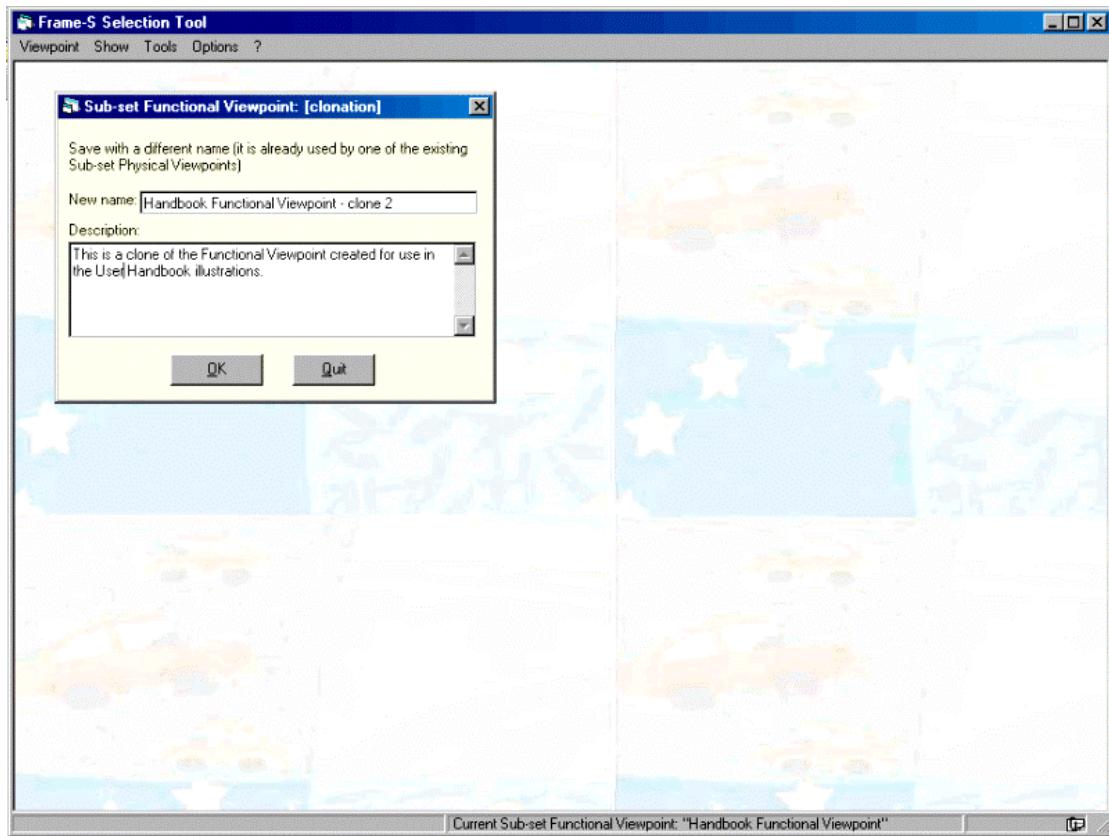
9.3 ***Modifying a used Functional Viewpoint***

It is only possible to modify a Functional Viewpoint if it has not been used as the starting point for the creation of one or more Physical Viewpoints. This is because changing the Functional Viewpoint would invalidate any Physical Viewpoints that had been created from it. However it is possible to create a "clone" of a used Functional Viewpoint so that it can be modified. To do this the "Edit previous work" option in Figure 7 should be selected, which will produce the screen shown in Figure 40.

Figure 40 – Creating a "clone" Functional Viewpoint



What is being offered is the creation of a "clone" of the selected Functional Viewpoint so that changes can be made to it. Clicking on the "Yes" button will cause the screen shown in Figure 41 to appear. Using this screen a name needs to be provided (it must be different to any other Functional Viewpoint names as they may have been user to create Physical Viewpoints) and (optionally) a description for the new Functional Viewpoint.

Figure 41 – naming a cloned Functional Viewpoint

Once this is done and subsequently confirmed, the user will be presented by the Selection Tool with a screen that is similar to that shown in Figure 9. The main difference will be that the User Needs shown in the right-hand part of the screen will be those selected for the original Functional Viewpoint. The User Needs in the left-hand part of the screen will be those that were not selected.

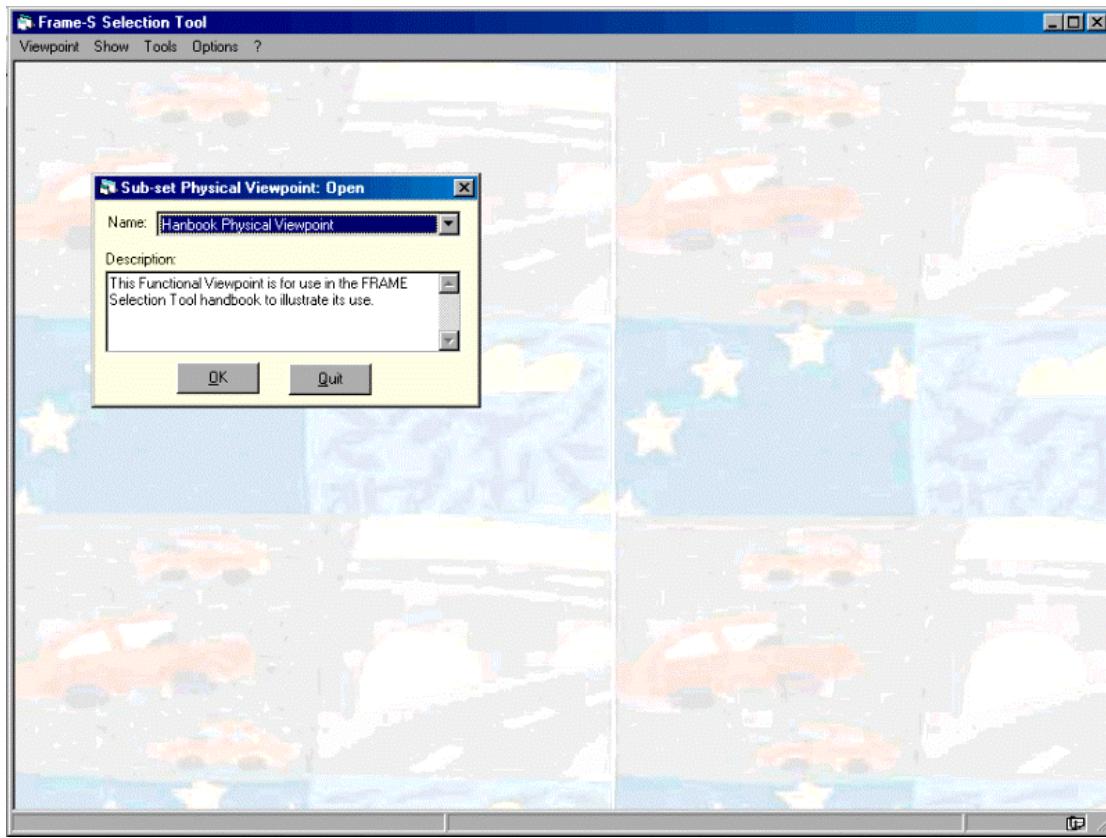
The process of creating the Functional Viewpoint described in sections 6.4 to 6.9 can then be followed. Once it has been completed the new Functional Viewpoint can then be used as the starting point for the creation of one or more new Physical Viewpoints.

9.4 *Modifying the Physical Viewpoint*

If at any point in the process described in Chapter 7 it decided that the previous work on a Physical Viewpoint needs to be changed, the "OK" button found on the remaining main screens should be "clicked" until the end of the Physical Viewpoint creation process is reached. At this point the screen in Figure 31 appears and the "Save and close" option should be "clicked" to return the "blank" screen.

Clicking on the "Viewpoint" option will produce the display shown in Figure 7 to appear. This time the "Edit previous work" option is selected which will cause the screen shown in Figure 42 to be produced.

Figure 42 – Selecting a previously completed Physical Viewpoint for modification



If only one Physical Viewpoint has been created, then it will appear in the "Name" window. Otherwise the down arrow on the right of the "Name" window should be used to show the Viewpoint that is to be changed. Clicking on the "OK" button makes it possible to restart the creation process for the selected Physical Viewpoint. The steps in Chapter 7 can then be followed so that any necessary changes can be made to the previously created Sub-systems and Modules and/or the Functions and Data Stores allocated to them.

Unfortunately there is no process for creating a "clone" of a Physical Viewpoint so that a slightly different version can be produced. One way that this can be done is to use the "reports" option to create records of what is in a Physical Viewpoint and then create a new one from the same Functional Viewpoint, replicating as much of the Sub-systems and Modules, plus the allocation of Functions and Data Stores to them as is required. An alternative is to make a copy the Microsoft® Access® database and use the Selection Tool to modify the particular Physical Viewpoint for which a slightly different version is needed.

10. Hints and Tips for using the FRAME Selection Tool

10.1 *Introduction*

The purpose of this Chapter is to provide some hints and tips for new or inexperienced users of the FRAME Selection Tool. They are organised in alphabetical order of their titles and are based on experiences of other users.

10.2 ***When no errors or warnings are reported, does this mean that the Functional Viewpoint is correct?***

Not necessarily. The checks that are done are *logical* not *semantic*. Thus, for example, whilst the checks will confirm that each Functional Data Flow has a Function, Data Store or Terminator/Actor at each end, it cannot confirm that this particular Functional Data Flow should have been selected for this Architecture.

10.3 ***Can a Functional Viewpoint be modified after it has been used to create a Physical Viewpoint?***

No not directly. Please see section 9.3 for the reasons why and a description of how such a Functional Viewpoint can be modified.

10.4 ***Can a Physical Viewpoint be modified after its creation has been finished***

Yes it can. It is possible to modify a Physical Viewpoint once its creation has been completed. The "Edit previous work" option described in section 5.4 should be used to select the Physical Viewpoint that is to be modified. However this is not recommended as it will destroy the previously created Viewpoint. A better option is to create a modified version of the original Physical Viewpoint as a new Viewpoint using the same Functional Viewpoint as the starting point. For more discussion of this option, please see section 9.4.

10.5 ***Can different Physical Viewpoints be produced from the same Functional Viewpoint?***

It is perfectly possible to produce different Physical Viewpoints from the same Functional Viewpoint. In Step 1 of the Physical Viewpoint creation process (section 7.2) the same Functional Viewpoint as used for another Physical Viewpoint is specified in Figure 25.

For example, if a Physical Viewpoint is to include facilities for both traffic and Public Transport management, should they be Modules in one Sub-system (e.g. a "Mobility Management Centre") or as separate Sub-systems. The Selection Tool will enable either option to be created once a choice has been made.

In fact one of the benefits of using the Selection Tool is that it is possible to create two or more Physical Viewpoints from the same Functional Viewpoint. Thus in the above example both of the options for the location of the different functionalities can be studied.

10.6 *Locations for use in the Physical Viewpoint*

It may not be easy to define an actual set of locations for use by the Sub-systems created in the Physical Viewpoint. The FRAME-S project in its "Workshop Training Materials" defined the following as generic locations:

Central – the place that is used by parts of a System to collect, collate and manage the storage of traffic data, toll payments, freight shipping orders, and/or the generation of traffic management measures, or fleet management instructions, with or without human intervention, e.g. TMC, or TIC, or Freight and Fleet Management Centre.

Roadside – the place that is used by parts of a System for the detection of traffic, vehicles and pedestrians, or the collection of tolls, and/or the generation of traffic management measures, and/or the provision of information and commands to drivers and/or pedestrians.

Kiosk – this is a device usually located in a public place, into which part of the System can be installed to enable Travellers to have limited and controlled access to some of its facilities.

Vehicle – this is a device that is capable of moving through the road network and carrying one or more people (bicycles, motorcycles, cars, Public Transport Vehicles) and/or goods (vans and any other form of road going freight carrying vehicle) in which parts of System can be installed during manufacture or can be added to later.

Freight Device – this is a device in which part of the System can be installed so that it is an integral part of a freight carrying unit, e.g. freight container, trailer, or vehicle body.

Personal Device – this is a device in which part of the System can be installed so that it can be easily used (and possibly carried) by Travellers as one of their personal possessions.

These are generic locations have been chosen because they are unique and are directly relevant to the deployment of ITS. However there may in fact be more than one Sub-system in the same generic location. For example, there may be more than one Sub-system within the generic "Central" location because they can be in different physical places and the characteristics of data flows between them needs to be highlighted.

Therefore it is strongly recommended that Selection Tool users adopt their own names for each of the Sub-systems and their locations. Once a location has been used for one Sub-system, the Selection Tool will prevent it being used for another Sub-system.

10.7 *Locating Data Stores*

Each Data Store has to be specifically located – see section 7.6. Thus it is possible to choose a location in any of the available Sub-systems. However to avoid any potential problems with Data Flows it is recommended that Data Stores are located in the same Sub-systems as the Functions that use them. The situation where a Data Store is in one Sub-system and one or more of the Functions that use it are in another Sub-system should be avoided as it can create problems in the Communications Viewpoint. In the event that this happens it may be necessary to create a number of Physical Viewpoints to investigate the full effects.

10.8 Pre-preparation – Functional Viewpoint

The creation of a Functional Viewpoint is almost a mechanical process, with guidance being provided by the Selection Tool. This process can be helped by deciding in advance which User Needs are to be included. The main constraint in selecting the Functions is deciding what to include in the Viewpoint. For example, the inclusion of some User Needs will cause Functions from the Electronic Payment Functional Area (1) to be included to collect potential payments as part of the creation of trip plans by Travellers. However it may be that this is not seen as something that is to be included in the Architecture.

The way to resolve this issue is to include the relevant Functions in the Functional Viewpoint. In the Physical Viewpoint they can be allocated to what is called an undeveloped Sub-system with only the minimum of Functions allocated to it to avoid consistency errors. This Sub-system can be made to look like a Terminator in the subsequent steps of the ITS implementation process.

The choice of Data Stores, Data Flows and Terminators/Actors to select is a necessary consequence of the Functions that have been selected. The user can either use the Selection Tool to make sure that the right choices have been made (no error or warning messages are produced when this is true) or use the FRAME Browsing Tool to assist with making the correct choices.

10.9 Pre-preparation – Physical Viewpoint

The creation of the Physical Viewpoint demands a lot more pre-preparation than the Functional Viewpoint. This is because the Selection Tool does not provide very much help in the choice of names and locations for Sub-systems and Modules. Thus it is strongly recommended that these are decided before the creation of a Physical Viewpoint is started.

Before starting the allocation of Functions to physical locations in a Physical Viewpoint it is a very good idea to do a rough allocation outside the Selection Tool first. This will enable adjustments and reviews to be made more easily than if the locations have only been set up in the Tool.

Assigning the physical locations outside the Selection Tool can be helped if the "reports" facility is used to produce a list of the selected Functions in a CSV file, which is then converted into a Microsoft® Excel® spreadsheet. The way of doing this is described in section 8.2 and section 8.5. Specifying the locations can then be done either by annotating a print of the Excel file, or adding an extra column to the spreadsheet into which the locations are entered.

10.10 Which Functional Viewpoint "reports" to produce?

As can be seen in Figure 32 there are several "reports" that can be produced by the Selection Tool to show the contents of a Functional Viewpoint. However some are more useful than others.

The most useful is the list of selected Functions, as it can be used to create a first pass of their allocation to physical locations before the creation of the Physical Viewpoint is started. However the "All elements" report is probably the least useful as it is very large and it is not possible to optimise the way that the information is

viewed for each set of elements to enable easy viewing and minimal paper use if a printout is needed.

10.11 Which Physical Viewpoint "reports" to produce?

It is highly likely that all of the Physical Viewpoint "reports" shown in Figure 33 will have to be produced. Examples of them are shown in section 8.5 (Physical Data Flows) and section 8.6 (Allocations of Functions and Data Stores to Sub-systems and Modules).

The production of the first of these "reports" will be useful if diagrams are to be produced to illustrate the contents of the Physical Viewpoint. Their availability is also a requirement for any analysis of the Physical Data Flows that may be required as part of the work to create a Communications Viewpoint. This Viewpoint needs to know about the Physical Data Flows that will pass from one physical location to another, i.e. they cross a physical boundary. Knowing what data is being carried by these Data Flows, other criteria can be assessed which will provide valuable information when tenders are issued for the communications infrastructure needed to support the ITS implementation.

The second of these "reports" is also required as it will provide the input for the descriptions of the entities (or "building blocks") that will be needed for the ITS implementation. When these descriptions are produced, they will need to take account of any relevant User Needs that are in Group 1, as these cover such things as reliability, maintainability, social inclusion, etc. When they have been produced these descriptions should be included in the entity specifications that are issued as part of tenders for their supply.

10.12 Creating Physical Data Flow Names

The names that are assigned to Physical Data Flows are generated automatically by the Selection Tool. An example of this is shown in Figure 35. As can be seen, each name is in fact a number of the form Pxxxxx. In order to keep all of the names unique, a new set is produced each time the Selection Tool creates the Physical Data Flows. This applies even when the new set of Data Flows will replace an identical set previously created for a particular Physical Viewpoint.

Thus it is recommended that users think very carefully about changing the names of the Physical Data Flows, as suggested at the end of section 8.5. In fact it may not be necessary to do this if the Physical Data Flows can be defined by their sources and targets.

10.13 Using the same Function in different Sub-systems or Modules

The Selection Tool does not permit the same Function or Data Store to be used in more than one Sub-system or Module within a Physical Viewpoint. For many Physical Viewpoints this will not be a problem, but for those that it does, the only solution is to create duplicate Functions and/or Data Stores. The precise way of doing this is described in section 14 of this Reference Manual and will need to be associated with the introduction of new Data Flows, as described in section 19.

10.14 Allocating Functions and Data Stores to physical locations

One of the unfortunate "quirks" of the Selection Tool user interface is that when a Function or Data Store has been allocated to a Sub-system the list of Functions or Data Stores returns to its start point. Thus in Figure 28, F3.1.1.1 will appear at the top of the list, even though the Function just allocated was hidden from view and can only be seen by using the scroll bar. This can make the process of selecting and allocating Functions and Data Stores rather tiring and take a long time.

A more convenient method is to press the "down arrow" (\downarrow) key on the computer keyboard after the "OK" button has been "clicked" in the "Allocation of Function and Data Stores" window to confirm the allocation. This will make the next Function to be allocated appear at the bottom of the windows showing the list of Functions or Data Stores.

Another way of speeding up the selection process is to click on the entry in the Sub-system column of either the Function or Data Store list. This will open the "Allocation of Function and Data Stores" window in the same way that clicking on the Function or Data Store number does.

Similar methods can be applied when allocating Functions and Data Stores to Modules – see Figure 30.

10.15 How many Viewpoints can be produced?

There is a limit to the number of Viewpoints that can be created, but for all practical purposes it does not exist. The data for each Viewpoint is provided with a separate identifier within the FRAME Database. Thus the data for one Viewpoint cannot be confused with that for another.

Thus several Physical Viewpoints to be produced from the same Functional Viewpoint, and/or different versions of the same Functional Viewpoint to be created, each with its own set of Physical Viewpoints. This enables the creation of the desired Physical Viewpoint to be achieved as an iterative process, rather than something that is "got right" at the first attempt.

10.16 Are there any constraints on the scope of a Functional Viewpoint

No there are not. It is possible to create a Functional Viewpoint that fulfils all of the User Needs and in consequence uses all of the functionality. Equally it is possible to create a Viewpoint that only includes one particular functional sub-set, e.g. to manage PT operations, to manage car parks.

The main problem will be with the Viewpoints containing functional sub-sets. These functional sub-sets will almost certainly need to exchange data with functionality in other sub-sets that are not included, e.g. to payment facilities. The way to resolve this issue is to include the relevant Functions in the Functional Viewpoint and then allocate them to an undeveloped Sub-system in the Physical Viewpoint. This Sub-system can be made to look like a Terminator in the subsequent steps of the ITS implementation process.

Part 3 Adding extra functionality into the FRAME Database

This part of the Reference Manual contains a guide to the addition of extra functionality into the FRAME Architecture as it is represented in the FRAME Database. This enables the FRAME Selection Tool to be used to create super-sets of the FRAME Architecture for particular ITS implementations. However it will not be possible to see the additional functionality using the FRAME Browsing Tool.

It contains the following nine Chapters:

Chapter 11: Methodology for making additions to the FRAME Database

Chapter 12: General principles for modifying the FRAME Database

Chapter 13: Entering new User Needs

Chapter 14: Entering new Functions

Chapter 15: Entering new Functional Areas and Sub-Areas

Chapter 16: Entering new Data Stores

Chapter 17: Entering new Actors and Terminators

Chapter 18: Entering new User Needs to Function relationships

Chapter 19: Entering new Data Flows

Chapter 20: Actions when all additions to the FRAME Database are complete

11.Methodology for making additions to the FRAME Database

11.1 *Introduction*

This Chapter provides a description of the methodology that must be used to add any extra (and therefore new) entries to the Microsoft® Access® database used by the Selection Tool – called the FRAME Database. These entries may be new User Needs, Functions, Data Stores, Terminators and Data Flows. The descriptions of the way in which these new entries are made are provided in the following chapters and rely on this methodology being followed.

11.2 *Before starting to make additions*

There also some important points to note about the additions that you are about to make to the FRAME Database. These are as follows.

- (a) New User Needs in Group 1 cannot be added. This is because there are no functions related to them and there is therefore no use for them in the FRAME Database.
- (b) Whilst the checking the checking of any logical errors in the new entities that you enter into the FRAME Database will be done by the FRAME Selection Tool, avoiding semantic errors are your problem!
- (c) You should not delete any of the existing entries from the FRAME Database. This is for two reasons: you don't have to select them when using the FRAME Selection Tool and you might need them at some point later in some further additions that you might make at later date.

There are several actions that need to be taken before making adding any new entries to the FRAME Database. They are as follows.

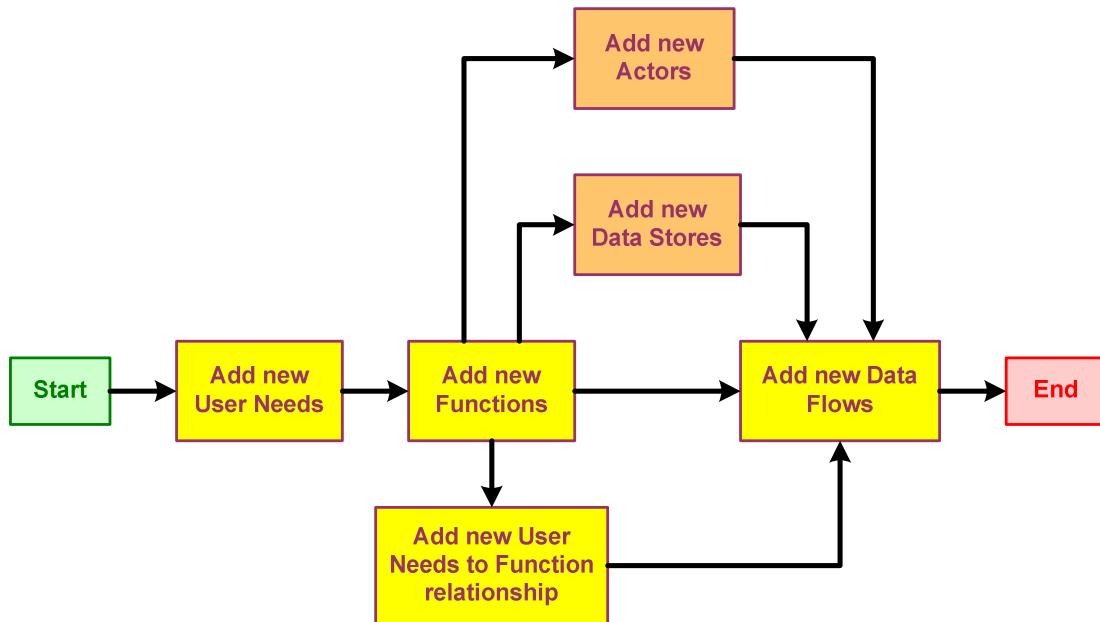
1. Make sure that the “prerequisites” discussed in section 1.4 have been fulfilled.
2. Work out the details of all of the additions that need to be made before starting with the steps that are described in the next chapters. This will include producing all of the descriptive texts for the new entries. In order to accomplish this, it will be necessary to have studied the existing FRAME Architecture using the FRAME Browsing Tool. If necessary it may be prudent to print some of the pages of Function, Data Store, Data Flow and Terminator descriptions, as well as some of the Data Flow Diagrams. The details of any new Functions that may be required can be produced using the template provided in Appendix 4.
3. Either down load a new version of the FRAME Database from the FRAME website – see section 4.2, or make a copy of the existing Database that is to be modified.

Of the above, the most time consuming is the second. However it is very important that all additions are planned in advance. This will enable any potential problems to be resolved before making the additions to the FRAME Database.

11.3 *The process for making additions to the FRAME Database*

The process for making a complete set of additions to the FRAME Database is shown by the diagram in Figure 43.

Figure 43 – Complete process for adding to the FRAME Database



Note that the addition of new Functional Areas and Sub-Areas is included in the box labelled "Add new Functions". However it is entirely possible that none of these will be needed because the new Functions can be added to an existing Functional Area and/or Sub-Area.

Not every ITS implementation will require the addition of all of the different types of entity that are included in Figure 43. For example, it is possible that no new Data Stores and/or Actors will be required, so those parts of the process can be omitted. Whatever is added, the last part of the process is to add the new Data Flows since entry of the data about them require the identities of the new Functions, Data Stores and Actors to be known.

11.4 *Limitations of the additions process*

It is not possible to alter the structure of the FRAME Database. The use of the "forms" described in this Reference Manual will not alter the structure of the Database and it will continue to work with the FRAME Selection Tool after any additions have been made.

It is possible to delete existing entities from the Database. However it is strongly recommended that this is not done, even if those entities are being replaced. If this "rule" is followed, then it will help to minimise any errors that may be created in the course of adding the new entities.

12. General principles for modifying the FRAME Database

12.1 *Introduction*

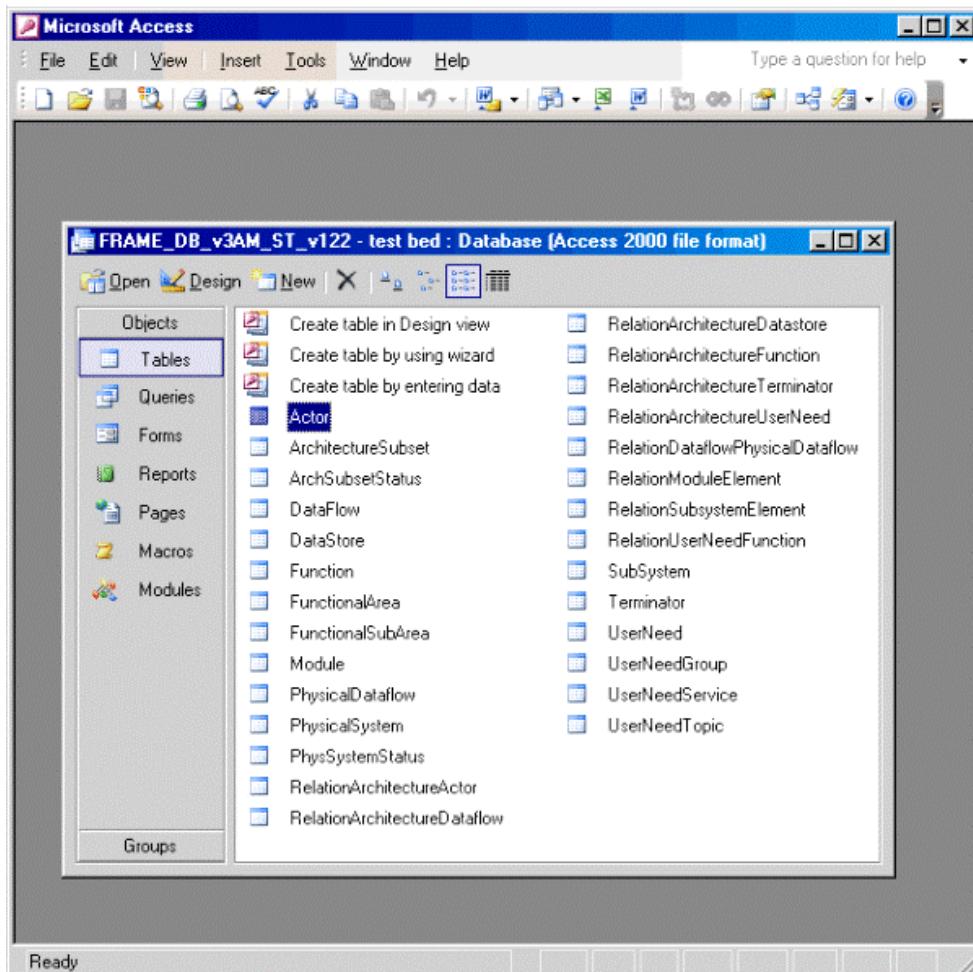
This chapter describes the general points about modifying the FRAME Database using the “Forms” facility in Microsoft® Access®. The individual processes involved in modifying particular parts of the FRAME Database are described in the subsequent chapters.

12.2 *Opening the FRAME Database*

Those readers who are familiar with using Microsoft® Windows XP® can omit this section.

The FRAME Database is opened by “double-clicking” on its file name in the folder where it was extracted from the file that has been downloaded from the FRAME website – see section 4.2. This will produce the initial Microsoft® Access® screen shown in Figure 44.

Figure 44 – Microsoft® Access® Opening Screen Shot



Note that the file name is displayed at the top of the database window in the Microsoft® Access® operating area, rather than at the top of area window.

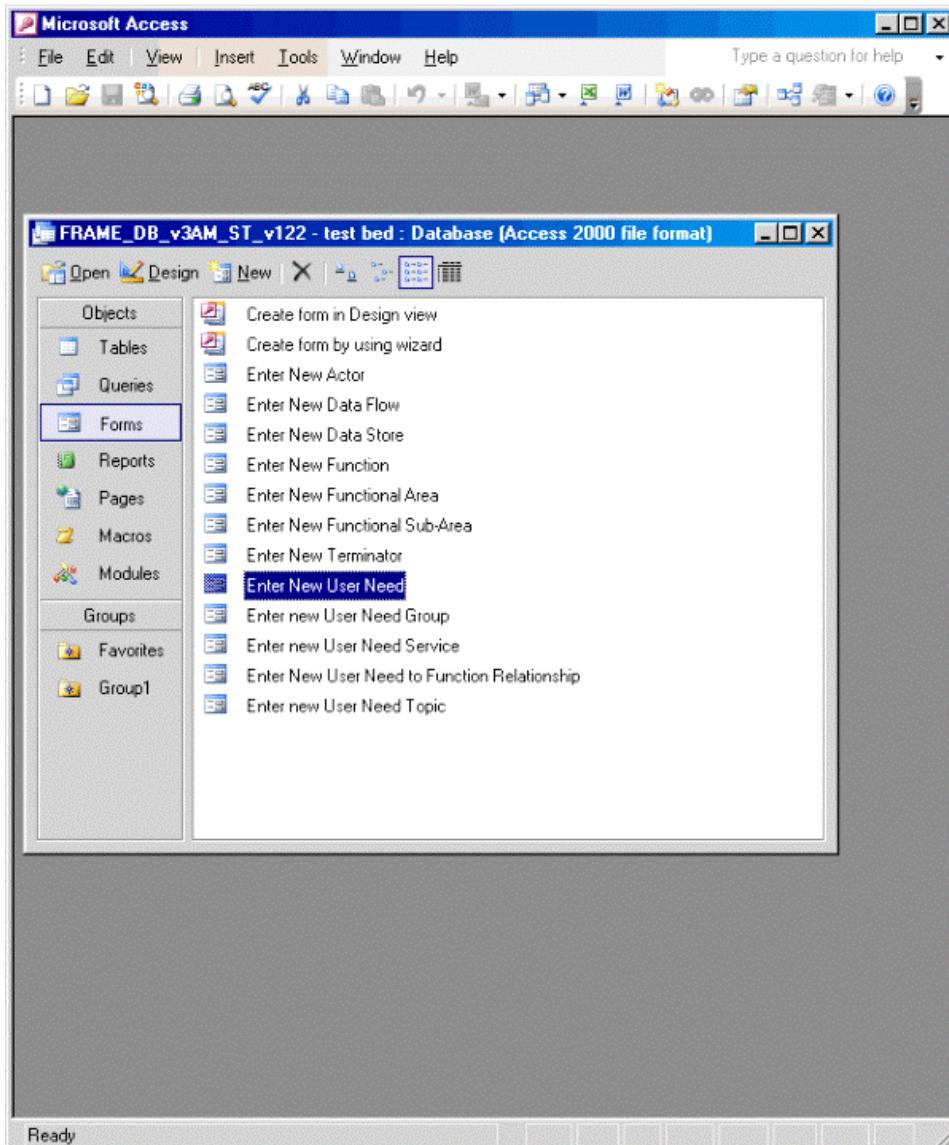
It will be assumed that the Microsoft® Access® window is set to the maximum size for the computer monitor and display resolution being used. For the screen shots shown in the rest of this Reference Manual this was 17 inches (approximately 43 cm) with a display resolution of 1024 x 768 pixels, using the highest (32 bit) colour quality. If other sizes of monitor and display resolution are used then the size of the text and the amount that can be seen in the screen shots may be different.

In the rest of this Manual, the screen shots only show the FRAME Database window. This is to enable them to be seen in the largest possible size.

12.3 **Selecting the “Forms” option**

To be able to use the Forms, click on the “Forms” option on the left hand side of the FRAME Database window as shown in Figure 44. This will produce the “Forms” facility operating window that is shown in Figure 45.

Figure 45 – Microsoft® Access® "Forms" Facility Screen Shot



Further steps in the process of making additions to the FRAME Database are described in the following chapters. These describe the way in which additions can be made to particular groups of Database entries. The order in which the additions are made has been described in section 11.3 and illustrated in Figure 43.

13.Entering new User Needs

13.1 *Introduction*

This Chapter describes the process for entering new User Needs into the FRAME Database. It is part of the overall process of modifying the Database to add extra system facilities that are required by a particular ITS implementation. What is described in the following sections should not be carried out in isolation to the addition of other new entries to the Database. It assumes that the methodology described in Chapter 11 is followed.

13.2 *Opening the new User Need Form to add extra User Needs*

In Figure 45 select the “Enter New User Need” form and “double click” on it. This will produce the User Need input Form shown in Figure 46.

Figure 46 – Adding new User Need screen

The screenshot shows the 'Enter New UserNeed' dialog box. At the top right is a 'Close Form button'. The main area contains several input fields with validation help text:

- User Need Number: 10.1.0.3 [Help: Format: m.n.p]
- User Need Description: The system shall be able to assist PT operators in planning for the optimum use of existing resources to meet the demand. [Help: Free text]
- User Need Topic Number: 10.1.0 [Help: Format: m.n.p]
- User Need Topic Description: Enter new User Need Topic if it does not already exist or edit existing name and/or description.
- Topic Number: 10.1.0 [Help: Format: m.n.p] Topic Name: Objectives [Help: Free text < 30 chars.]
- Topic Description: ??? [Help: Optional - use free text format if required.]
- User Need Service Number: 10.1 [Help: Format: m.n]
- User Need Service Description: Enter new User Need Service if it does not already exist or edit existing name and/or description.
- Service Number: 10.1 [Help: Format: m.n] Service Name: Public Transport Management [Help: Free text < 25 chars.]
- Service Description: ??? [Help: Optional - use free text format if required.]
- User Need Group Number: 8 [Help: Format: m]
- User Need Group Description: Enter new User Need Group if it does not already exist or edit existing name and description.
- Group Number: 8 [Help: Format: m] Group Name: Intelligent Vehicle Systems [Help: Free Text < 20 chars.]
- Group Description: This group contains the functions found within a vehicle, including vision enhancement, longitudinal and lateral collision avoidance, lane keeping, platooning, speed control, driver alertness, 'May Day' initiation, etc. [Help: Free text]

At the bottom of the dialog box are three record navigation buttons: <, >, and <>. Below these buttons is the text: 'Help: All free format text strings should not include spaces'. To the right of the <> button is a red box labeled 'Entities management part of screen'.

The initial screen shown in Figure 46 will show all the information about the first User Need entry in the FRAME Database. To add the new User Needs, click on the select new entry button at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new User Need to be entered. The number next to the new entry button should also increase by one.

13.3 **User Needs numbering system**

The starting point for entering information is the number and description of the new User Need. The number should follow the four digit sequence for User Need numbers, which is defined as follows:

User Needs Number: m.n.p.q, where:

- m – User Needs Group Number
- n – Service Number within the Group
- p – Topic Number within the Service
- q - Unique Number for each User Need within the Group, Service and Topic.

The selection of these numbers for each new User Need is discussed in the following section.

13.4 **Structure of the User Needs in the FRAME Database**

The numbering of the User Needs in the FRAME Database conforms to a structure that groups together the User Needs that are about the same ITS facilities. This structure has 10 Groups, of which Group 1 covers architecture properties and general constraints and is not included in the FRAME Database. The organisation of the remaining 9 Groups is as follows:

- 2 Infrastructure Planning and Maintenance - This group contains the activities associated with long term planning, modelling and reporting as well as the maintenance of the infrastructure;
- 3 Law Enforcement - This group contains the activities associated with the enforcement of traffic laws and regulations, and the collection of evidence;
- 4 Financial Transactions - This group contains the activities associated with the payment for traffic or travel services, and includes the manner of the transaction, its enforcement, and the sharing of revenues;
- 5 Emergency Services - This group contains 'May Day' and stolen vehicle management, the prioritising of emergency vehicles, and hazardous goods incident management;
- 6 Travel Information and Guidance - This group contains all the activities concerned with the handling of pre-trip and on-trip information, including modal choice and change, and route guidance;
- 7 Traffic, Incidents and Demand Management - This group contains the activities associated with traffic control, incident management and demand management, including monitoring, planning, flow control, exceptions management, speed management, lane and parking management, High Occupancy Vehicle (HOV) management, road pricing and access control, and Vulnerable Road Users (VRU) facilities;
- 8 Intelligent Vehicle Systems - This group contains the functions found within a vehicle, including vision enhancement, longitudinal and lateral collision avoidance, lane keeping, platooning, speed control, driver alertness, 'May Day' initiation, etc.;
- 9 Freight and Fleet Management - This group contains all the activities associated with FFM, including statutory data collection and reporting, orders and document management, planning, scheduling monitoring,

reporting and operations management, vehicle and cargo safety, and management of the inter-modal interface;

- 10 Public Transport Management - This group contains the activities associated with public transport (PT), demand responsive PT, shared PT, on-trip PT information and traveller security. It includes management, scheduling, monitoring, information handling, communications and PT priority.

Within each Group are a number of Services that cover particular aspects of ITS facility to which it applies. Within each Service there can be one or more Topics that show various aspects. The current full list of Groups, Services and Topics is provided in Appendix 2 of this Reference Manual.

13.5 Selecting the identifier for each new User Need

The choice of identifiers for the new User Needs will depend on the Group, Service and Topic that the requirements they are defining fit into. This will be a matter of judgement rather than applying a particular rule or rules, other than the simple rule that existing User Needs **must not be replaced**. Thus any new User Needs **must** use new identifiers, regardless of the Group, Service and Topic in which they are entered.

The FRAME-S project produced some guidance on selecting new numbers for User Needs. This is contained in a document called “Guide to Configuration Management (D14)” that can be downloaded from the “LIBRARY” page of the FRAME website at <http://www.frame-online.net>.

13.6 Entering the new User Needs

The starting point for entering the new User Needs is their identifiers and definitions, together with their Topic numbers. Their point of entry in the form is shown in Figure 47.

Figure 47 – Entry of information for a new User Need

The screenshot shows a Windows-style dialog box titled "Enter New UserNeed". It contains three input fields:

- "User Need Number": The value is "8.8.1.1" and the help text is "Help: Format: m.n.p.q".
- "User Need Description": The value is "The system shall enable vehicles to communicate whilst they are moving through the road network."
- "User Need Topic Number": The value is "8.8.1" and the help text is "Help: Format: m.n.p".

If the Topic already exists within the Service and Group then the rest of the form will be filled in automatically with its information plus that for the Service and Group to which it belongs. If not then the Topic information and its Service number must be entered as shown in Figure 48.

Figure 48 – Entry of information a new User Need Topic

Enter New UserNeed	
User Need Number	8.8.1.1 Help: Format: m.n.p.q
User Need Description	The system shall enable vehicles to communicate whilst they are moving through the road network.
User Need Topic Number	8.8.1 Help: Format: m.n.p
Enter new User Need Topic if it does not already exist or edit existing name	
Topic Number	8.8.1 Help: Format: m.n.p
Topic Description	These are the general requirements for vehicle
User Need Service Number	8.8 Help: Format: m.n

Again if the Service already exists within this Group, the rest of the form will be automatically filled in with its information plus that for the Group to which it belongs. If not then the Service information and its Group number must be entered.

Similarly should the Group already exist its part of the form will be filled in with its information. If not then that information will have to be provided.

Once the entry of the information for the first new User Need has been completed, then again click on the select new entry button at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first or previous new User Need, enabling the information for the next new User Need to be entered. The number next to the new entry button should also increase by one.

This process is repeated for every new User Need that needs to be added to the FRAME Database. Note that a new Topic, Service or Group should not be added without adding at least one constituent new User Need.

Once all the information for the new User Needs has been entered, the “close” button at the top right hand corner of the form should be clicked. The new User Need entry form will disappear and the screen will revert to that shown in Figure 45.

14.Entering new Functions

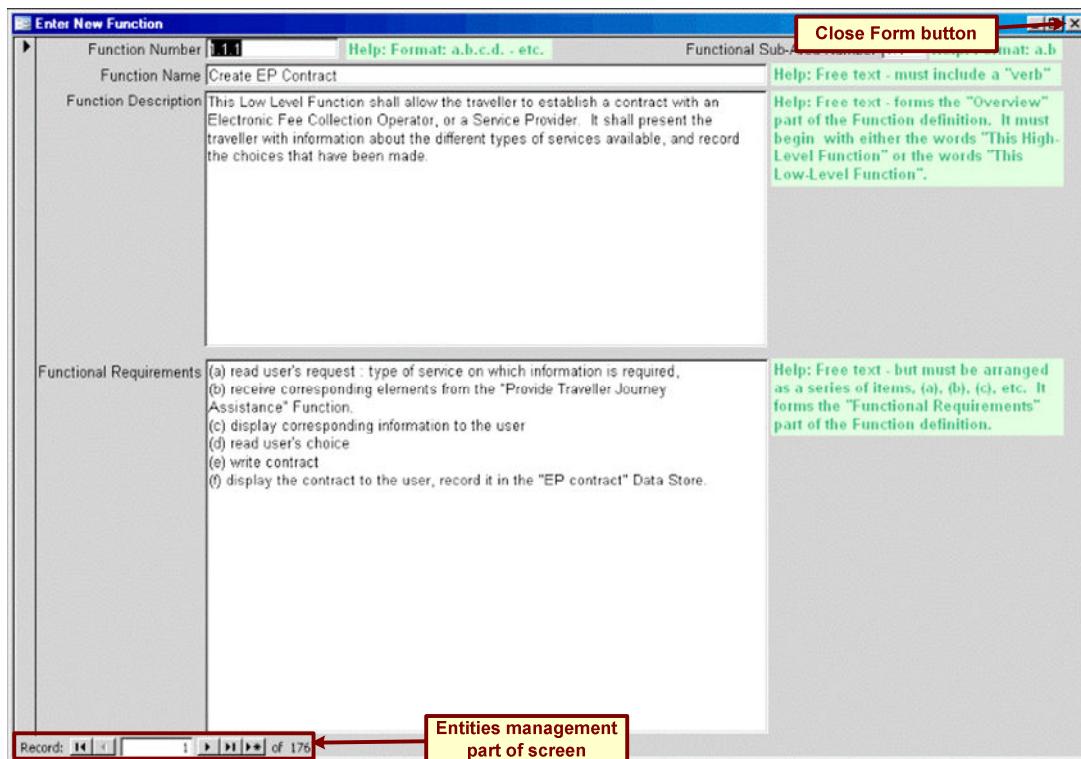
14.1 *Introduction*

This Chapter describes the process for entering new Functions. It is part of the overall process of modifying the FRAME Database to add extra system facilities that are required by a particular ITS implementation. What is described in the following sections should not be carried out in isolation to the addition of other new entries to the Database. It assumes that the methodology described in Chapter 11 is followed.

14.2 *Opening the new Function Form*

In Figure 45 select the “Enter New Function” form and “double click” on it. This will produce the new Function entries Form shown in Figure 49.

Figure 49 – Adding new Function screen



The initial screen shown in Figure 49 will show all the information about the first Function entry in the FRAME Database. To add the new Function, click on the select new entry button  at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Function to be entered. The number next to the new entry button should also increase by one.

14.3 *Function numbering and naming system*

The starting point for entering information is the number and name of the new Function. Functions are numbered and arranged in a strict hierarchical sequence

that starts with the Functional Area. The FRAME Architecture contains 8 Functional Areas which are listed in Appendix 3.

Within each Functional Area there are varying numbers of High-Level and Low-Level Functions. In simple terms a High-Level Function is one that has at least two component Functions. One or both of these may also themselves be High-Level Functions. The highest level of High-Level Function within each Functional Area is called a “Functional Sub-Area”. The definition of a High-Level Function only contains its “Function Description”. At first sight this may not appear to be important, as the FRAME Selection Tool only uses Low-Level Functions. However the FRAME Database contains details of all the High-Level Functions and it is considered important to preserve this for any new Low-Level Functions that are created.

Low-Level Functions have no component Functions. The definition of a Low-Level Function contains its “Function Description” and its “Functional Requirements”, plus other data that is entered through other forms. The FRAME Database only contains information about Low-Level Functions. New High-Level Functions should only be created to preserve the hierarchical structure of the functionality in the FRAME Architecture.

A complete definition of High-Level and Low-Level Functions will be found in section 3.5 of the “Functional Viewpoint (Report 3.1)” that can be downloaded from the FRAME website at <http://www.frame-online.net>. This same document also contains a detailed description of the convention for naming Functions. In simple terms it requires that each name must always contain a “verb”.

14.4 Selecting the number for each new Function

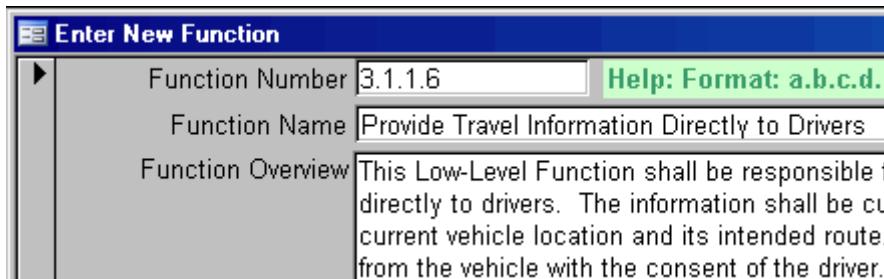
The choice of numbers for new Functions will depend on the Functional Area and/or Sub-Area and/or High-Level Function into which their functionality fits. This will be a matter of judgement based on an analysis of the existing Functions, which can be gained from a study of the FRAME Architecture using the FRAME Browsing Tool. The Tool can be down loaded from its own page on the FRAME website – see section 4.2.

There is however one simple rule which is that existing Functions **cannot be replaced**. Thus any new Functions **must** use new numbers and new names, regardless of the Functional Area, sub-area, or High-Level Function in which they are entered.

The FRAME-S project produced some guidance on selecting new numbers for new functionality. This is contained in a document called “Guide to Configuration Management (D14)” that can be downloaded from the FRAME website at <http://www.frame-online.net>.

14.5 Entering the new Functions

The starting point for entering the new Functions is their numbers, names and definitions. This is shown in Figure 50.

Figure 50 – Entry of information for a new Function

The Functional Sub-Area number is in the top right-hand corner of the form. It contains two digits that are the first two in the Function number. If a new Low-Level Function is being created that is at the highest level within a Functional Area, then its number will only have two digits and these must be also entered as the Functional Sub-Area number.

The Function Overview should then be entered using a similar format to that on each Function page shown by the FRAME Browsing Tool. For a High-Level Function, the Description must start with the phrase, “This High-Level Function shall...” The Low-Level Function description must start with the phrase, “This Low-Level Function shall...”

For Low-Level Functions, the Functional Requirements should be entered in the format shown in part of Figure 5 in the “Functional Viewpoint (Report 3.1)” that can be downloaded from the FRAME web-site – at <http://www.frame-online.net>. Care should be taken to ensure that anyone reading the Requirements can clearly understand what the Function is to do with each of its input Data Flows in order to produce its output Data Flows. As an aid to entry of this information about new Functions a template is provided in Appendix 4.

Once the entry of the information for the first new Function has been completed, then again click on the select new entry button at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Function, enabling the information for the next new User Need to be entered. The number next to the new entry button should also increase by one.

This process is repeated for every new Function that needs to be added to the FRAME Database. Once all the information for all the new Functions has been



entered, the “close” button at the top right hand corner of the form should be clicked. The new Function entry form will disappear and the screen will revert to that shown in Figure 45.

15.Entering new Functional Areas and Sub-Areas

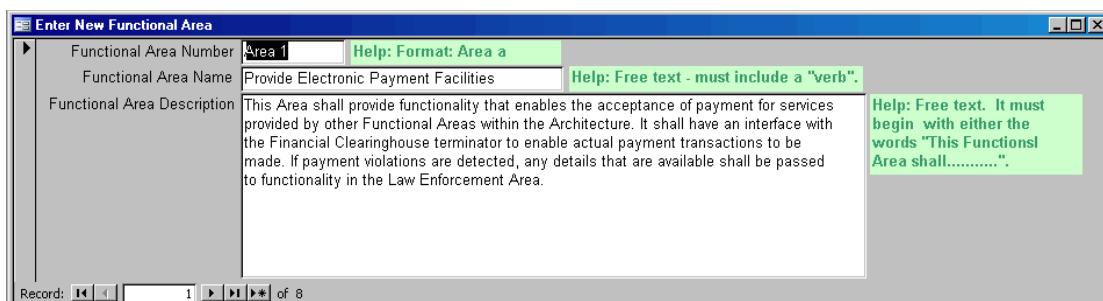
15.1 *Introduction*

This Chapter describes the process for entering new Functional Areas and Sub-Areas as part of the overall process of modifying the FRAME Database to add extra system facilities that are required by a particular ITS implementation. It assumes that the methodology described in Chapter 11 is followed. Any new Functional Areas **must** be added before their constituent new Sub-Areas.

15.2 *Opening the new Functional Area Form*

In Figure 45 select the “Enter Functional Area” form and “double click” on it. This will produce the Functional Sub-Area entries Form shown in Figure 51.

Figure 51 - Adding new Functional Area screen



The initial screen shown in Figure 51 will show all the information about the first Functional Area entry in the FRAME Database. To add a new Functional Area, click on the select new entry button  at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Functional Area to be entered. The number next to the new entry button should also increase by one.

For a new Functional Area, the first thing to enter is its number and is followed by its description. This must start with the phrase, “This Area shall provide functionality...” and contain a short description of what functionality the Area is to provide.

The FRAME-S project produced some guidance on selecting new numbers for new functionality. This is contained in a document called “Guide to Configuration Management (D14)” that can be downloaded from the FRAME website at <http://www.frame-online.net>.

Once the entry of the information for the first new Functional Area has been completed, then again click on the select new entry button  at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Functional Area, enabling the information for the next new Functional Sub-Area to be entered. Again the number next to the new entry button should also increase by one.

This process is repeated for every new Functional Sub-Area that needs to be added to the FRAME Database. Once all the information for all the new Functional Sub-



Areas has been entered, the “close” button at the top right hand corner of the form should be clicked. The new Functional Sub-Area entry form will disappear and the screen will revert to that shown in Figure 45.

15.3 Opening the new Functional Sub-Area Form

In Figure 45 select the “Enter Functional Sub-Area” form and “double click” on it. This will produce the Functional Sub-Area entries Form shown in Figure 52.

Figure 52 – Adding new Functional Sub-Area screen

The initial screen shown in Figure 52 will show all the information about the first Functional Sub-Area entry in the FRAME Database. To add a new Functional Sub-Area, click on the select new entry button at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Functional Sub-Area to be entered. The number next to the new entry button should also increase by one.

For a new Functional Sub-Area, the first thing to enter is its number and is followed by its description. This must start with the phrase, “This High-Level Function shall...” and contain a description of what the functionality in the new Sub-Area will do. Finally the Functional Area Number should be entered, which will cause the remainder of the window to be populated with its details.

The starting phrase in the Sub-Area description is needed because Functional Sub-Areas are usually High-Level Functions. These may themselves have constituent High-Level Functions or Low-Level Functions. Although it is possible for a Functional Area to only contain a set of Low-Level Functions, this is not recommended. It will cause confusion, particularly if further Functions may be added and there is then a

need to create High-Level Functions. Best practice is to add a new Sub-Area, even if it will be the only one in a Functional Area.

Once the entry of the information for the first new Functional Sub-Area has been completed, then again click on the select new entry button  at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Functional Sub-Area, enabling the information for the next new Functional Sub-Area to be entered. Again the number next to the new entry button should also increase by one.

This process is repeated for every new Functional Sub-Area that needs to be added to the FRAME Database. Once all the information for all the new Functional Sub-



Areas has been entered, the “close” button  at the top right hand corner of the form should be clicked. The new Functional Sub-Area entry form will disappear and the screen will revert to that shown in Figure 45.

16.Entering new Data Stores

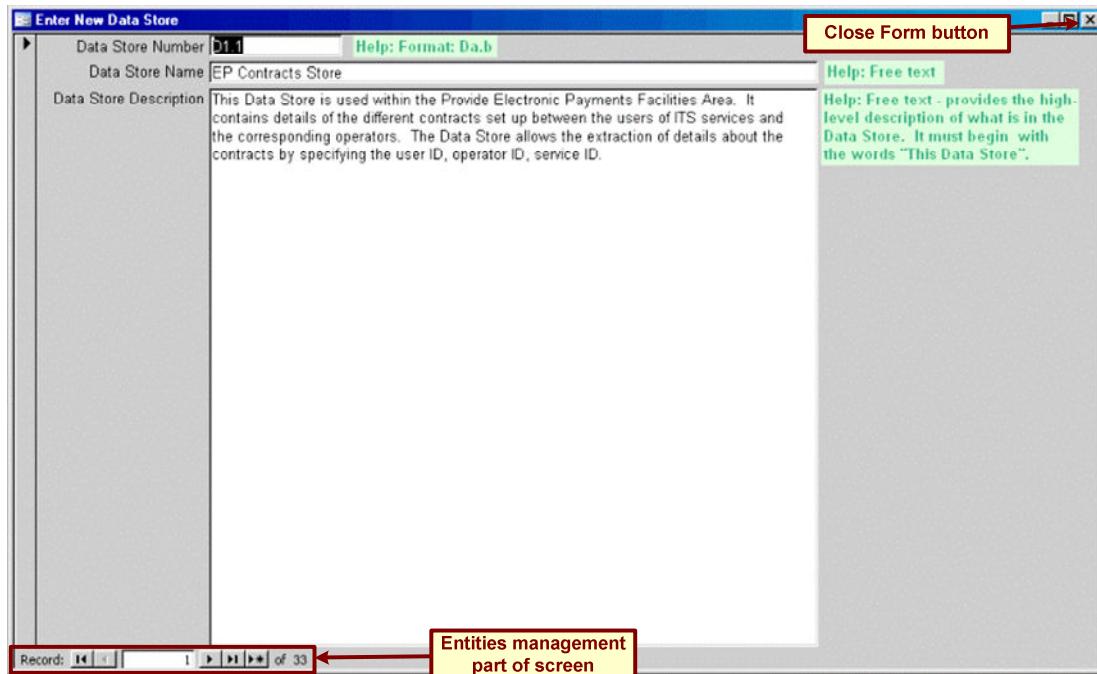
16.1 *Introduction*

This Chapter describes the process for entering new Data Stores as part of the overall process of modifying the FRAME Database to add extra system facilities that are required by a particular ITS implementation. It assumes that the methodology described in Chapter 11 is followed.

16.2 *Opening the new Data Store Form*

In Figure 45 select the “Enter New Data Store” form and “double click” on it. This will produce the Data Store entries Form shown in Figure 53.

Figure 53 – Adding new Data Store screen



The initial screen shown in Figure 53 will show all the information about the first Data Store entry in the FRAME Database. To add a new Data Store, click on the select new entry button at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Data Store to be entered. The number next to the new entry button should also increase by one.

The information required for each Data Store comprises its number, name and description. The format of the number is Da.n, where “a” is the number of the Functional Area in which the Data Store is located and “n” is the next available number within that Functional Area. A study of the FRAME Architecture using the FRAME Browsing Tool (down load it from the FRAME web-site – see 4.3) will enable the next available number to be determined for the Functional Area in which the Store is to be located.

The Data Store Description must start with the phrase, “This Data Store...” What then follows should be a high-level description of its contents. An example is provided in Figure 7 in the “Functional Viewpoint (Report 3.1)” that can be downloaded from the FRAME website at <http://www.frame-online.net>.

Care should be taken to ensure that anyone reading the description can clearly understand what data is to be held by the Store and whether it will be text or numbers. However the description should not contain any details of what database should be used or a proposed database structure. This is left to be decided and defined by the actual organisations that will eventually supply the Data Store.

Once the entry of the information for the first new Data Store has been completed, then again click on the select new entry button  at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Data Store, enabling the information for the next new Data Store to be entered. The number next to the new entry button should also increase by one.

This process is repeated for every new Data Store that needs to be added to the FRAME Database. Once all the information for all the new Data Stores has been

 entered, the “close” button  at the top right hand corner of the form should be clicked. The new Data Store entry form will disappear and the screen will revert to that shown in Figure 45.

17.Entering new Actors and Terminators

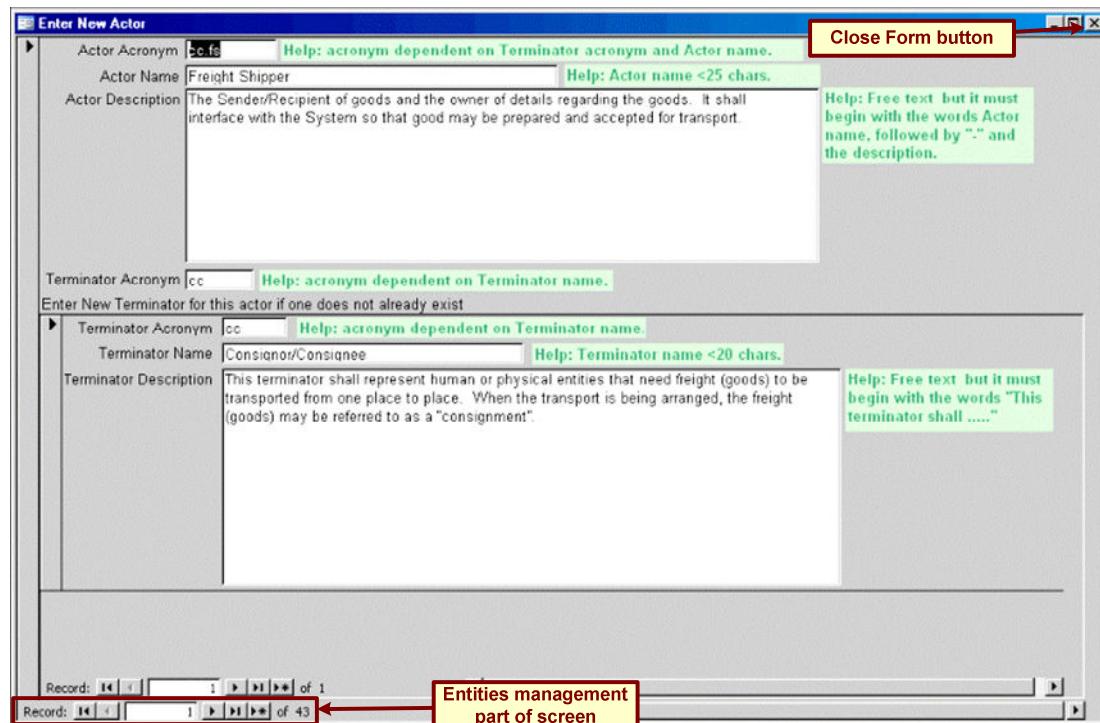
17.1 *Introduction*

This Chapter describes the process for entering new Actors (and Terminators) as part of the overall process of modifying the FRAME Database to add extra system facilities that are required by a particular ITS implementation. It assumes that the methodology described in Chapter 11 is followed.

17.2 *Opening the new Actors Form*

In Figure 45 select the “Enter New Actor” form and “double click” on it. This will produce the Actor entries Form shown in Figure 54.

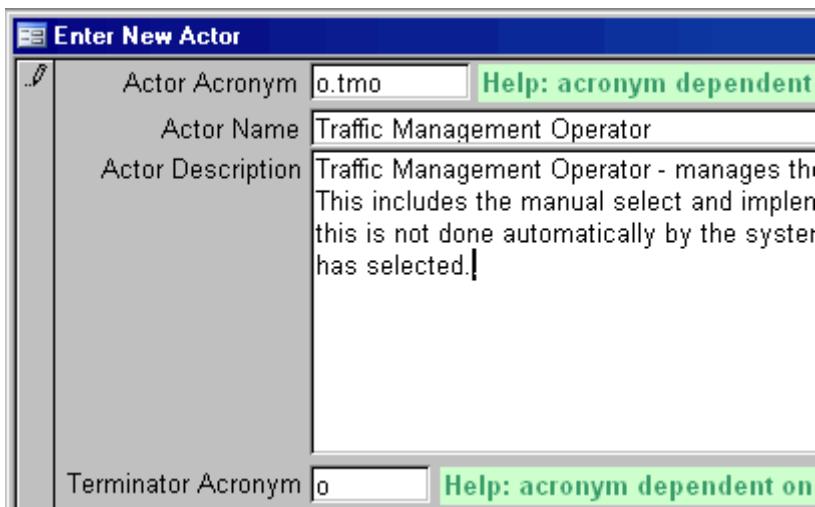
Figure 54 – Adding new Actor screen



The initial screen shown in Figure 54 will show all the information about the first Actor and its Terminator in the FRAME Database. To add a new Actor, click on the select new entry button

 at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Actor to be entered. The number next to the new entry button should also increase by one.

The entry of the information about the first new Actor should begin with the acronym by which it will be known within the FRAME Architecture. This acronym is in two parts and is of the form *aaaa.bbbb*, where “*aaaa*” is the acronym for the Terminator to which the Actor belongs and “*bbbb*” is the acronym for the Actor. Acronyms for Actors are produced by taking the initial letters of the Actor name, as shown by the example in Figure 55.

Figure 55 – Entry of information for a new Actor

The next piece of information to be entered is the Actor's name followed by its description. As can be seen from Figure 54, the Actor Description starts with its name. This is followed by a brief description of what it is and what the System to be created from the Architecture will expect it do. Finally the acronym for the Terminator ("aaaa") to which the Actor belongs is entered.

If the acronym is for a Terminator that already exists then the rest of the form will be automatically filled in with its information. The rest of the form will remain blank if the acronym is for a Terminator that does not exist and will require entry of the information about the new Terminator. This is similar to that for a new Actor, except that the acronym will be of the form "aaaa" and its Description should start with the phrase, "This terminator shall..."

Once the entry of the information for the first new Actor has been completed, then again click on the select new entry button at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Actor, enabling the information for the next new Actor to be entered. The number next to the new entry button should also increase by one.

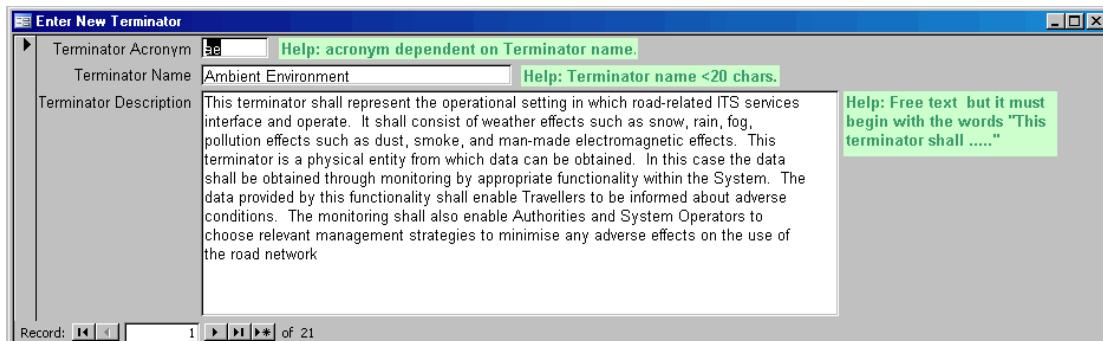
This process is repeated for every new Actor that needs to be added to the FRAME Database. Once all the information for all the new Actors has been entered,

at the top right hand corner of the form should be clicked. The new Actor entry form will disappear and the screen will revert to that shown in Figure 45.

17.3 *Creating new Terminators*

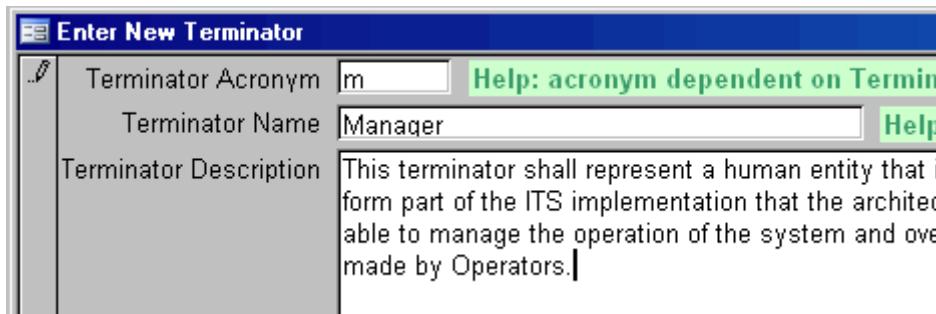
The creation of new Terminators should not be needed as those that already exist should be sufficiently generic for new Actors to be added to them to meet the specific requirements of individual ITS implementations. However if it is found to be impossible to add an Actor to an existing Terminator, a new Terminator must be created.

To create a new Terminator, "double click" on the "Enter New Terminator" form in Figure 45. This will produce the Terminator entries Form shown in Figure 56.

Figure 56 - Adding new Terminator screen

The initial screen shown in Figure 56 will show all the information about the first Terminator in the FRAME Database. To add a new Terminator, click on the select new entry button at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Terminator to be entered. The number next to the new entry button should also increase by one.

The entry of the information about the first new Terminator should begin with the acronym by which it will be known within the FRAME Architecture. This acronym is of the form aaaa. Acronyms for Terminators are produced by taking the initial letters of the Actor name, as shown by the example in Figure 57.

Figure 57 – Entry of information for a new Terminator

Once the entry of the information for the first new Terminator has been completed, then again click on the select new entry button at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Terminator, enabling the information for the next new Terminator to be entered. The number next to the new entry button should also increase by one.

This process is repeated for every new Terminator that needs to be added to the FRAME Database. Once all the information for all the new Terminators has been

entered, the “close” button at the top right hand corner of the form should be clicked. The new Terminator entry form will disappear and the screen will revert to that shown in Figure 45.

18.Entering new User Needs to Function relationships

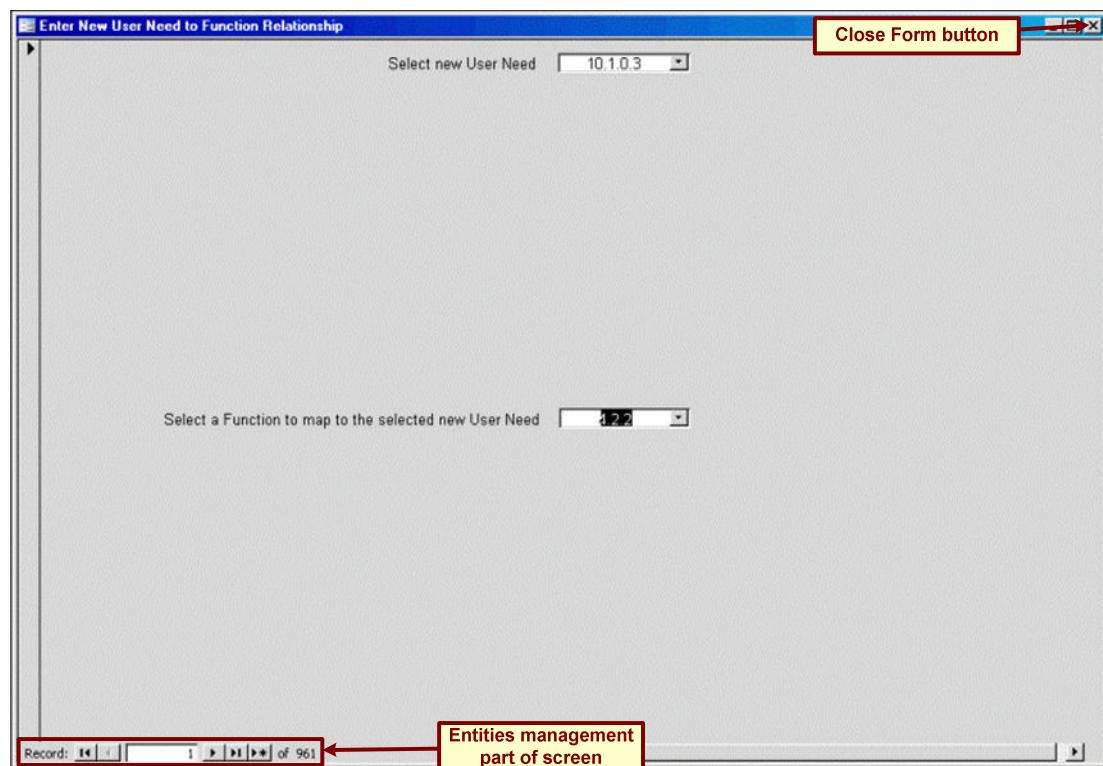
18.1 *Introduction*

This chapter describes the process for entering new User Needs to Function relationships as part of the overall process of modifying the FRAME Database to add extra system facilities that are required by a particular ITS implementation. It assumes that the methodology described in Chapter 11 is followed.

18.2 *Opening the new User Needs to Function relationships Form*

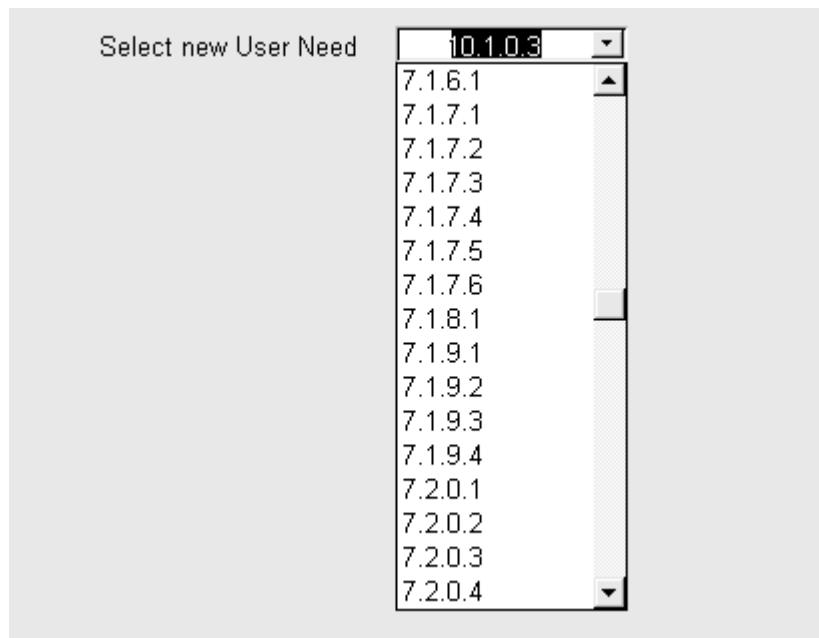
In Figure 45 select the “Enter New User Need to Function Relationship” form and “double click” on it. This will produce the User Needs to Function relationships entries Form shown in Figure 58.

Figure 58 – Adding new User Needs to Function relationships screen



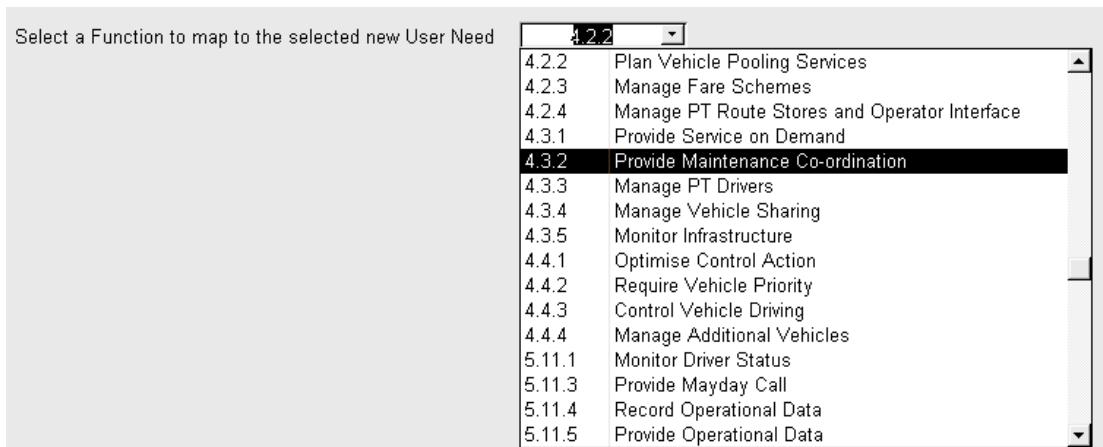
The initial screen shown in Figure 58 will show all the information about the first User Need to Function relationship in the FRAME Database. To add a new relationship, click on the select new entry button  at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new relationship to be entered. The number next to the new entry button should also increase by one.

The form has a large area of “blank” screen because it has to accommodate the drop down lists of User Needs and Functions. So the first step is to click on the small arrow to the right of the “Select new User Need” field. This will produce the drop down list as shown in Figure 59.

Figure 59 – New User Need selection drop down list

The User Need in the relationship should then be selected (use the scroll bar to see more of the list), bearing in mind that the User Needs will appear in a slightly unusual order. Those in Group 10 are at the start of the list and will be followed by those in Groups 2 to 8. Any new User Needs will appear in their correct place in that numerical order.

The process repeated for the small arrow to the right of the “Select a Function to map to the selected new User Need” field. This will produce the drop down list of all Functions shown in Figure 60.

Figure 60 – Function selection drop down list

The Function to which the User Need is to be related should then be selected (again use the scroll bar to see more of the list). Note that the Function names are also provided to make it easier to select the appropriate Function. As there is no Functional Area 10, the Functions will be shown in strict numerical order. Any new Functions will appear in their correct place in the numerical order.

Although new User Needs will require new Functions, it is possible to add new Functions that support existing User Needs. Therefore it is perfectly possible for existing User Needs to be mapped to existing Functions. One instance when this situation might arise is if an existing Function has to be split into two or more lower level Functions in order to make it easier to allocate Functions to Sub-systems and Modules in a Physical Viewpoint.

Once the entry of the information for the first new User Need to Function relationship has been completed, then again click on the select new entry button  at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new User Need to Function relationship enabling the information for the next new relationship to be entered. The number next to the new entry button should also increase by one.

This process is repeated for every new User Need to Function relationship that needs to be added to the FRAME Database. Once all the information for all of the

new relationships has been entered, the “close” button  at the top right hand corner of the form should be clicked. The new User Needs to Function relationships entry form will disappear and the screen will revert to that shown in Figure 45.

19.Entering new Data Flows

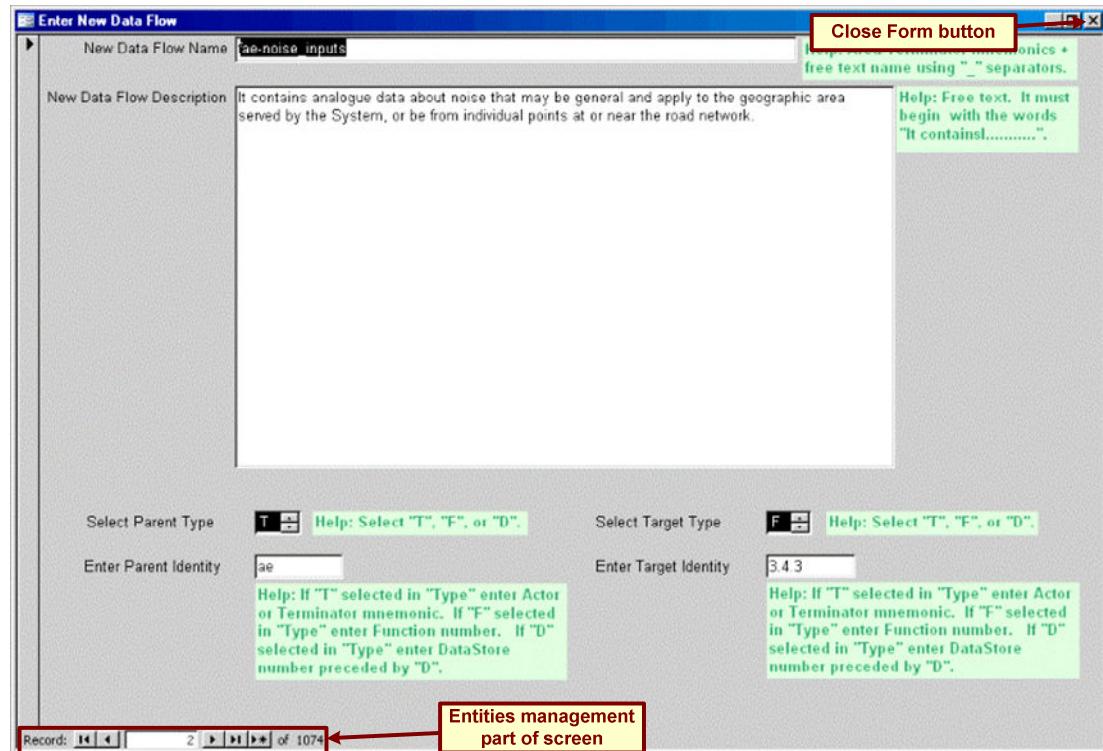
19.1 *Introduction*

This Chapter describes the process for entering new Data Flows as part of the overall process of modifying the FRAME Database to add extra system facilities that are required by a particular ITS implementation. It assumes that the methodology described in Chapter 11 is followed.

19.2 *Opening the new Data Flows Form*

In Figure 45 select the “Enter New Data Flow” form and “double click” on it. This will produce the Data Flow entries Form shown in Figure 61.

Figure 61 – Adding new Data Flows screen



The initial screen shown in Figure 61 will show all the information about the first Data Flow in the FRAME Database. To add a new Data Flow, click on the select new entry button

at the very bottom of the screen. This should blank out all of the information shown on the screen, enabling the information for the new Data Flow to be entered. The number next to the new entry button should also increase by one.

The first item of information to be entered is the name of the new Data Flow, which is in two parts. The first part contains the mnemonic of either the Functional Area(s) in which the Data Flow starts and finishes, or the mnemonic of the origin/destination Actor/Terminator. It must conform to the conventions for Data Flow names that are described in section 3.6 of the “Functional Viewpoint (Report 3.1)” that can be downloaded from the FRAME website at <http://www.frame-online.net>.

The second part of the Data Flow name contains the free text name. Each part of the name is separated with a “_” character. It is advisable to keep the free text part of the name as short as is possible, whilst still providing a rough indication of its contents. The same free text part of the name can be used with different mnemonics if this is helpful

The next item to enter is the Data Flow description, which by convention must start with “It contains...”. As with the Data Store descriptions, the purpose is to provide an indication of what the Data Flow contains, not a detailed specification. Data Flows can have hierarchies in a similar way to Functions, and (as with Functions) the FRAME Selection Tool only uses the lowest level of Data Flow. If it is necessary to have a hierarchy of new Data Flows then the constituents of the higher level Data Flows should be listed in its description.

The origin (Parent) and destination (Target) for the new Data Flow must now be selected. The “Type” fields can be “T”, “F” or “D” and are selected using the up and down arrows to the right of the field. The Parent and Target identities must be entered by typing the actual Terminator or Actor acronym, Function Number or Data Store Number.

Once the entry of the information for the first new Data Flow has been completed, then again click on the select new entry button  at the very bottom of the screen. This should blank out all of the information that was shown on the screen for the first (or previous) new Data Flow, enabling the information for the next new User Need to be entered. The number next to the new entry button should also increase by one.

This process is repeated for every new Data Flow that needs to be added to the FRAME Database. Once all the information for all of the new Data Flows has been

 entered, the “close” button  at the top right hand corner of the form should be clicked. The new Data Flow entry form will disappear and the screen will revert to that shown in Figure 45.

20. Actions when all additions to the FRAME Database are complete

20.1 *Introduction*

Once all the additions to the FRAME Database have been made as described in Chapters 13 to 19 the screen in Figure 45 will have appeared. Microsoft® Access® must be closed by clicking on the "X" symbol in the top right-hand corner of the screen.

20.2 *Checking for errors in the modified FRAME Database*

The only way in which the introduction of errors into the FRAME Database by the addition of any new entities can be checked is through the use of the FRAME Selection Tool. This must be used to create a new Functional Viewpoint using the new entries from the modified Database. There is no error checking within Microsoft® Access® itself beyond the usual "duplication" errors, i.e. you cannot create a new entity that has the same identity as one that is already in the Database.

The use of the FRAME Selection Tool to create a Functional Viewpoint has already been described in Chapters 5 and 6. However in order to check for errors in the modified FRAME Database, it must be selected for use by the Tool. The process for selecting the database is described in section 5.3.

When the steps 1 to 7 in Chapter 6 (see sections 6.2 to 6.8) have been completed the error correction process in step 8 (see section 6.9) will have to be followed. Once the FRAME Selection Tool can find no errors, the new Functional Viewpoint can be used as the starting point for the creation of Physical Viewpoints.

Note that it will not be possible to see the changes to the FRAME Database using the FRAME Browsing Tool. This will continue to show what is in the original Database because it does not use the Database as a source of its information.

20.3 *Making the changes to the FRAME Database permanent*

Using the "forms" described in this Reference Manual will only make changes to a copy of the current version of the FRAME Database that has been downloaded from the FRAME website – see section 4.2. Thus if the Database on the FRAME website is changed to a new version, the changes made using these "forms" will not appear in that version.

To make the changes permanent requires the approval of the FRAME Board, which is the custodian of the FRAME Database. Any requests for permanent changes must be sent to the support facility provided through the FRAME web-site. This can be accessed by sending an e-mail to: info@frame-online.net to which is attached a document containing the full description of the changes. Acceptance is not guaranteed and reasons for acceptance or rejections will be provided to the originator.

Appendix 1 FRAME Database Structure and Relationships

A 1.1 Introduction

The purpose of this Appendix is to provide a description of the structure of the “data parts” of the FRAME Database. Thus it only describes the parts (tables) in the Database that contain data that can be modified by FRAME Selection Tool users.

A 1.2 FRAME Database Tables

A 1.2.1 Introduction

The tables that are in the FRAME Database will be seen when it is first opened using Microsoft® Access®. The opening window in Access® is shown in Figure 62 and provides list of all the tables that it contains. Those that can be changed through the update process are highlighted in the Figure 62. They comprise the following:

Actor	Data Flow
Data Store	Function
Functional Area	Functional Sub-Area
RelationUserNeedFunction	Terminator
UserNeed	UserNeedGroup
UserNeedService	UserNeedTopic

A description of each of these tables is provided in the following sections of this Appendix. The sections are arranged in alphabetical order for ease of reference. Note that the tables use "User Need" instead of "User Need".

A 1.2.2 Table: Actor

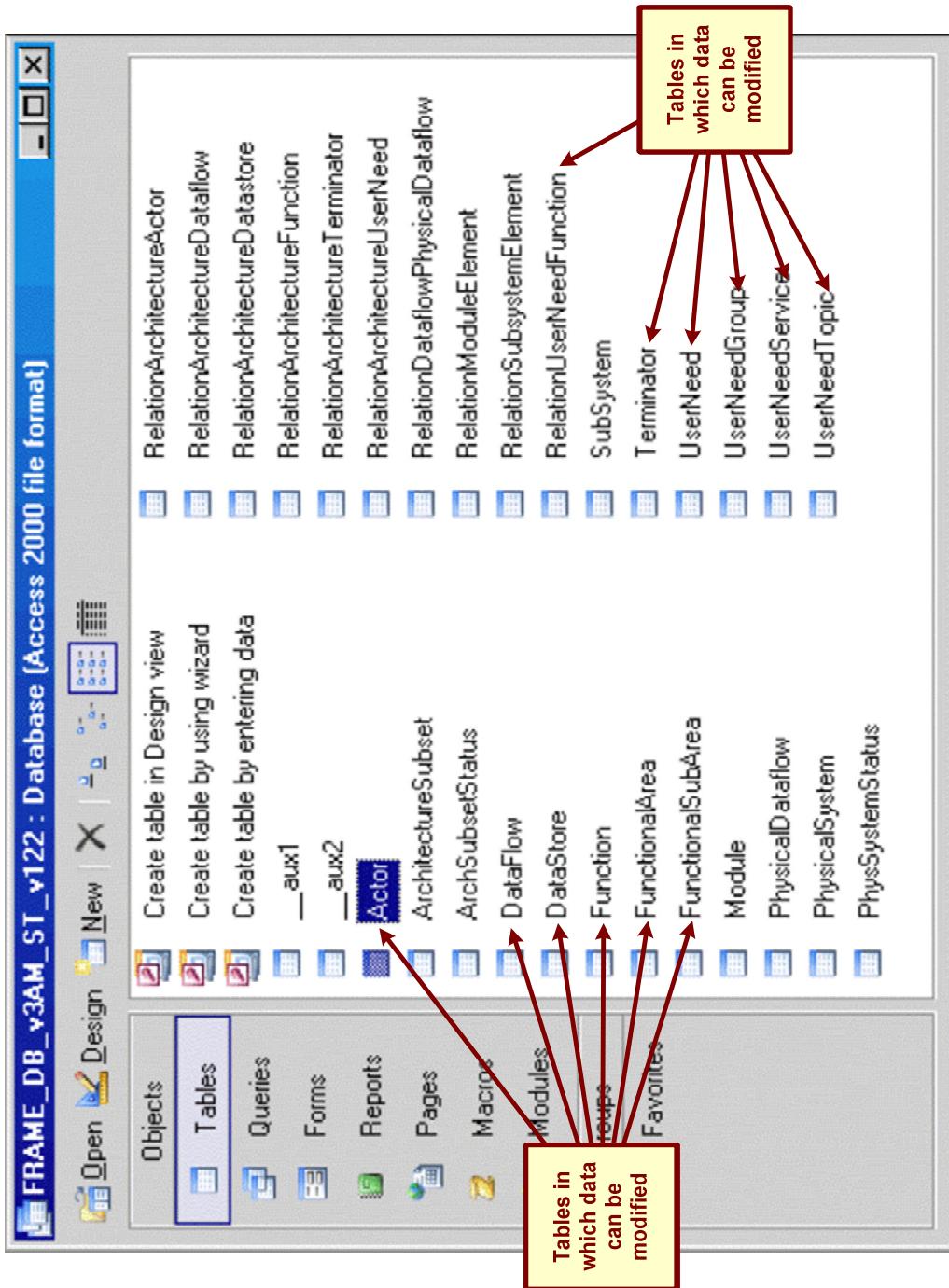
Contains: actorID – the full Actor acronym including that for the Terminator name – the actual Actor name
terminatorID – the Terminator acronym
description – the actual Actor description

A 1.2.3 Table: Data Flow

Contains: dataflowID – a number that is outside the user's control
name – the actual Data Flow name
description – the actual Data Flow description
parentType – may be: “T” for Terminator/Actor
“F” for Function
“D” for Data Store
parentID – may be: Terminator or Actor acronym
Function Number
Data Store Number preceded by “D”
targetType – may be: “T” for Terminator/Actor
“F” for Function

“D” for Data Store
 targetID – may be: Terminator or Actor acronym
 Function Number
 Data Store Number preceded by “D”

Figure 62 – Microsoft® Access® Opening Screen Shot for FRAME Database



A 1.2.4 Table: Data Store

Contains:
 - datastoreID – the Data Store number preceded by “D”
 - name – the actual Data Store name
 - description – the actual Data Store description

A 1.2.5 Table: Function

Contains: functionID – Function Number, Low-level Functions only
name – the actual name
description – the Overview description
subAreaID – the first two digits of the Function number
requirements – the actual requirements

A 1.2.6 Table: Functional Area

Contains: functionalAreaID – “Area” followed by the Area number
name – the actual Functional Area name
description – the actual Functional Area description

A 1.2.7 Table: Functional Sub-Area

Contains: subAreaID – the first two digits of the Function number
name – the actual Functional Sub-Area name
description – the actual Functional Sub-Area description
functionalAreaID – “Area” followed by the Area number

A 1.2.8 Table: RelationUserNeedFunction

Contains: userNeedID – User Need Number
functionID – Function Number, Low-level Functions only

A 1.2.9 Table: Terminator

Contains: terminatorID – the Terminator acronym
name – the actual Terminator name
description – the actual Terminator description

A 1.2.10 Table: UserNeed

Contains: userNeedID – User Need Number
description – the actual User Need description
topicID – the first three digits in the User Need Number

A 1.2.11 Table: UserNeedGroup

Contains: groupID – the first two digits in the User Need Number
name – the actual User Need Group name
description – the actual User Need Group description

A 1.2.12 Table: UserNeedService

Contains: serviceID – the first two digits in the User Need Number
name – the actual User Need Service name
description – the actual User Need Service description (some
are “????” as this data has not been provided)
groupID – the first two digits in the User Need Number

A 1.2.13 Table: UserNeedTopic

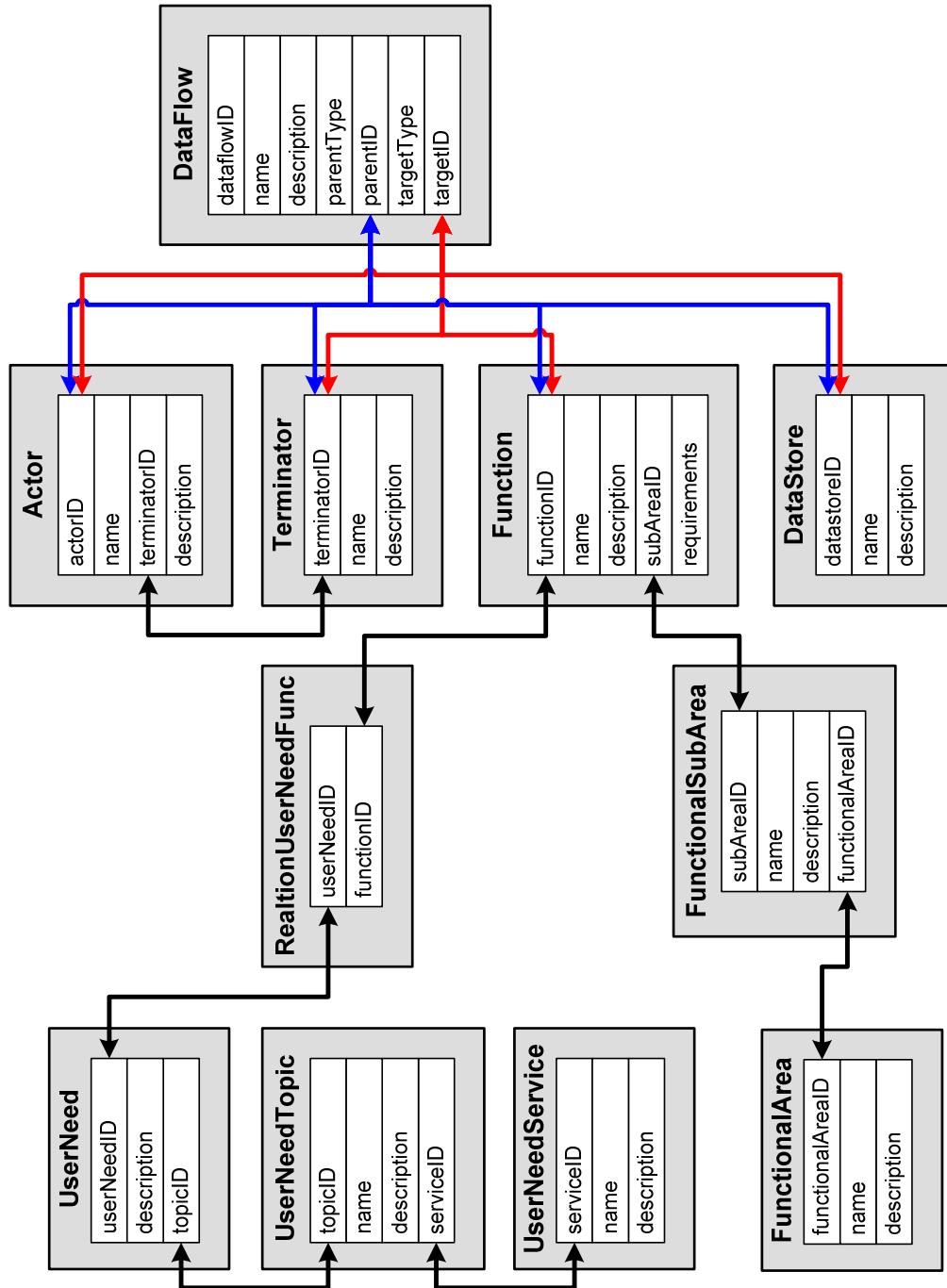
Contains: topicID – the first three digits in the User Need Number

name – the actual User Need Topic name
 description – the actual User Need Topic description (most are “???” as this data has not been provided)
 serviceID – the first two digits in the User Need Number

A 1.3 Relationships between the tables

Various relationships exist between the FRAME Database tables that have been described in the previous sections. These are shown in Figure 63 below.

Figure 63 – FRAME Database Table Relationships



Appendix 2 FRAME User Needs structure

Table 1 shows the structure of the Groups, Services and Topics in the FRAME User Needs and is for **Version 3.2** of the FRAME Architecture and Database. Group 1 is not included in this table as it covers architecture properties and other general requirements not associated with specific functionality.

Table 1 – Groups, Services and Topics in the FRAME User Needs

Number			Group, Service, Topic Title
Group	Service	Topic	
2			Infrastructure Planning and Maintenance
	2.1		Transport Planning Support
		2.1.0	Objectives
		2.1.1	Information Management
		2.1.2	Planning
		2.1.3	Evaluation
		2.1.4	Reporting
	2.2		Infrastructure Maintenance Management
		2.2.0	Basic Services
		2.2.1	Activation
		2.2.2	Monitoring
		2.2.3	Maintenance Units
		2.2.4	Contracts
3			Law Enforcement
	3.1		Policing/Enforcing Traffic Regulations
		3.1.0	Objectives
		3.1.1	Evidence Collection
4			Financial Transactions
	4.1		Electronic Financial Transactions
		4.1.0	Objectives
		4.1.1	Traffic Management
		4.1.2	Revenue Sharing
		4.1.3	Transaction
		4.1.4	Enforcement
5			Emergency Services
	5.1		Emergency Notification and Personal Security
		5.1.0	Basic Services
		5.1.1	Stolen Vehicles
	5.2		Emergency Vehicle Management

Number			Group, Service, Topic Title
Group	Service	Topic	
		5.2.0	Basic Services
	5.3		Hazardous Materials and Incident Notification
		5.3.0	Basic Services
		5.3.1	Incident Management
		5.3.2	Planning
6			Travel Information and Guidance
	6.1		Pre-trip Information
		6.1.0	Objectives
		6.1.1	Modal Choice
		6.1.2	Information Handling
		6.1.3	Traveller Interaction
	6.2		On-trip Information
		6.2.0	Objectives
		6.2.1	Mode Change
		6.2.2	Information Handling
		6.2.3	Traveller Interaction
	6.3		Personal Information Services
	6.4		Route Guidance and Navigation
		6.4.0	Objectives
		6.4.1	Information Handling
		6.4.2	Traveller Interaction
7			Traffic, Incidents and Demand Management
	7.1		Traffic Control
		7.1.0	Objectives
		7.1.1	Monitoring
		7.1.2	Planning
		7.1.3	Traffic Control Centres
		7.1.4	Traffic Flow Control
		7.1.5	Exceptions Management
		7.1.6	O/D Computations
		7.1.7	Speed Management
		7.1.8	Roadside-Vehicle Communications
		7.1.9	Adaptive Traffic Control
		7.1.10	Lane Management
		7.1.11	Parking Management
		7.1.12	Vulnerable Road Users
	7.2		Incident Management

Number			Group, Service, Topic Title
Group	Service	Topic	
		7.2.0	Objectives
		7.2.1	Emergency Services
		7.2.2	Information Management
		7.2.3	Reporting
		7.2.4	Post-Incident Management
		7.2.5	Pre-Incident Management
		7.2.6	Hazardous Goods
	7.3		Demand Management
		7.3.0	Objectives
		7.3.1	Zoning
		7.3.2	Pricing Management
		7.3.3	Parking Management
		7.3.4	Vulnerable Road Users
		7.3.5	Car Sharing
	7.4		Safety Enhancements for Vulnerable Road Users
	7.5		Intelligent Junctions and Links
8			Intelligent Vehicle Systems
	8.1		Vision Enhancement
		8.1.0	Basic Services
	8.2		Automated Vehicle Operation
		8.2.0	Objectives
		8.2.1	Collision Avoidance
		8.2.2	Lane Keeping
		8.2.3	Platooning
		8.2.4	Short Range Communications
		8.2.5	Speed Control
		8.2.6	Supporting Tasks
	8.3		Longitudinal Collision Avoidance
		8.3.0	Objectives
		8.3.1	Collision Avoidance
		8.3.2	Supporting Tasks
	8.4		Lateral Collision Avoidance
		8.4.0	Objectives
		8.4.1	Collision Avoidance
		8.4.2	Lane Keeping
		8.4.3	Supporting Tasks
	8.5		Safety Readiness

Number			Group, Service, Topic Title
Group	Service	Topic	
		8.5.0	Basic Services
		8.5.1	May Day
		8.5.2	Automatic Parking
		8.5.3	Environmental Monitoring
		8.5.4	Accident Data Recording
		8.5.5	Traffic Information & Signs
		8.5.6	Vehicle Information
		8.5.7	Improper Use
	8.6		Pre-crash Restraint Deployment
		8.6.0	Basic Services
9			Freight and Fleet Management
	9.1		Commercial Vehicle Pre-Clearance
		9.1.0	Basic Services
	9.2		Commercial Vehicle Administrative Processes
		9.2.0	Basic Services
	9.3		Automated Roadside Safety Inspection
		9.3.0	Basic Services
	9.4		Commercial Vehicle On-Board Safety Monitoring
		9.4.0	Basic Services
	9.5		Commercial Fleet Management
		9.5.0	Objectives
		9.5.1	Road Freight Management
		9.5.2	Road Freight Fleet Management
		9.5.3	Road Vehicle, Driver, Equipment and Cargo Management
		9.5.4	Freight Distribution
		9.5.5	Inter-Modal Interface
10			Public Transport Management
	10.1		Public Transport Management
		10.1.0	Objectives
		10.1.1	Scheduling
		10.1.2	Monitoring
		10.1.3	Incident Management
		10.1.4	Information Handling
		10.1.5	Communications
		10.1.6	Priority
	10.2		Demand Responsive Public Transport
		10.2.0	Objectives

Number			Group, Service, Topic Title
Group	Service	Topic	
		10.2.1	Information Handling
		10.2.2	Communications
		10.2.3	Route Guidance
		10.2.4	Reporting
	10.3		Shared Transport Management
		10.3.0	Basic Services
	10.4		On-Trip Public Transport Information
		10.4.0	Objectives
		10.4.1	Information Handling
		10.4.2	Traveller Interaction
	10.5		Public Travel Security
		10.5.0	Basic Services

Appendix 3 Functional Areas included in the FRAME Architecture

Table 2 shows the Functional Areas in the FRAME Architecture and is for **Version 3.2** of the FRAME Architecture and Database. Within each Functional Area there are varying numbers of High-Level and Low-Level Functions.

Table 2 – a list of the Functional Areas within the FRAME Architecture

Identity	Name	Description
Area 1	Provide Electronic Payment Facilities	This Area shall provide functionality that enables the acceptance of payment for services provided by other Functional Areas within the Architecture. It shall have an interface with the Financial Clearinghouse terminator to enable actual payment transactions to be made. If payment violations are detected, any details that are available shall be passed to functionality in the Law Enforcement Area.
Area 2	Provide Safety and Emergency Facilities	This Area shall provide functionality that enables the Emergency Services to respond to incidents. The Functions in this Area shall have links with the Manage Traffic Area to enable the reporting and detection of incidents, the management of their impacts and the granting of priority to Emergency Vehicles. It shall be possible for priority to be provided either locally at each controlled point on the road network, or as a "route" through the network. There shall be links to the Provide Traveller Journey Assistance Area to enable priority routes for Emergency Vehicles to be produced.
Area 3	Manage Traffic	This Area shall provide functionality enabling the management of traffic in urban and inter-urban environments. Functionality shall be included to detect and manage the impact of incidents, produce and implement demand management strategies, monitor car park occupancies and provide road transport planning. Links shall be provided to the Provide Safety and Emergency Facilities and Manage Public Transport Areas so that their vehicles are given priority through the road network and to enable assistance to be provided in the implementation of incident and demand management strategies. The External Service Provider terminator shall be sent data about traffic conditions and strategies.
Area 4	Manage Public Transport Operations	This Area shall provide functionality to enable the management of Public Transport. It shall include the scheduling of services and the generation of information that can be made available to travellers. The Area shall have links with the Manage Traffic Area to provide priority for its vehicles, and to provide data on the use of services so that an assessment can be made of demand for different modes of transport. The Manage Traffic Area shall also provide requests for service changes to enable a move towards a better balance in the use of transport modes. There shall also be links to other Areas to provide information about fraud and incidents that have been detected in the Public Transport network.

Identity	Name	Description
Area 5	Provide Advanced Driver Assistance Systems	This Area shall provide functionality that enables the control of vehicles whilst they are using the road network. Interfaces shall be provided to the Provide Safety and Emergency Facilities Area to provide a speedy response to "mayday" calls from vehicles. Vehicle identities shall be provided to other Areas for payment collection and the identification of fraud. Functionality shall also be provided to enable the output of traffic and travel information provided by the Manage Traffic Area.
Area 6	Provide Traveller Journey Assistance	This Area provides functionality that enables the provision of information to all types of Travellers about traffic conditions and about other modes of transport. Functions also provide route determination and guidance, plus travel planning. This includes access to other services such as accommodation.
Area 7	Provide Support for Law Enforcement	This Area shall provide functionality to enable the provision of an interface to Law Enforcement agencies. This interface shall be used to provide information about frauds and violations that have been detected by functionality within other Areas. Examples of frauds and violations shall include but not be limited to invalid or missing payment, speeding, incorrect use of lanes in the road, incorrect observance of other commands sent to drivers. Over-weight vehicles shall be detected by functionality within the Area itself and the details passed to the Law Enforcement Agency.
Area 8	Manage Freight and Fleet Operations	This Area shall provide functionality that enables the management of Freight and Fleet Operations. This shall control the use of freight vehicles and their transportation of goods. The use of other modes of freight transport shall also be supported. An interface to the Provide Safety and Emergency Facilities Area shall also be included to enable the provision of information about hazardous goods. Route planning for this and other types of goods shall be provided through the interface to the Provide Traveller Journey Assistance Area.

Appendix 4 A template for creating new Functions

The template on the following page is designed for use as an aid when creating new Functions. It provides a place in which data for use in the Function, User Need to Functions Relationships and Data Flow forms can be created prior to entry. The data in the template that is used by each of these forms is shown in Figure 64.

Figure 64 – Use of Function Data Entry Template Data

<div style="border: 2px solid red; padding: 5px; width: 100%; height: 100%;"> <p>Function Number: <input type="text" value="F"/></p> <p>Functional Sub-Area Number: <input type="text" value="0"/></p> <p>Function Name: <input type="text" value=""/></p> <p>Function Description (Overview): </p> <p>Function Requirements: <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ... </p> </div>																
<div style="border: 2px solid blue; padding: 5px; width: 100%; height: 100%;"> <p>Parent High-Level Function: <input type="text" value=""/></p> <p>Input Logical Data Flow: <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="2"/></p> <p>Output Logical Data Flow: <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="2"/></p> </div>																
<div style="border: 2px solid green; padding: 5px; width: 100%; height: 100%;"> <p>User Needs served by this Function:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25px;"></th> <th style="width: 25px;">*</th> <th style="width: 25px;">*</th> <th style="width: 25px;">*</th> </tr> </thead> <tbody> <tr> <td>*</td> <td>*</td> <td>*</td> <td>*</td> </tr> <tr> <td>*</td> <td>*</td> <td>*</td> <td>*</td> </tr> <tr> <td>*</td> <td>*</td> <td>*</td> <td>*</td> </tr> </tbody> </table> </div>		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	*	*	*													
*	*	*	*													
*	*	*	*													
*	*	*	*													

The template itself is shown on the following page with no heading of Figure number so that it can be printed and the data entered manually. It is often a quicker and easier way to enter the data this way than to use WORD or other forms of computerised data entry.

Function Number		Functional Sub-Area Number	
Function Name:			
Function Overview:			
Function Requirements: (a) ; (b) ; (c) ; etc....			
Parent High-Level Function:			
Input Logical Data Flows:			
Output Logical Data Flows:			
User Needs served by this Function:			