THE FRAME ARCHITECTURE

A (Short) Overview

Version 1

Peter H Jesty, Richard A P Bossom and Mert Aksaç

December 2020

Co-financed by the European Union
Connecting Europe Facility

1 Introduction

The European ITS Framework Architecture, now more familiarly known as “The FRAME Architecture”, was developed initially during the 1990’s and the first version was published in 2000 (see Figure 1). It is analogous to, but ***not*** the same as, the US ITS Architecture ARC-IT. FRAME has since been updated, and its tools enhanced, by the projects FRAME-S (2001 – 04) and E-FRAME (2008 – 11).

From 2017 the project FRAME NEXT has been continuing this work and, in particular, has revisited the tools that had been developed by the earlier projects. After a review of various possible new tools, it was decided to transfer the FRAME ITS Architecture to *Enterprise Architect* (EA) from SPARX, which also provides additional features.

This document provides a brief introduction to the FRAME Architecture in its new development environment.

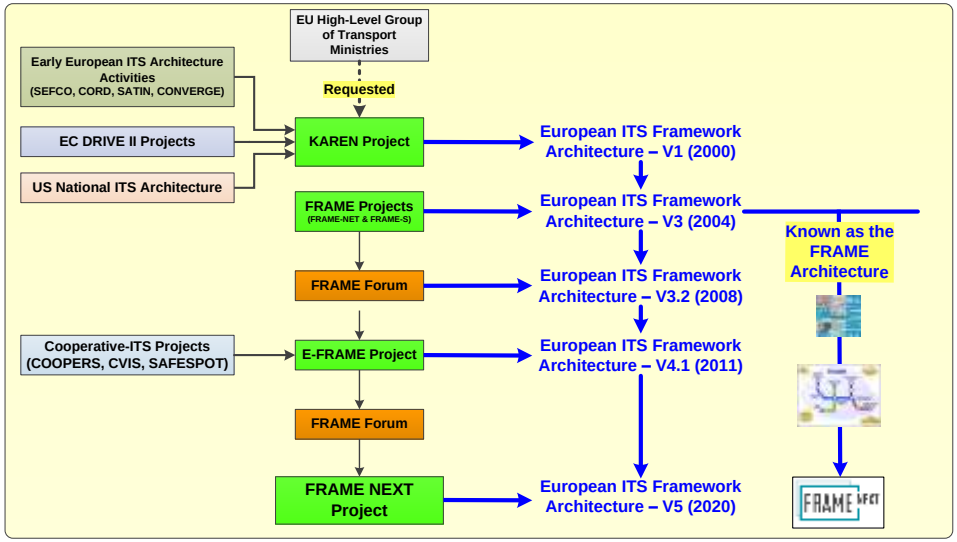

Figure 1 - The history of the FRAME Architecture

2 ITS Architecture Development

The FRAME Architecture covers most ITS applications and shows the relationships between their Functions and Data. Figure 2 shows the principal processes that should be gone through during the creation of an ITS Architecture sub-set for a specific deployment.

The light-blue boxes show the principal results that are obtained through the use of the EA tool, whose basic features are described later in this document. The white boxes show the process that must be done initially (Motivation), and the additional information that may also be obtained for the ITS Architecture sub-set that is created.

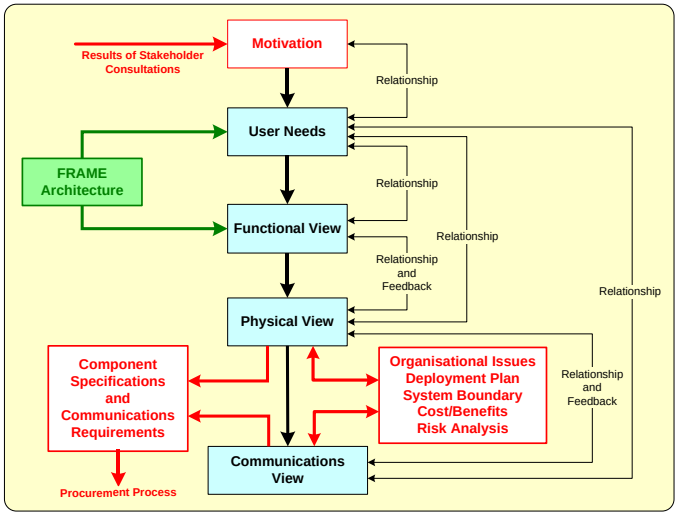

Figure 2 – ITS Architecture Development

Co-financed by the European Union
Connecting Europe Facility

3 Motivation

The ITS Architecture creation process should begin by collecting the aspirations of the various Stakeholders in the development of the new, or revised, set of services. These stakeholders are usually high-level engineers, managers and (sometimes) politicians, and it usually takes some time (months) to arrange the necessary meetings, and to formulate the desired set of ITS Mission Statements and hence the description(s) of the desired ITS Services (see Figure 3).

Figure 3 shows a simplified, and traditional, set of processes. In practice, especially for a large proposal, additional aspects need to be considered (see Figure 4). Traditionally the management of road transport was always the responsibility of the public sector, but in more recent times the private sector is now also included, and hence the requirement for payment and profit, i.e. the Business Outcomes.

Figure 4 shows that these additional factors can have an influence on the later stages of the development and, if it is to go smoothly, and with no expensive 'surprises', these early stages must be concluded properly.

"Time spent in reconnaissance is seldom wasted" (various)

Once the results of the Motivation analysis are known, those that relate to the desired ITS equipment can be developed further. This is done with the assistance of the User Needs.

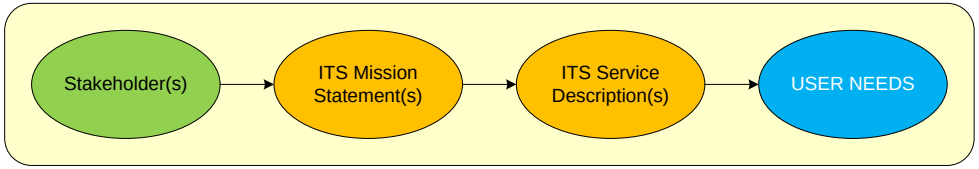

Figure 3 – The High-Level Objectives

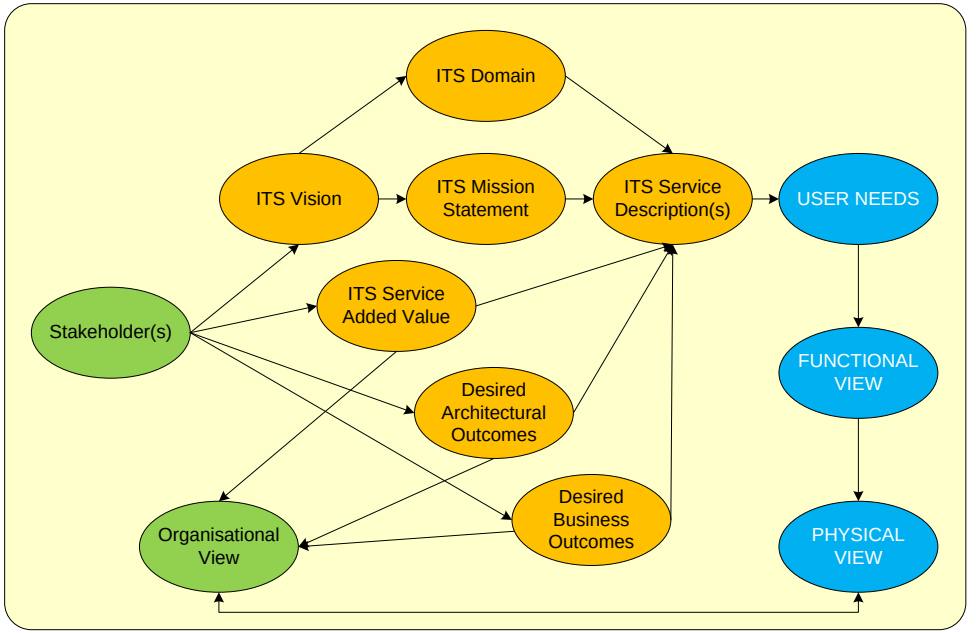

Figure 4 – More Detailed Objectives

Co-financed by the European Union
Connecting Europe Facility

4 User Needs

The results of the Motivation analysis are likely to contain a number of ideas, written in a variety of styles, and which may be open to interpretation. It is therefore necessary to re-write them in a consistent manner, and the User Needs provide a way of expressing these ideas in a way that is Unambiguous, Testable (with objective tests), Traceable (with unique reference numbers) and Singular (with only one idea at a time). They are therefore written in a standard manner – see below.

The User Needs are split into 10 main groups (see Figure 5), which are themselves split further into two further sub-groups (see Figure 6). It will be noted that all the User Needs begin with the words "The system shall …"; this is to ensure that it is clear what the system should provide and can later be tested to confirm it does.

1 General
(not yet added to EA model)
2 Management Activities
3 Policing/Enforcing
4 Financial Transactions
5 Emergency Services
6 Travel Information
7 Traffic Management
8 In-vehicle systems
9 Freight and Fleet Operations
10 Public Transport

Figure 5 – FRAME User Needs Groups

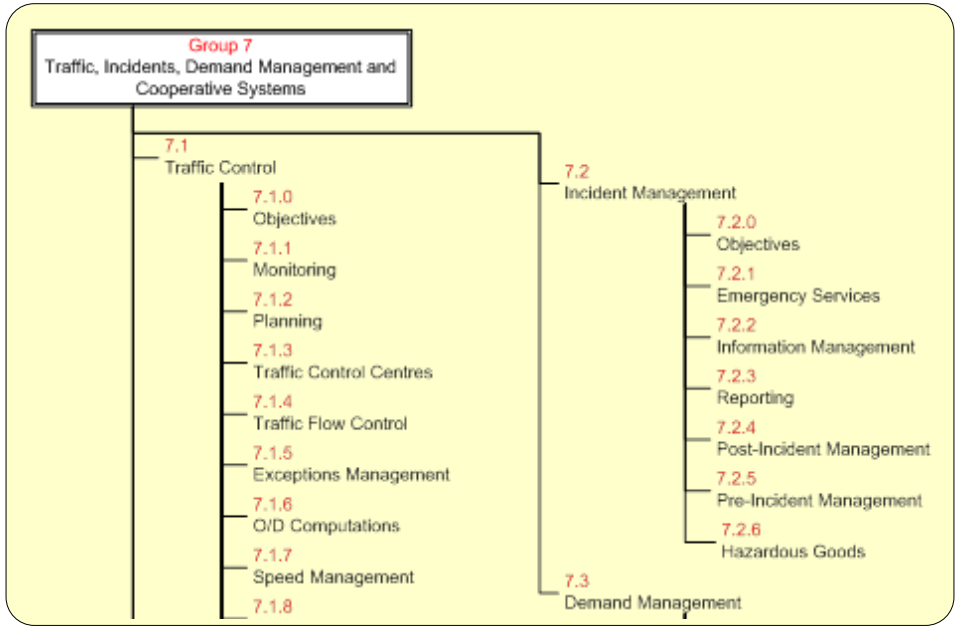

Figure 6 – Example Tree-structure of the Sub-Groups

4.1 Example

The following are the User Needs for a simple Urban Car Park Management system.

7 Traffic Management		
7.1 Traffic Control		
7.1.4 Traffic Flow Control	7.1.4.4	The system shall be able to provide advice to drivers as they approach car parks
7.1.11 Parking Management	7.1.11.1	The system shall be able to monitor the current usage of the parking facilities
	7.1.11.4	The system shall be able to collect and store data from all car parks to provide a historical record

Co-financed by the European Union
Connecting Europe Facility

5 Functional View

5.1 Terminators

The FRAME Architecture is defined using the Function-Oriented design technique of Data Flow Diagrams. These can be defined in an hierarchical manner to cater for ITS of any size or complexity. The highest level in the hierarchy is the Context Diagram (see Figure 7), which shows the relationship between a system and those parts of its environment with which it interacts (known as 'Terminators' – as these are where the influences to/from the system start/finish (or terminate)).

The System Boundary is the 'line' between the ITS under consideration, and for which its developer or owner is responsible, and the people, other organisations (e.g. finance, weather monitoring, other transport modes), and equipment (e.g. vehicle systems) that interact with it.

Terminators can be a person, organisation or another system (e.g. vehicle). It has a description which states what is expected of that terminator. Some Terminators comprise a number of Sub-Terminators (see Figure 8): thus, for example, when a traffic signal is for all drivers the data goes to the Terminator "Driver" (d), but if it is for only one class of driver the data goes to that Sub-Terminator, e.g. "Public Transport Driver" (d.ptd).

Figure 9 is a schematic semi-pictorial diagram for a Tolling System. Whilst is shows all the principal components, they are not in a form that will be easy to describe further in a systematic manner. Meanwhile Figure 10 shows a Context Diagram for the same system, which highlights the Terminators that are required. The description of the "System" can then be expanded further as described in Section 5.2.

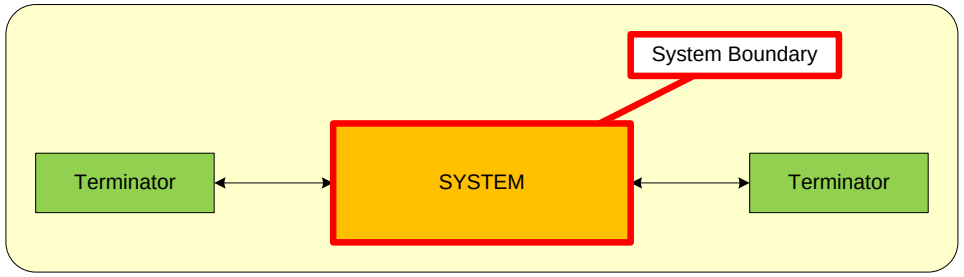

Figure 7 – Context Diagram

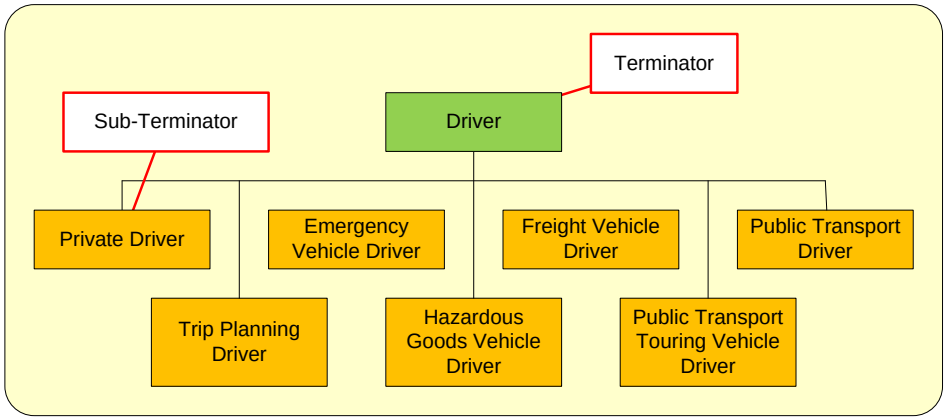

Figure 8 – Example Terminator and Sub-Terminators

Co-financed by the European Union
Connecting Europe Facility

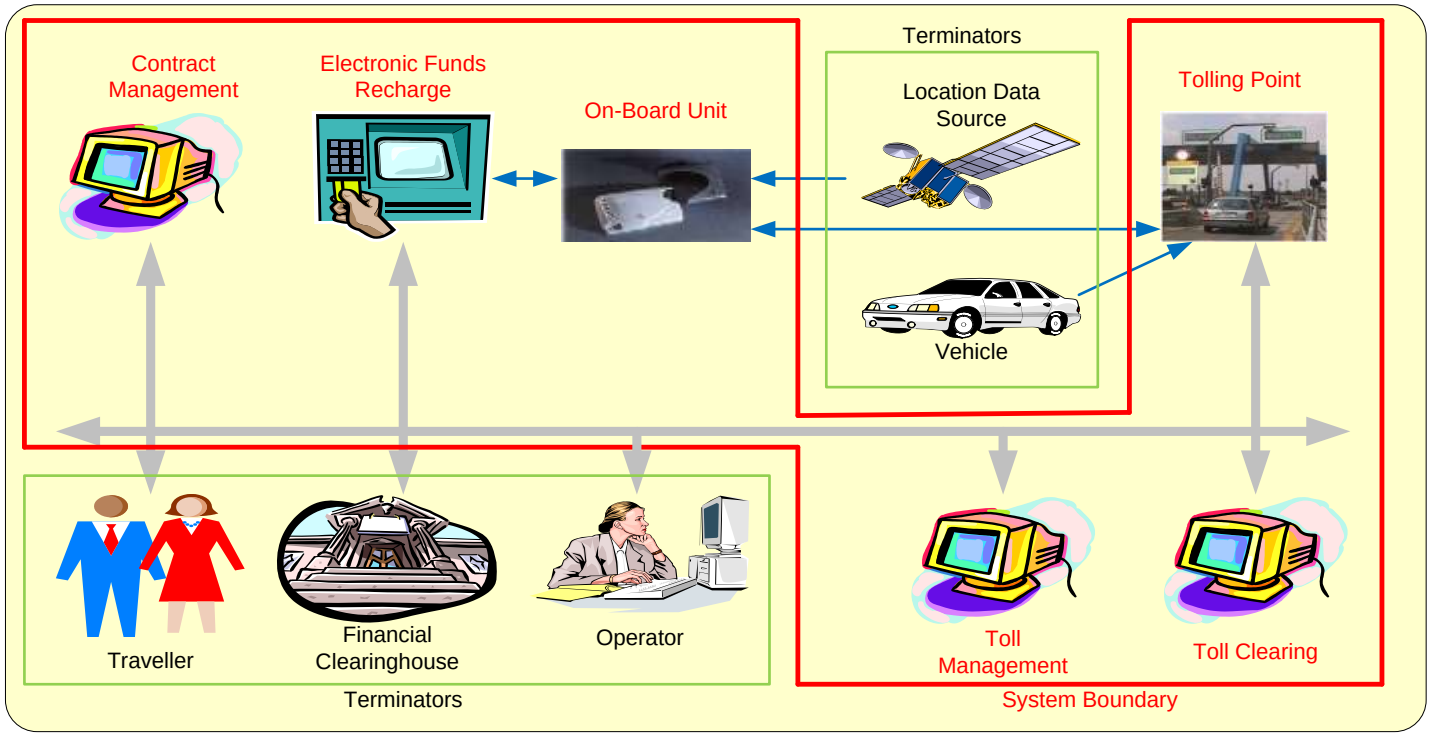

Figure 9 – An example Tolling System

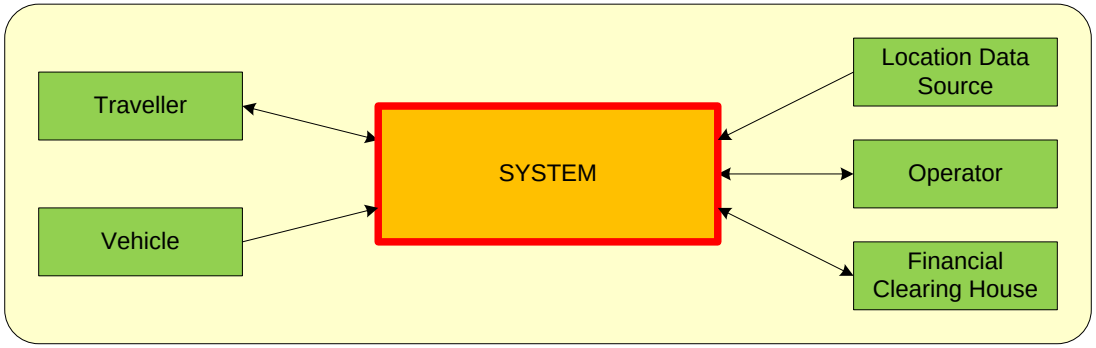

Figure 10 – Tolling System Context Diagram

Co-financed by the European Union
Connecting Europe Facility

5.2 Data Flow Diagrams

Data Flow Diagrams (DFDs) comprise (see Figure 11):

- Functions – they "do" things within the system to fulfil the User Needs. They are defined in terms that are technology independent, i.e. by stating *what* they do, and *not* how they do it.
- Data Stores – they contain data that is stored, either temporarily or permanently, for later use by one, or more functions. Some may be large Relational Data-Bases, but most are not.
- Data Flows – these transfer data between Functions; between Data Stores and Functions; and between Functions and Terminators.

It should be noted that, in order to reduce the complexity of the DFDs, the Terminators are normally only indicated by the existence of a Terminator Data Flow. These can be identified by their name, which begins either with the letter "f" (*from* xyz), or a letter "t" (*to* xyz).

Figure 12 Shows a DFD from the EA Tool showing how the various symbols are displayed.

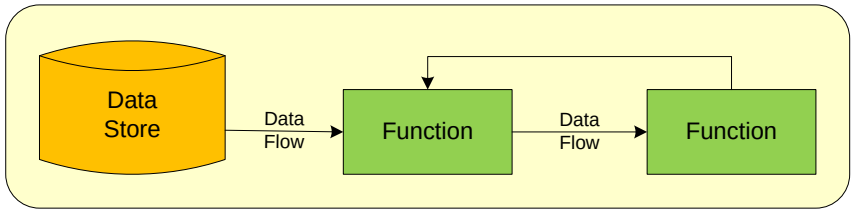

Figure 11 – Simple Data Flow Diagram

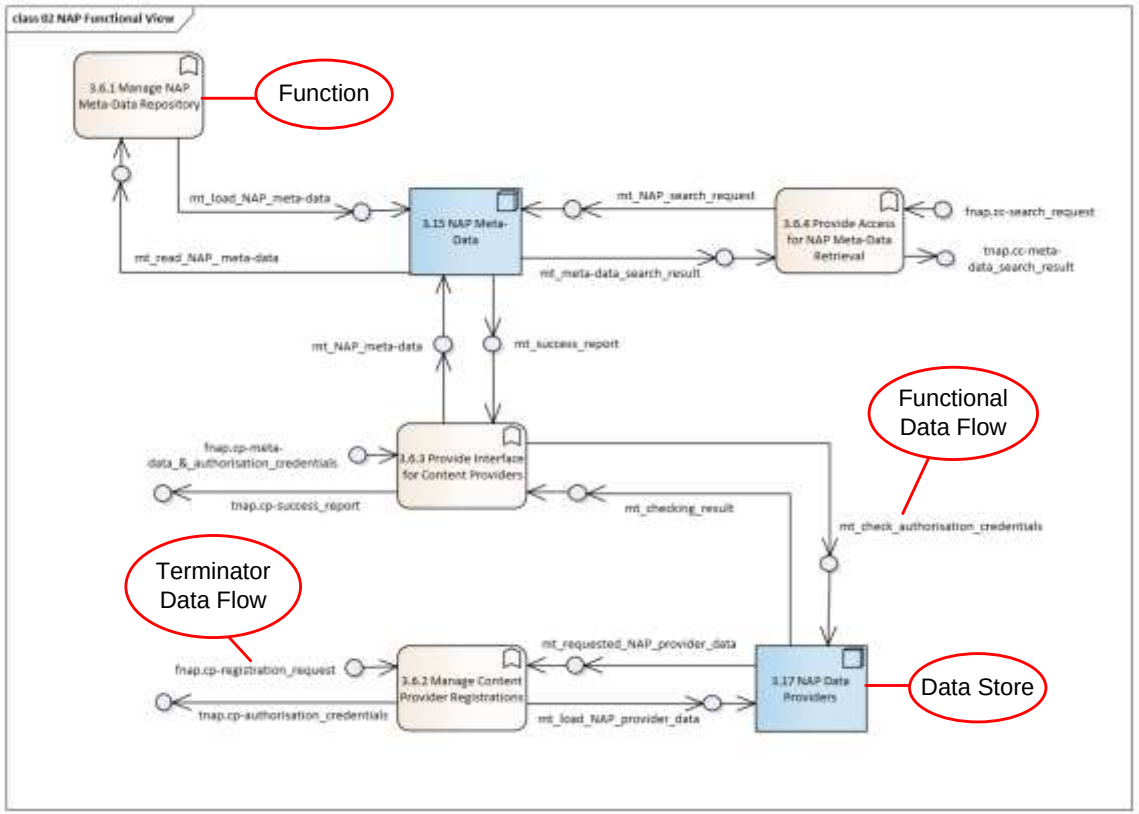

Figure 12 –Data Flow Diagram from Enterprise Architect

Co-financed by the European Union
Connecting Europe Facility

5.3 Functions

The Functions are structured into a hierarchy such there are rarely more than 10 Functions in a single diagram, i.e. the Higher Level Functions provide a mechanism to manage the very large number of Low Level Functions in the entire FRAME Architecture. This is known as Functional Decomposition, and it should be noted that only the lowest level Functions are used in the Physical View (see Section 0).

5.3.1 Functional Areas

At the highest level of Decomposition are the Functional Areas, of which there are nine, as follows (with their mnemonic in brackets):

1. Provide Electronic Payment Facilities (pepf)
2. Provide Safety and Emergency Facilities (psef)
3. Manage Traffic (mt)
4. Manage Public Transport Operations (mpto)
5. Provide Support for Host Vehicle Services (pshvs)
6. Provide Traveller Journey Assistance (ptja)
7. Provide Support for Law Enforcement (psle)
8. Manage Freight and Fleet Operations (mffo)
9. Provide Support for Cooperative Systems (pscs)

The mnemonics are used in the name of the data flows, e.g. mt_urban_road_use_data begins and ends in Area 3; whilst mpto.mt_vehicle_priority_request begins in Area 4 and ends in Area 3.

5.3.2 Functional Decomposition

As all the Areas comprise more than 10 Functions, they are sub-divided into a hierarchy of lower level functions (functional decomposition), using the usual multi-level numbering system. Each function that comprises lower level functions has its own Data Flow Diagram. Thus, for example:

DFD 3 Manage Traffic
Comprises Functions 3.1 to 3.5

DFD 3.1 Provide Traffic Control
Comprises Functions 3.1.1 to 3.1.3

DFD 3.1.1 Provide Urban Traffic Management
Comprises Functions 3.1.1.1 to 3.1.1.5

DFD 3.1.1.5 Provide Urban Traffic Management Facilities
Comprises Functions 3.1.1.5.2 to 3.1.1.5.24

F3.1.1.5.8 Detect Urban Traffic Violations
This is a Low Level Function that may be used in Physical View. Each one has a detailed description of what the function does (for use in specifications), and (normally) a list of one, or more, User Needs that this Function helps to satisfy.

5.3.3 Configuration Management

It will be noted that not all DFD and Function numbers are actually visible in the latest version of the FRAME Architecture. This is because if, during maintenance, a Function has to be modified, or removed, its replacement is given a new number in case an existing architecture has already used the old one.

Co-financed by the European Union
Connecting Europe Facility

5.4 Data Flows

A consequence of the Functional hierarchy is that the Data Flows also have a hierarchy, with those in the higher levels often being composed of more than one low level Data Flow (see Figure 13). This shows that Data Flows C, E and G are used (only) to make the diagram easier to understand, with only A, B, D and F being available for selection when creating a Physical View (see Section 0).

There are a number of naming conventions are used for Data Flows, depending on where they in the functional hierarchy, and their use.

- Data Flows within a Functional Area begin with the initials of that area:
  e.g. Manage Public Transport Operations : *mpto_veh_in_alarm*
- Data Flows between Functional Areas begin with the initial of both areas, the source being first:
  e.g. from Provide Support for Host Vehicle Services to Manage Traffic : *pshvs.mt_rest_area_eta*

The names of Terminator Data Flows have a different convention

- At the Top Level the full (Sub-)Terminator name is used:
  e.g. *To/From Driver*
- Medium Level Terminator Data Flows use the initials of both the (Sub-)Terminator and the relevant Functional Area, with “t” or “f” for ‘to’ and ‘from’ respectively:
  e.g. *td-pepf_payment_request*
- Low Level Terminator Data Flows just use the initial of the (Sub-) Terminator, with “t” or “f” for ‘to’ and ‘from’ respectively:
  e.g. *td.ptd-scheduling*

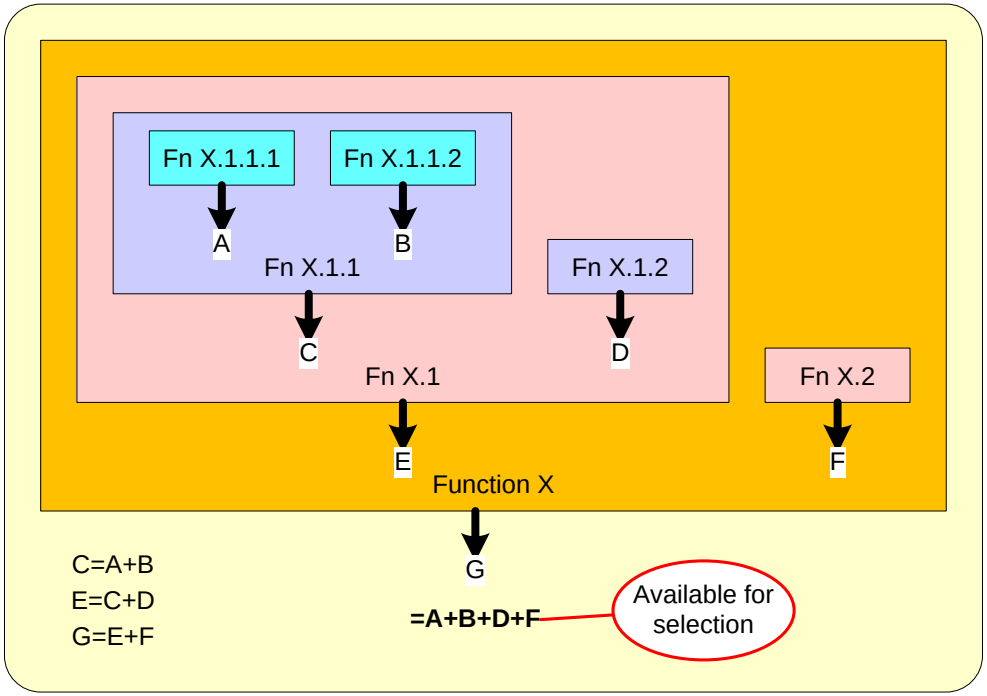

Figure 13 – Data Flow Decomposition

Co-financed by the European Union
Connecting Europe Facility

5.5 Creating an ITS Architecture Sub-set

The choice of which Low Level Functions to use is determined, initially, through the selection of the User Needs (see Section 4) that best describe the required system (see Section 5.3.2 and Figure 14). Occasionally, especially for a novel, or unusual ITS, the relevant User Needs may not exist and it is then necessary to add the relevant Function(s).

The User Needs, however, only reference the principal Low Level Functions that may be required, and it is first necessary to confirm those that have been referenced, and then to add the additional Low Level Functions, Data Stores (see Section 5.2), Data Flows (see Section 5.4) and Terminators (see Section 5.1), that are required to create the overall desired system.

Once a consistent sub-set Functional View has been created, one can then move to the creation of the desired Physical View (see Section 6).

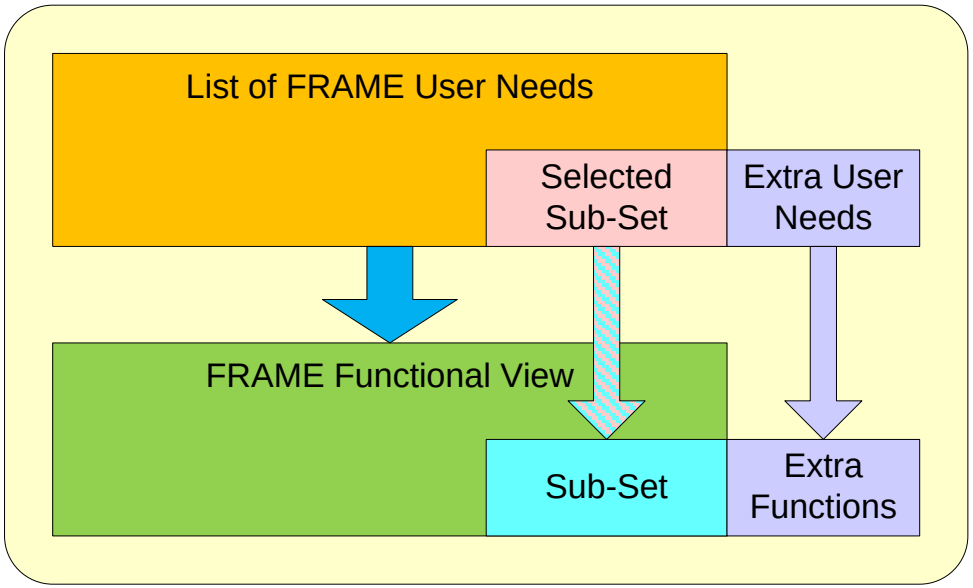

Figure 14 – Selecting a Sub-set Functional View

Co-financed by the European Union
Connecting Europe Facility

6 Physical View

A Physical View is a description of how a Functional View sub-set will be deployed in a given situation. Figure 15 shows a Functional View (in orange), with Functional Data Flows passing data between the various Functions and the Data Store. If Functions A, D and the Data Store are to be in one location (X), and Functions H, M and T are to be in another location (Y), one consequence is that Functional Data Flows FDFi and FDFj will be required to pass data from Sub-System X to Sub-System Y, and will require a suitable communications link with which to do so. Thus:

- The specifications for Sub-Systems X and Y can each be formed from the descriptions of the Functions of which they are composed;
- The communications requirements can be found by analysing the combined Functional Data Flows (FDFi + FDFj) between Sub-Systems X and Y.

Sometimes it can be useful to divide a Sub-System into two or more Modules, either for clarity, or to provide options (see Figure 16). In this situation it should be noted that there may be additional Physical Data Flows for each Module: e.g. Figure 17 shows two Modules with Physical Data Flows 1 and 2 respectively, which make up Physical Data Flow A for the Sub-system.

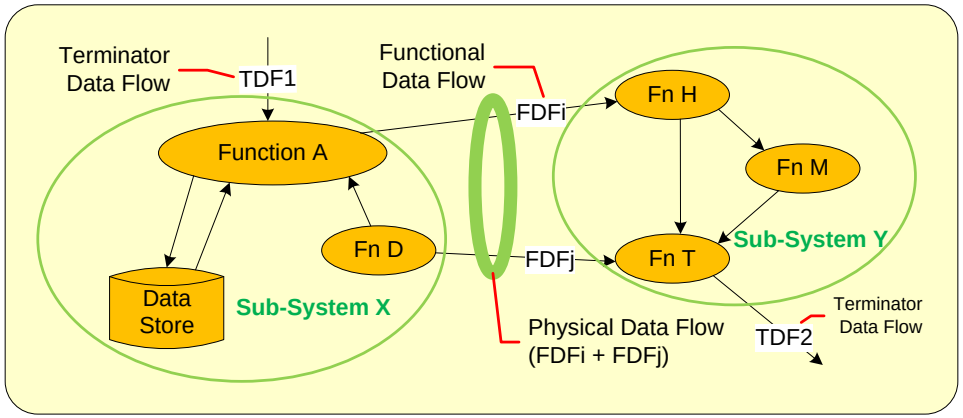

Figure 15 – Creating a Physical View from a Functional View

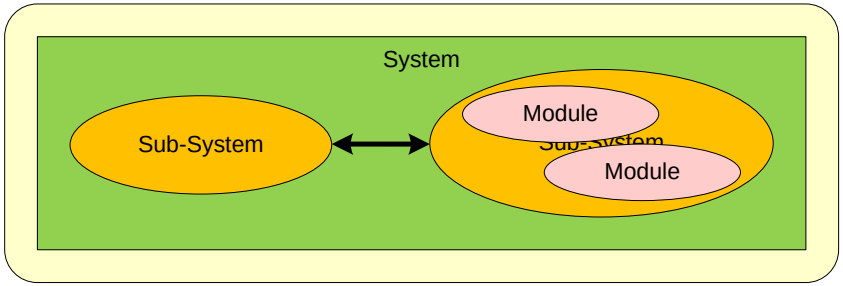

Figure 16 – Sub-systems and Modules

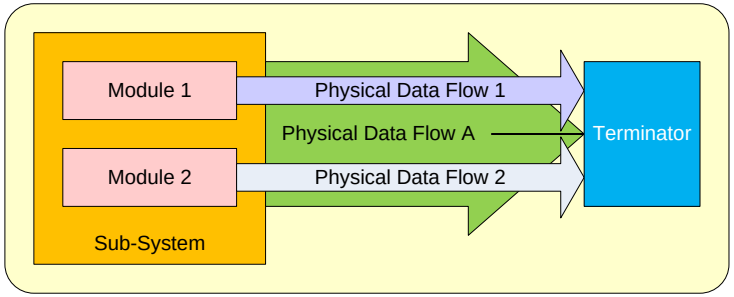

Figure 17 – Constituent Physical Data Flows

Co-financed by the European Union
Connecting Europe Facility

7 Communications View

The Communications View requires an analysis of the Physical Data Flows to identify the characteristics of the physical links that will carry the data, e.g.

- The types of data to be transferred, e.g. numeric, voice, video
  - The data transfer capacity required
- Any security requirements, e.g.
  - None – public information
  - Low – No unauthorised changing of data
  - High – No unauthorised reading of data

There will normally be suitable Standards available to support this analysis.

8 Organisational View

There are a number of issues that will need to be considered which include:

- Who owns, operates, regulates and maintains:
  - Each component
  - Each communication link
- Who is (legally) responsible for its correct operation
- What is the financial set up:
  - Who is paying for ITS implementation
    - Is there a revenue stream, and if so
    - Who collects it, and Who gets it?
- What happens to the data in the system
  - Who collects it? Who processes it? Who owns it?
  - To whom is it made available?

Analyse involvement of organisations with components:

- For each component and communication link identify its:
  - Owner, Operator, Regulator, Maintainer
- Characterise the links between organisations
  - Number of organisations involved
  - Do relationships already exist
  - Potential for difficult relationships
- Take action to resolve issues, e.g.
  - Produce contractual relationships, Revise components

9 Safety Issues

- Functional System Safety
  - Malfunctions due to faults in the design, operation, or failures of the system that lead to dangerous and/or unanticipated modes
- Human-Machine Interaction
  - Relating to faults generated by the visual, auditory and tactile interfaces within the vehicle, at the roadside, in a control room
- Traffic Safety
  - Relating to the safe operation of the traffic system as a whole, e.g. design of junctions

10 Security Issues

- Availability – data must be available when required, and not corrupted by external agents
- Personal data – GDPR: only to those who have a legitimate need
- Potential threat agents include
  - Dishonest drivers, Hackers, Criminals and terrorists, Dishonest organisations, “rogue states”

Co-financed by the European Union
Connecting Europe Facility