
**Transport information and control
systems — Reference model architecture(s)
for the TICS sector —**

**Part 3:
Example elaboration**

*Systèmes de commande et d'information des transports — Architecture(s)
du modèle de référence du secteur TICS —*

Partie 3: Élaboration d'exemple



Reference number
ISO/TR 14813-3:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO 2000

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Foreword.....	ix
Introduction	xi
1 Scope	1
2 Normative References.....	2
3 Terms and Definitions	2
4 Symbols and Abbreviated Terms.....	4
4.1 Use Case diagram.....	4
4.2 Package Diagram.....	5
4.3 Class diagram	5
4.4 Association	6
4.5 Sequence (interaction) diagram	6
5 Elaboration Method	7
6 Elaboration of the Classes.....	7
6.1 Control Classes	9
6.1.1 Roadway Classes	9
6.1.2 Transport Classes	10
6.1.3 Vehicles Classes.....	12
6.1.4 Event Classes	12
6.1.5 Payment Classes	13
6.2 Interface Classes	14
6.2.1 Operating Interface.....	14
6.2.2 Travel Terminal	14
6.2.3 Vehicle Interface	15
6.2.4 Roadside Peripheral	15
6.3 Information Classes	16
6.3.1 Information Classes for Traffic Management and Traveller Information	16
7 Elaboration of the Sequence Diagrams.....	27
7.1 Traveller Information	27
7.1.1 Pre-journey Information	27
7.1.2 Route Guidance and Navigation	28
7.1.3 Journey Schedule.....	30
7.1.4 Journey Payment.....	30
7.1.5 On-trip Traveller Information	31
7.2 Traffic Management.....	37
7.2.1 Traffic and Pollution Measurement and Control	37
7.2.2 Performance Evaluation.....	39
7.2.3 Performance Prediction	40
7.2.4 Traffic Control	40
7.2.5 Incident Management.....	41
7.2.6 Demand Management.....	42
7.2.7 Transport Planning Support.....	43
7.2.8 Infrastructure Maintenance and Management	45
7.2.9 Package Classes for Traffic Management.....	45
7.3 Vehicle	63
7.3.1 Vehicle Status	63
7.3.2 Vehicle Operation	63
7.4 Commercial Vehicle.....	64
7.4.1 Order and Shipment	64

7.4.2	Commercial Vehicle Tour Planning	65
7.4.3	Commercial Vehicle Administrative Processes	66
7.4.4	Commercial Vehicle Road Operation.....	67
7.5	Public Transport	73
7.5.1	Route and Schedule Planning	73
7.5.2	Fixed Route Public Transport.....	73
7.5.3	Demand Responsive Public Transport.....	74
7.6	Emergency.....	79
7.6.1	Emergency Notification and Personal Security.....	79
7.6.2	Emergency Resources Allocation.....	79
7.6.3	Emergency Vehicle Management.....	80
7.7	Electronic Payment.....	84
7.7.1	Payment Means.....	84
7.7.2	Fare Collection	84
7.7.3	Vehicle Charges	84
7.7.4	Payment Transaction.....	85
7.8	Safety	91
7.8.1	Safety Enhancement for Vulnerable Road Users	91
8	Elaboration of the Packages.....	93
8.1	Roadway	93
8.2	Transport	96
8.3	Vehicles	99
8.4	Events	101
8.5	Payment	103
8.6	Interfaces	105
8.7	Class-Sequence Diagram Matrix	107
8.7.1	Matrix for Key Control and Information Classes	107
8.7.2	Matrix for Interface Classes.....	112
9	Package Interfaces	115
9.1	Roadway Interfaces	117
9.2	Transport Interfaces	119
9.3	Vehicle Interfaces	120
9.4	Events Interfaces	121
9.5	Payment Interfaces	122
10	Dependencies between Packages	123
10.1	Roadway - Transport Collaboration.....	123
10.2	Roadway - Vehicles Collaboration	123
10.3	Roadway - Events Collaboration.....	123
10.4	Roadway - Payment Collaboration.....	126
10.5	Transport - Events Collaboration.....	126
10.6	Transport - Payment Collaboration.....	127
10.7	Vehicles - Events Collaboration	127
Annex A (informative)	Management and Information Centres Information Classes.....	128
A.1	Infrastructure Map	128
A.2	Moving Components	134
A.3	Effects on Operations.....	138
A.4	Traffic Usage	144
A.5	Traffic Management.....	152
A.6	User Information	155
A.7	Transport Systems	157
A.8	Financial Data.....	159
A.9	TMIC Management	160
Bibliography		164

List of figures

Figure 1 — A use case diagram consisting of two use cases and one actor	4
Figure 2 — Package diagram showing nested packages and a dependency relationship	5

Figure 3 — Class diagrams showing the use of a single name compartment and three compartments	5
Figure 4 — Common types of class association	6
Figure 5 — A sequence diagram consisting of three interactions	6
Figure 6 — Steps in the elaboration of the Core Reference Architecture	8
Figure 7 — Control class diagram of the Roadway package	10
Figure 8 — Control class diagram of the Transport package	12
Figure 9 — Control class diagram of the Events package	13
Figure 10 — Top level aggregation of Information classes derived from the 14827 naming tree	17
Figure 11 — Information classes in the aggregation of infrastructureMap	18
Figure 12 — Information classes in the aggregation of movingComponents	19
Figure 13 — Information classes in the aggregation of effectsOnOperations	20
Figure 14 — Information classes in the aggregation of trafficUsage	21
Figure 15 — Information classes in the aggregation of trafficManagement	22
Figure 16 — Information classes in the aggregation of userInformation	23
Figure 17 — Information classes in the aggregation of transportSystems	24
Figure 18 — Information classes in the aggregation of financialData	25
Figure 19 — Information classes in the aggregation of tmicManagement	26
Figure 20 — Sequence diagram for Pre-journey Information	32
Figure 21 — Sequence diagram for Route Guidance and Navigation	33
Figure 22 — Sequence diagram for Journey Schedule	34
Figure 23 — Sequence diagram for Journey Payment	35
Figure 24 — Sequence diagram for On-trip Traveller Information	36
Figure 25 — Sequence diagram for Traffic and Pollution Measurement and Control	46
Figure 26 — Packages, control and information classes for Traffic and Pollution and Measurement and Control ...	47
Figure 27 — Sequence diagram for Performance Evaluation	48
Figure 28 — Packages, control and information classes for Performance Evaluation	49
Figure 29 — Sequence diagram for Performance Prediction	50
Figure 30 — Packages, control and information classes for Performance Prediction	50
Figure 31 — Sequence diagram for Traffic Control	51
Figure 32 — Packages, control and information classes for Traffic Control	52

Figure 33 — Sequence diagram for Incident management.....	53
Figure 34 — Packages, control and information classes for Incident Management	54
Figure 35 — Sequence diagram for Demand Management.....	55
Figure 36 — Packages, control and information classes for Demand Management	56
Figure 37 — Sequence diagram for Transportation Planning Support	57
Figure 38 — Packages, control and information classes for Transportation Planning Support	58
Figure 39 — Sequence diagram for Infrastructure Maintenance Management	59
Figure 40 — Packages, control and information classes for Infrastructure Maintenance and Management.....	59
Figure 41 — Roadway package class operations and information class associations for Traffic Management.....	60
Figure 42 — Events package class operations and information class associations for Traffic Management.....	61
Figure 43 — Transport package class operations and information class associations for Traffic Management	62
Figure 44 — Sequence diagram for Vehicle Status	64
Figure 45 — Sequence diagram for Vehicle Operation.....	64
Figure 46 — Sequence diagram for Order and Shipment.....	69
Figure 47 — Sequence diagram for Commercial Vehicle Tour Planning.....	70
Figure 48 — Sequence diagram for Commercial Vehicle Administrative Processes.....	71
Figure 49 — Sequence diagram for Commercial Vehicle Road Operation.....	72
Figure 50 — Sequence diagram for Route and Schedule Planning.....	76
Figure 51 — Sequence diagram for Fixed Route Public Transport.....	77
Figure 52 — Sequence Diagram for Demand Responsive Public Transport.....	78
Figure 53 — Sequence diagram for Emergency Notification and Personal Security	81
Figure 54 — Sequence diagram for Emergency Resources Allocation	82
Figure 55 — Sequence diagram for Emergency Vehicle Management	83
Figure 56 — Sequence diagram for Payment Means	87
Figure 57 — Sequence diagram for Fare Collection	88
Figure 58 — Sequence diagram for Vehicle Charges	89
Figure 59 — Sequence diagram for Payment Transaction	90
Figure 60 — Sequence diagram for Safety Enhancement for Vulnerable Road Users	92
Figure 61 — Key classes of the Roadway package	93
Figure 62 — Key classes of the Transport package.....	96

Figure 63 — Key classes of the Vehicle package	100
Figure 64 — Key classes of the Events package	101
Figure 65 — Key classes of the Payment package.....	103
Figure 66 — Key classes of the interface packages	105
Figure 67 — Roadway Package Interfaces	116
Figure 68 — Transport Package Interfaces.....	118
Figure 69 — Vehicle Package Interfaces	120
Figure 70 — Events Package Interfaces	121
Figure 71 — Payment Package Interfaces.....	122
Figure 72 — Roadway - Transport Collaborations	124
Figure 73 — Roadway - Vehicles Collaborations	124
Figure 74 — Roadway - Events Collaborations.....	125
Figure 75 — Roadway - Payment Collaborations	126
Figure 76 — Transport - Events Collaborations	127
Figure 77 — Transport - Payment Collaborations	127
Figure 78 — Vehicles - Events Collaborations	127
Figure A.1 — The top level of the naming tree for the information classes defined in the data dictionary	128

List of tables

Table 1 —Control classes and operations of the Roadway package.....	9
Table 2 — Control classes and operations of the Transport package	11
Table 3 — Control classes and operations of the Vehicle package	12
Table 4 — Control classes and operations of the Events package.....	13
Table 5 — Control classes and operations of the Payment package.....	14
Table 6 — Operating Interface class and operations	14
Table 7 — Travel Terminal class and operations	15
Table 8 — Vehicle Interface class and operations	15
Table 9 — Roadside Peripheral class and operations	15
Table 10 — Matrix count of Roadway class Interface class collaborations	117
Table 11 — Matrix count of Transport class Interface class collaborations	119

Table 12 — Matrix count of Vehicles class Interface class collaborations 120

Table 13 — Matrix count of Events class Interface class collaborations.....121

Table 14 — Matrix count of Payment class Interface class collaborations 122

Table 15 — Matrix count of Roadway class Transport class collaborations 123

Table 16 — Matrix count of Roadway class Events class collaborations.....125

Table 17 — Matrix count of Roadway class Payment class collaborations 126

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Technical Reports are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Attention is drawn to the possibility that some of the elements of this part of ISO TR 14813 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TR 14813-3, which is a Technical Report of type 2, was prepared by Technical Committee ISO/TC 204, *Transport information and control systems*.

This document is being issued in the Technical Report (type 2) series of publications (according to subclause G.3.2.2 of Part 1 of the ISO/IEC Directives, 1995) as a "prospective standard for provisional application" in the field of transport information and control systems because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not to be regarded as an "International Standard". It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO Central Secretariat.

A review of this Technical Report (type 2) will be carried out not later than three years after its publication with the options of: extension for another three years; conversion into an International Standard; or withdrawal.

ISO TR 14813 consists of the following parts, under the general title *Transport information and control systems* — Reference model architecture(s) for the TICS sector:

- *Part 1: TICS Fundamental Services*: This document presents the definition of 32 TICS fundamental services that are the informational products or services or applications areas provided to a TICS user.
- *Part 2: Core TICS Reference Architecture*: This document describes an abstract object-oriented system architecture based on the TICS Fundamental Services.

- *Part 3: Example Elaboration:* This document refines the Core TICS Reference Architecture (Part 2) with some emphasis on traffic management.
- *Part 4: Reference Model Tutorial:* This document describes the basic terms, graphical representations and modelling views exploited in the object-oriented definition of the architecture development of Parts 2 and 3.
- *Part 5: Requirements for Architecture Description in TICS Standards:* Requirements for Architecture Description in TICS Standards: This document describes the terminology and form to be used when documenting or referencing aspects of architecture description in TICS standards.
- *Part 6: Data Presentation in ASN.1:* This document establishes the use of ASN.1 as the normal syntax notation to be used in standards for the TICS sector and a common message form for such ASN.1 based data elements.

Annex A of this part of ISO/TR 14813 is for information only.

Introduction

TC204/WG1 is a working group whose prime objectives are to provide services to ISO TC204 and its working groups. A specific mission of WG1 is to:

“Provide ISO TC204, its working Groups, related bodies and those involved in the TICS sector, with a reference model of Conceptual Reference Architecture(s) that show the structure and inter-relationships of the sector ...”

It is expected that there may well be more than one single TICS Architecture approach to be considered and documented and that existing architecture approaches will have previously-produced documentation developed according to disparate standards and conventions.

It is also implicit in the work being undertaken by WG1, that working group members will require a clear, well-structured understanding of the work of the following participant groups:

- Other TC 204 Working Groups
- CEN TC 278 Working Groups
- Japanese initiatives
- European Road Transport and Traffic Telematics programs
- US Intelligent Transportation Systems program
- Australian initiatives
- Canadian Initiatives

Full documentation of all possible architectural approaches is obviously not feasible given the high level of resources required to carry this out. Indeed full documentation and description of all possible approaches is undesirable as an item for Standardisation.

A defined and consistent approach is however required to facilitate the specification of architecture requirements to enable a clear view to be developed and presented of the work of each participant group. This document is one of a set of WG1 documents intended to respond to stated WG1 objectives regarding the production of a TICS Reference Architecture.

In order to document an architecture, graphical and textual components of a model are required. WG1 has adopted a methodology based on the Unified Modelling Language (UML) for documenting the TICS Reference Architecture. A tutorial on the UML is provided in ISO/TR 14813 Part 4. UML is a visual modelling language for building object-oriented and component-based systems. A commercially available Computer Aided Software Engineering (CASE) tool has been used by WG1 to document the Architecture. While the tool is a commercial product, UML is open and non-proprietary.

Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 3: Example elaboration

1 Scope

The architecture of an information and control system merges hardware and software considerations into a coordinated and integrated system view. The system architecture is a high level abstraction, or model, of the system. A system architecture should embrace both today's applications and the applications that are expected in the future. Architecture begins with the definition of the conceptual services (e.g. Part 1 - TICS Fundamental Services). There are several identifiable stages of system architecture development.

- a) Reference architecture
- b) Logical architecture
- c) Physical architecture

A reference architecture is the first of all architectures. It is a concise generic framework which guides the development of more concrete system architectures. It is large enough that distinct concepts are not merged out of necessity and small enough that it does not become unwieldy.

A most significant example of a reference architecture in information systems is the Reference Model of Open Systems Interconnection (often called the seven layer model) developed by ISO in the 1970's. This model has underpinned the development of all modern computer networks, allowing services such as global networking, of which the prime example is the Internet, to become a reality.

A reference architecture is generic and non-prescriptive and captures the concepts of the system. A logical architecture elaborates the conceptual behaviour, and in so doing it provides more detail about the modularity. A physical architecture is reached when the actual distribution of the system modules is defined, thus leading to important implications for communications.

There is no firm demarcation between a reference architecture and a logical architecture. Thus the essence of behaviour and modularity is present in a reference architecture. The TICS Reference Architecture developed by WG1 shows important inter-relationships that arise in the provision of the services of the sector. However the TICS Reference Architecture is more abstract than, for example, the logical architecture of the US National Architecture.

It is envisioned that the TICS Reference Architecture will be used by the TC204 Working Groups to develop their own logical and physical architectures in a cohesive manner.

Some TICS Fundamental Services are already well developed by the industry, while others are less mature. Therefore the TICS Reference Architecture does not have a uniform granularity across all services. This characteristic is a direct result of the fore mentioned requirement that architecture embrace the applications that are intended in the future. This suggests one of the ways in which the architecture will undergo change in the future.

Architectures may present only static characteristics or both static and dynamic characteristics. Dynamic characteristics may be seen as belonging solely to the design/implementation stages of system development. However by including dynamic characteristics at the reference architecture stage one can gain important insights into the static architecture. Thus two orthogonal views of architecture are presented:

- a) static relationship view (class diagram)
- b) dynamic interactive view (sequence diagram)

Part 2 develops a Core TICS Reference Architecture. The static scope is determined by deriving the system boundary and the use cases from an analysis of the TICS Fundamental Services (Part 1).

The Core Reference Architecture is a reference for the development of national architectures.

This Part elaborates the core by refinement of the two orthogonal views. The elaboration calls upon domain expertise which would be provided by other TC204 Working Groups in the development of ISO standards, or by national groups developing national architectures and standards.

The Core Reference Architecture is elaborated in Clauses 6 to 8. Clause 5 introduces the elaboration method employed. Clause 6 elaborates the classes. Clause 7 elaborates the sequence diagrams. Clause 8 describes the elaborated packages. Clauses 9 and 10 identify some of the main dependencies between the packages.

Readers should refer to Part 4 (Tutorial) for an introduction to the modelling views used in this Part and the overall methodology.

2 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this part of ISO/TR 14813. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/TR 14813 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/TR 14813-1:1999, *Transport information and control systems – Reference model architecture(s) for the TICS sector – Part 1: TICS fundamental services*.

ISO/TR 14813-2:2000, *Transport information and control systems – Reference model architecture(s) for the TICS sector – Part 2: Core TICS reference architecture*.

ISO/TR 14813-4:2000, *Transport information and control systems – Reference model architecture(s) for the TICS sector – Part 4: Reference model tutorial*.

3 Terms and Definitions

For the purposes of this part of ISO/TR 14813, the following semantic definitions apply.

3.1

The **Unified Modeling Language** (UML) is the industry-standard language for specifying, visualising, constructing, and documenting the artefacts of software systems. It simplifies the complex process of software design, making a “blueprint” for construction.¹

3.2

A **use case** is a coherent unit of functionality provided by a system or class as manifested by sequences of messages exchanged among the system and one or more outside interactors (called actors) together with actions performed by the system.

¹ <http://www.rational.com/uml/>

3.3

An **actor** is a role of an object or objects outside of a system that interacts directly with it as part of a coherent work unit (a use case). An Actor element characterises the role played by an outside object; one physical object may play several roles and therefore be modelled by several actors.

3.4

Use case diagrams show elements from the use case model. The use case model represents functionality of a system or a class as manifested to external interactors with the system.

There are several standard relationships among use cases or between actors and use cases.

Communicates - The participation of an actor in a use case. This is the only relationship between actors and use cases.

Uses - A uses relationship from use case A to use case B indicates that an instance of the use case A will also include the behavior as specified by B.

To reinforce the externality of actors a system boundary separates the actor symbols from the use case symbols.

3.5

A **package** is a grouping of model elements. Packages themselves may be nested within other packages. A package may contain both subordinate packages and ordinary model elements. The entire system description can be thought of as a single high-level *system* package with everything else in it. All kinds of UML model elements and diagrams can be organised into packages.

Packages own model elements and model fragments and are the basis for configuration control. Each model element can be directly owned by a single package, so the package hierarchy is a strict tree.

Packages can reference other packages so the usage network is a graph. Relationships drawn between package symbols denote relationships between at least some of the elements in the packages. A dependency relationship implies one or more dependencies among the model elements, in which a change in the targeted element may require a change in the source element.

3.6

A **class** is the descriptor for a set of objects with similar structure, behavior, and relationships. UML provides notation for declaring classes and specifying their properties, as well as using classes in various ways. Classes are declared in class diagrams and used in most other diagrams. UML provides a graphical notation for declaring and using classes, as well as a textual notation for referencing classes within the descriptions of other model elements.

3.7

A **class diagram** is a graph of Classifier elements connected by their various static relationships. (Note that a "class" diagram may also contain interfaces, packages, relationships, and even instances, such as objects and links. Perhaps a better name would be "static structural diagram" but "class diagram" is shorter and well established.)

3.8

An **object** represents a particular instance of a class. It has identity and attribute values. The same notation also represents a role within a collaboration because roles have instance-like characteristics.

3.9

An **operation** is a service that an instance of the class may be requested to perform. It has a name and a list of arguments.

3.10

A binary **association** is an association among exactly two classes (including the possibility of a reflexive association from a class to itself).

3.11

Generalization is the taxonomic relationship between a more general element and a more specific element that is fully consistent with the first element and that adds additional information. It is used for classes, packages, use cases, and other elements.

3.12

A **sequence diagram** represents an **Interaction**, which is a set of messages exchanged among objects within a collaboration to effect a desired operation or result. A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the objects participating in the interaction by their "lifelines" and the messages that they exchange arranged in time sequence. It does not show the associations among the objects.

3.13

The **system boundary** depicted in a sequence diagram maps to the same entity in a use case diagram. Thus any interaction emanating or terminating in the system boundary involves an actor.

In the methodology classes are invented for one of three purposes: information, control and interface.

An information class defines objects, which will store data relevant to the operation of the system and the actors and maintain that data with database like services.

A control class defines objects whose primary purpose is to implement the functions of the system.

An interface class defines objects that perform the data presentation and application interfaces for the actors.

3.14

The **architecture boundary** divides the interface classes from those classes, which form the actual architecture, namely the control classes and the information classes.

In the sequence diagrams developed in later clauses there is often an implicit interaction across the system boundary involving an actor. This may be implied whenever a message is initiated or terminated at an interface class.

4 Symbols and Abbreviated Terms

4.1 Use Case diagram

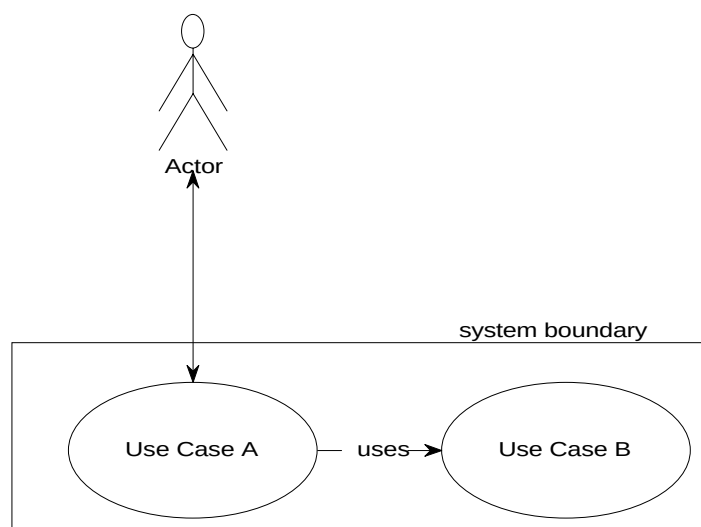


Figure 1 — A use case diagram consisting of two use cases and one actor

4.2 Package Diagram

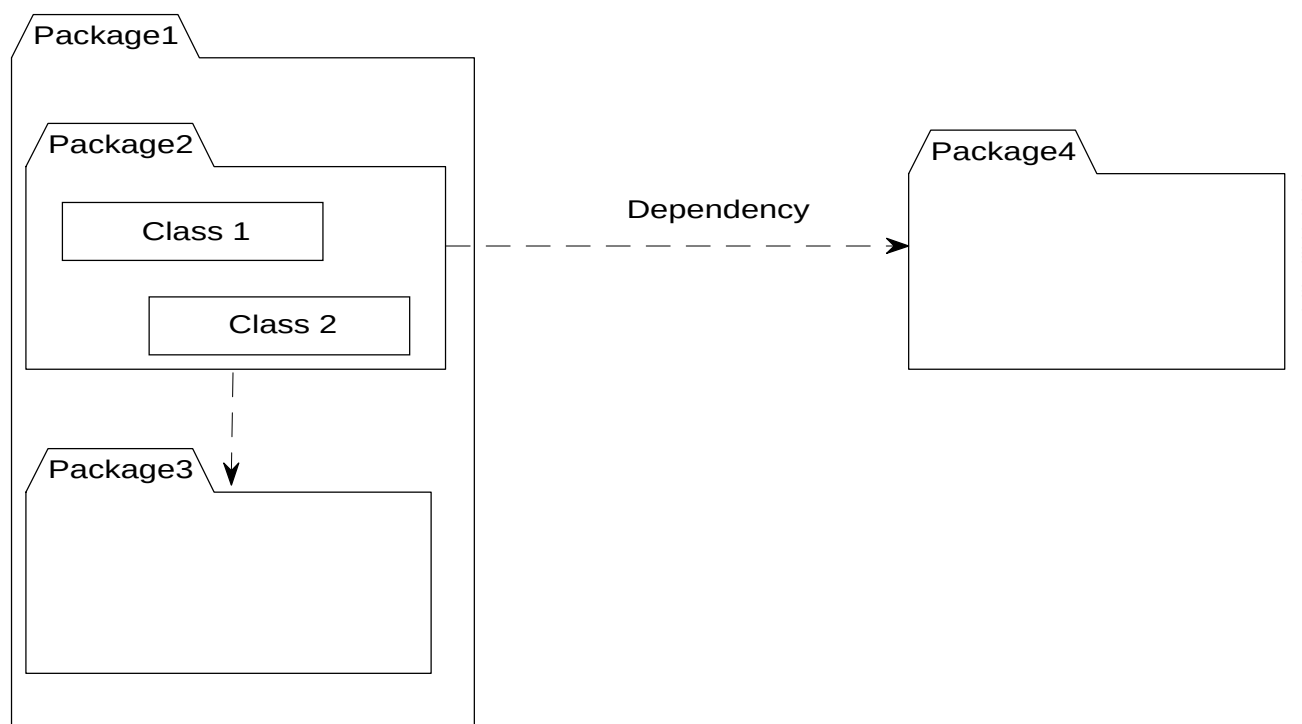


Figure 2 — Package diagram showing nested packages and a dependency relationship

4.3 Class diagram

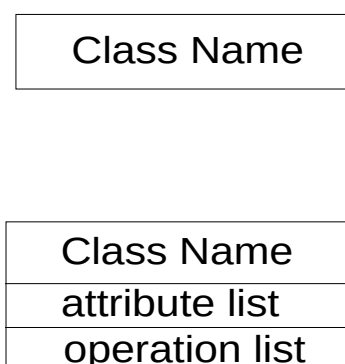
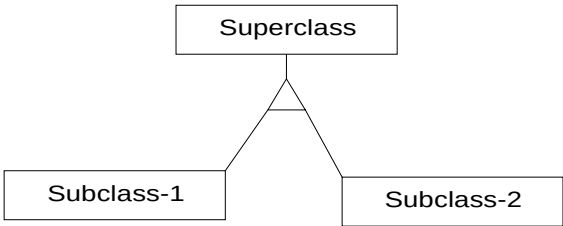


Figure 3 — Class diagrams showing the use of a single name compartment and three compartments

4.4 Association



Generalisation (inheritance)



Aggregation

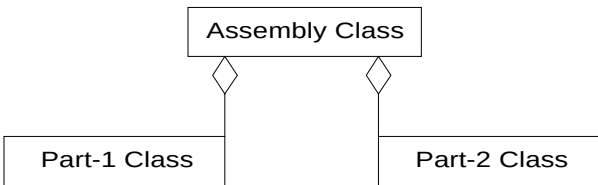


Figure 4 — Common types of class association

4.5 Sequence (interaction) diagram

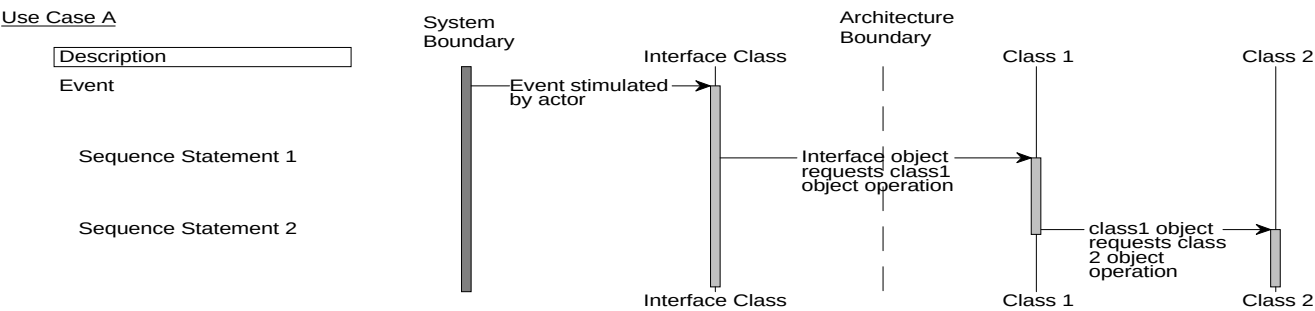


Figure 5 — A sequence diagram consisting of three interactions

5 Elaboration Method

The core Reference Architecture has been defined in ISO/TR 14813 Part 2, using the UML metatypes actor, use case, package and class, as well as sequence diagrams. All the instances of the metatypes in the core architecture are highly abstract. They require a good deal of elaboration before they can be of use in the standards process. That elaboration is now taken a step further.

The elaboration method is presented in Figure 6. There are four stages. While these are ordered in the figure, the whole process is iterative because of the necessity to formulate many new classes and their operations, guided by the principle of reuse.

The first stage is a transformation and refinement of the abstract classes defined in ISO/TR 14813 Part 2. The process is based on an analysis of the class operations identified in the core reference architecture. This analysis leads to some fairly obvious partitions in the light of domain knowledge already documented in the associated use case and the sequence diagram. Each partition is used to define a new control class. Other control classes are identified in the third stage. This first stage is conducted in clauses 6.1 and 6.2.

The second stage would actually be spread throughout the third stage. It involves the formulation of a wide range of information classes that serve to define the parameters of the class operations involved in the object interactions of the new sequence diagrams. In this report the activity is only partially completed. It is focused on traffic management where a large number of information classes are identified from a single reference source.

The third stage is the key to the elaboration and is presented in clause 7. This stage constitutes a major refinement of the sequence diagrams of the core architecture. A sequence diagram is developed for each of the use case identified in the diagrams of the core architecture. Elaborating the logic of each use case results in a significant set of operations being defined for each class. In some cases the parameters of these operations are identified. The process results in the definition of more classes not previously identified in the first two stages.

The fourth stage is a redefinition of the packages based on the expanded set of classes. This is presented in clause 8 for the key classes. Some detailed packaging is defined in 7.2 for classes relevant to Traffic Management.

6 Elaboration of the Classes

The class refinement replaces each abstract class from ISO/TR 14813 Part 2 with a number of classes. These classes fall into the categories:

- c) control classes
- d) information classes
- e) interface classes

Often there is not a clear-cut distinction between control and information classes, but there is usually a dominant role which will be noted. Each class is associated with one of the packages defined in ISO/TR 14813 Part 2.

As in ISO/TR 14813 Part 2 the classes will have operations defined in accordance with the logic of the use case. Here the elaboration of the architecture also includes the definition of the parameters of some of the operations. The abstract level of the parameters means that it is appropriate to define them as information classes associated with the relevant control class. (These information classes are considered to be attributes of the associated control class. Association is the normal means by which complex attributes are implemented in object-oriented software.)

The elaboration will be carried out in several stages. The next clause defines the most important control classes.

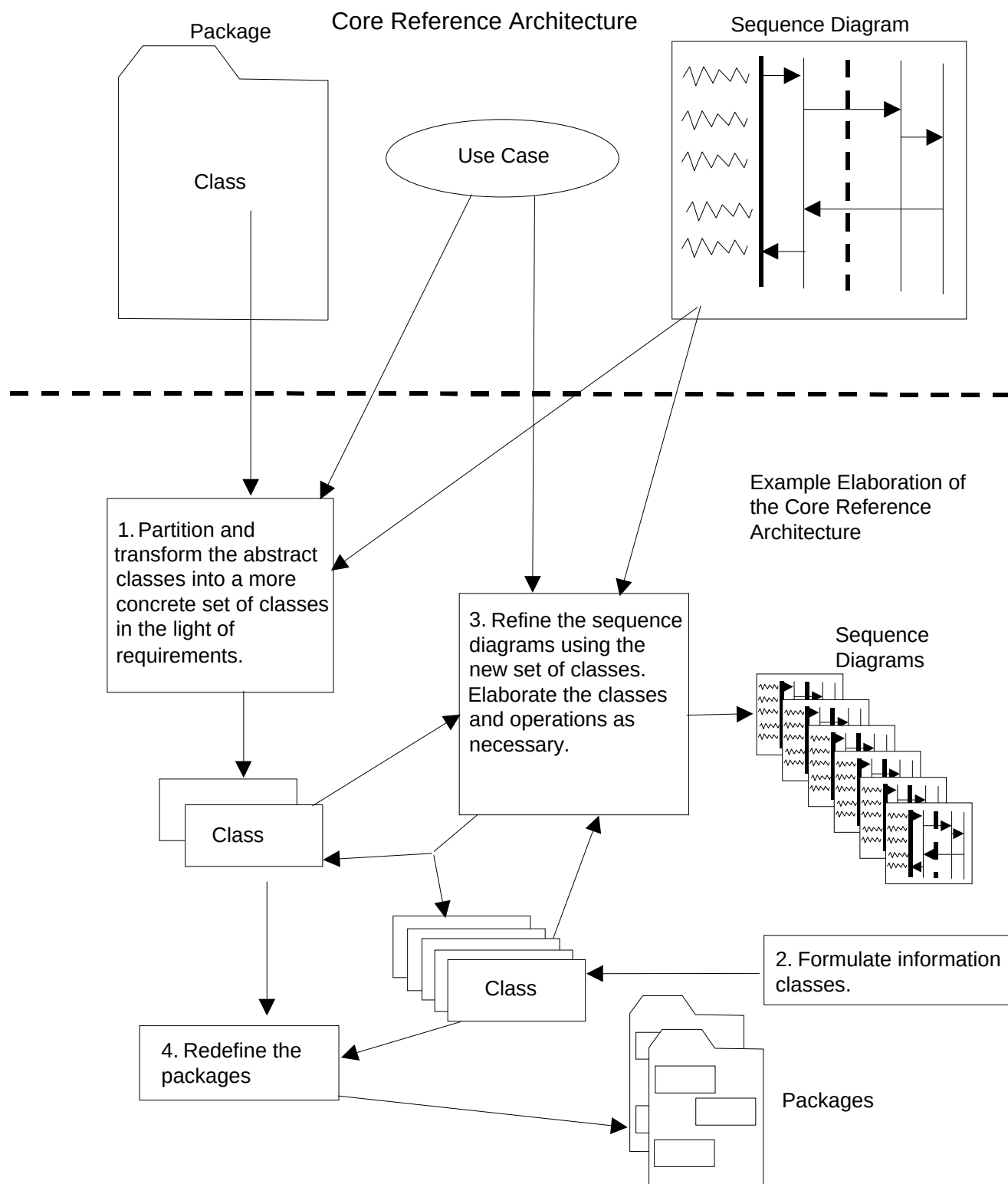


Figure 6 — Steps in the elaboration of the Core Reference Architecture

6.1 Control Classes

This clause infers a number of control classes by systematically grouping the operations of the abstract classes of ISO/TR 14813 Part 2 using general domain knowledge. This systematic grouping is a connection to reality. Each group corresponds to a new control class.

The operations of these new classes will be defined subsequently at a finer granularity, but they will have to deliver the functionality implied by the group of operations derived from the abstract class. Usually the new control classes will have association relationships. The rationale for the identification of each class is presented below.

The reader may wish to refer to ISO/TR 14813 Part 2 for definitions of the abstract operations occurring in the tables below.

6.1.1 Roadway Classes

The operations of the abstract Roadway Class in ISO/TR 14813 Part 2 are classified and grouped in Table 1. This permits the Roadway Class to be replaced in a refinement process by three classes.

Table 1 —Control classes and operations of the Roadway package

Resource	Local Control Group	Roadway Group
accept user credit detect and identify vehicle resource use statistics vehicle safety and logging	measure & local control priority request	access external data access roadway conditions compute control actions data fusion evaluate manage demand plan construction and maintenance predict priority request support planning update geographic data

6.1.1.1 Resource

This class is invented for the purpose of monitoring the use of parts of the roadway network, particularly where commercial vehicle taxes, tolling, and certain safety aspects are involved. A resource object would model a part of the network for which a financial or logging transaction may occur.

6.1.1.2 Local Control Group

This class is invented for the purpose of monitoring and controlling a small or confined part of the roadway such as an intersection or a section of motorway or several such elements which have a geometric relationship. An object of this class might contain one or more Resource objects as shown by the aggregation association in Figure 7.

6.1.1.3 Roadway Group

This class is invented for the purpose of exercising strategic control over the roadway network. An object of this class might contain some geographically extensive Resource objects and a number of Local Control Group objects, the aggregation being defined in strategic design steps. These associations are shown in Figure 7.

There will be additional operations defined for these control classes which allow their cooperation in line with defined associations. Many information classes will be defined and associated with the Roadway Group class to serve the control operations of the classes in Figure 7.

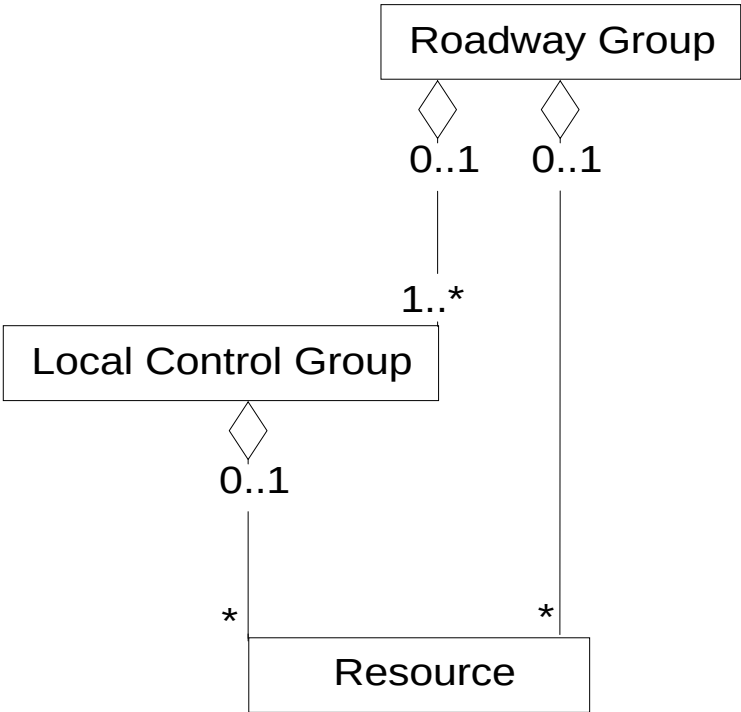


Figure 7 — Control class diagram of the Roadway package

6.1.2 Transport Classes

The operations of the abstract Transport Class in ISO/TR 14813 Part 2 are classified and grouped in Table 2. This permits the Transport Class to be replaced in a refinement process by six classes.

6.1.2.1 Route

This class is invented to model the geographic characteristics for a route. It is specialised into two important sub-classes as shown in Figure 8.

6.1.2.2 O-D Route

This sub-class of Route is invented for the purpose of describing routes for important representative trips. Operations of this class maintain data about services, prevailing conditions and performance, thus it has a role as an information class.

6.1.2.3 Public Transport Route

This sub-class of Route is invented for the purpose of defining and operating the routes of public transport services. The operations also maintain information about stops and interchanges on the roadway network of a public transport service.

Table 2 — Control classes and operations of the Transport package

O-D Route	Public Transport Route	Schedule	Journey	Commercial Vehicle	Forwarder
evaluate post predictions	access running data create & maintain routes & schedules process operational data response request service change request	ad hoc inquiry reservation service	establish journey conditions and options plan journey schedule journey	access incidents monitor vehicle & cargo recognise vehicle	select forwarder

6.1.2.4 Schedule

This class is invented to model all types of public transport schedule. The operations relate to time-tabled and demand responsive public transport services. They maintain time-tables, operational data, and demand responsive vehicle itineraries and dispatching. Every Public Transport Route object has a Schedule object associated with it.

6.1.2.5 Journey

This class is invented to serve route and schedule planning for a transportable item. The operations develop the overall plan for Travellers (i.e. Passengers and all types of Drivers).

It is also an information class for the total movement associated with a journey request. The request might be for a goods item to be taken from Consignor to Consignee, or for a Traveller planning a multi-leg Journey.

Every Journey object has one or more O-D Route objects associated with it, corresponding to the number of legs or trips which make up the Journey object.

6.1.2.6 Commercial Vehicle

This sub-class of Vehicle (see 6.1.3) is invented for the purpose of controlling commercial vehicles on route and interacting with classes in the Roadway package.

A Commercial Vehicle object which has been assigned to a tour will be associated with a Journey object.

6.1.2.7 Forwarder

This class is invented to model some functions of freight movement which are inside TICS.

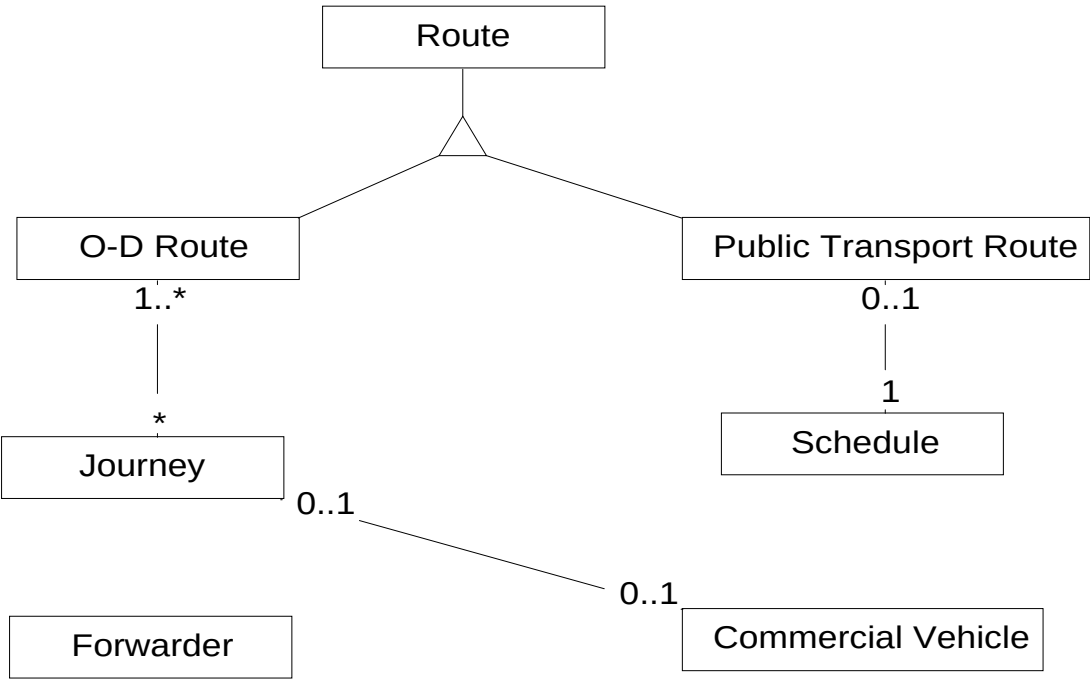


Figure 8 — Control class diagram of the Transport package

6.1.3 Vehicles Classes

The operations of the abstract Vehicle Class in ISO/TR 14813 Part 2 are grouped and retained under a single class in Table 3.

6.1.3.1 Vehicle

This class is invented to model the control of a vehicle for safety and basic operational purposes. There are special classes of vehicle which inherit these characteristics.

Table 3 — Control classes and operations of the Vehicle package

Vehicle
control vehicle
data fusion
monitor context
support planning

6.1.4 Event Classes

The operations of the abstract Events Class in ISO/TR 14813 Part 2 are classified and grouped in Table 4. This permits the Events Class to be replaced in a refinement process by three classes. (See Figure 9)

Table 4 — Control classes and operations of the Events package

Incident	Emergency	Violation
access incidents predicted incidents support planning update incidents & emergencies evaluate	develop strategy process emergency request emergency response update incidents & emergencies	process violations

6.1.4.1 Incident

This class is invented to control the process of predicting and handling incidents.

6.1.4.2 Emergency

This class is invented to control the process of detecting and handling an emergency. Thus an emergency object may be associated with an incident object, and vice versa.

6.1.4.3 Violation

This class is invented to control the processing of all violations of regulations. In the case of a serious violation a Violation object may be associated with an Emergency object.

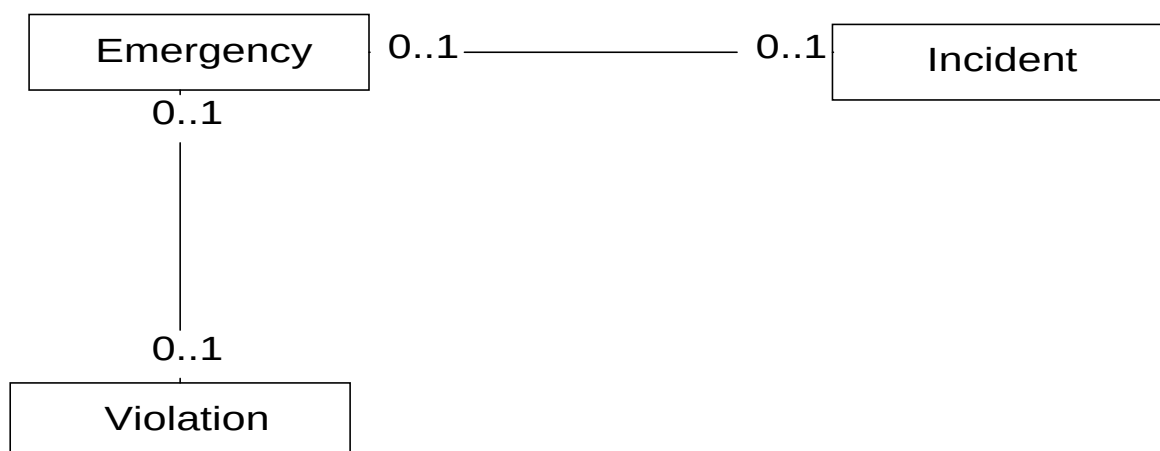


Figure 9 — Control class diagram of the Events package

6.1.5 Payment Classes

The operations of the abstract Payment Class in ISO/TR 14813 Part 2 are classified and grouped in Table 5. This permits the Payment Class to be replaced in a refinement process by three classes.

Table 5 — Control classes and operations of the Payment package

Billing Record	Tariff	Yellow Pages
compute charges create billing record make payment	declare tariff response request support planning	maintain yellow pages

6.1.5.1 Billing Record

This class is invented to support the initialisation and completion of a Resource use, and any payment. A Billing Record object may be associated with a Vehicle object in the case of tolling transactions.

6.1.5.2 Tariff

This class is invented to support the calculation of charges for Resource usage and variations which may be required. A Resource object is associated with a Tariff object.

6.1.5.3 Yellow Pages

This class is invented to maintain the wider information about a general service not held in the Tariff class.

6.2 Interface Classes

The operations of the interface classes are grouped at the same level of abstraction as in ISO/TR 14813 Part 2. They are shown in Tables 6 to 9. The main significance of the interface classes at this stage of architecture elaboration is in the definition of the system boundary and the architecture boundary.

6.2.1 Operating Interface

This interface class generalises all the different types of operating interface. The operations perform data display, support Operator interaction and interaction with a range of external operational systems.

Table 6 — Operating Interface class and operations

Operating Interface
action request conformance check request credit issuer payment reservation request

6.2.2 Travel Terminal

This interface class generalises all the street kiosk interfaces as well as internet and other interfaces. The operations perform data display and support Traveller interaction.

Table 7 — Travel Terminal class and operations

Travel Terminal
accept cash/credit
advertise
debit
payment request
pre-payment
publish traveller information
sense security violation or alert

6.2.3 Vehicle Interface

This sub-class of Travel Terminal is part of the Vehicle package and generalises in-vehicle interfaces. The operations perform data capture and display, other user communications, and support Driver and Passenger interaction.

Table 8 — Vehicle Interface class and operations

Vehicle Interface
access payment means
access vehicle & passenger data
command driver
deploy vehicle and instruct driver
interrogate fleet
notify driver
track vehicle & status

6.2.4 Roadside Peripheral

This interface class is part of the Roadway package. The objects interface signs, some types of sensors and communications beacons. The operations support information flow between Drivers and the Roadway control classes.

Table 9 — Roadside Peripheral class and operations

Roadside Peripheral
notify vulnerable
update control

6.3 Information Classes

With each step in the elaboration of the architecture there is a more rapid growth in the number of information classes than the other types. This is the nature of the large scale information and control system. Control is based on a large amount of real time data, which is reduced to important parameters for key operations. To serve historical, real time and transactional purposes the information classes and their associations constitute schemas for notional object databases.

As noted at the beginning of clause 5, the level of abstraction of an architecture means that the parameters of the class operations are themselves defined as information classes. These information classes are associated with other classes that generate and receive the messages of interest in the architecture.

The definition of the information classes calls for more detailed TICS domain knowledge. In TC204 this domain knowledge will be supplied by the various working groups.

Objects of an information class interact in the standard manner, that is through the operations defined in their interface. These operations maintain data stored in the attributes of the object.

In architecture every information class has the same implicit set of generic operations:

- create(...)
- update(...)
- retrieve(...)
- delete(...)

The difference between each class is the set of attributes and parameters that the operations manipulate. When they serve as parameters in the operations of control classes of the architecture, information objects are always used in their entirety.

6.3.1 Information Classes for Traffic Management and Traveller Information

The draft standard ISO 14827 **Message Format and Information Contexts for Traffic Management and Information Centres (TMIC)**²⁾ has recently been developed by TC204/WG9. The data definitions in draft ISO 14827 are not data dictionary definitions, rather they are informal definitions of complex data elements. Most of these data elements map to parameters of the operations in the sequence diagrams for the use cases in Traffic Management and Traveller Information.

In elaborating the architecture a uniform approach was adopted in the use of the ISO 14827 data elements. Each data element definition maps to an information class definition, and the text in ISO14827 is used unchanged. Since that document does not use an object-oriented model there is some inconsistency in terminology. For example, ISO 14827 often refers to a “list of objects”; this should be read as the set of objects belonging to a class. A dictionary of the information classes so derived is given in Annex A.

There are 360 data elements in draft ISO 14827, identified in a naming tree. The root structure of this naming tree has been defined as an aggregate of nine information classes in Figure 10. An information class is associated with each top level data element of ISO 14827, using the name in the ISO 14827 naming tree as part of the class name.

The digits in the leading part of the class name are those used in ISO 14827 to indicate the position in the naming tree. They are included because they are a convenient abbreviation for the information class names where they occur as parameters in the sequence diagrams.

2) The classes derived here are based on draft N173H, 20-01-98.

The remaining information classes were defined in nine hierarchies shown in Figures 11 to 19. The methodology was to first view the leaf nodes (data elements) of the naming tree as candidate information classes. Consolidation of these classes was performed as aggregations to the next level, according to the graph of the naming tree. In order to avoid clutter the leaf information classes are not shown in the Figures 11 to 19. However the parameters of the operations in the sequence diagrams of clause 7 of this report will usually be the aggregate classes.

There are a small number of exceptions where an aggregation association was not appropriate.

In Figure 11 “[1---] infrastructureMap” class is specialised into three subclasses. The ISO 14827 definitions prompting these associations are:

- [1---] infrastructureMap: These are maps of the static physical network.
- [11--] generalMap: This base map describes all relevant geographic features of the region.
- [12--] transportMap: These are facilities operated by the relevant service provider.
- [13--] roadMap: No definition.

In Figure 12 “[21--] vehicles” class is specialised into two sub-classes and it has two other classes related in standard associations. The ISO 14827 definitions prompting these associations are:

- [211-] individualVehicleID: No definition – clearly a vehicle attribute.
- [212] governmentID: This information class shall consist of the vehicles (objects) that an agency wishes to have identified in the manner indicated.
- [213-] policeListing: This information class shall consist of the vehicles (objects) that an agency wishes to have identified in the manner indicated.
- [214-] vehicleCondition: No definition – clearly a vehicle attribute.

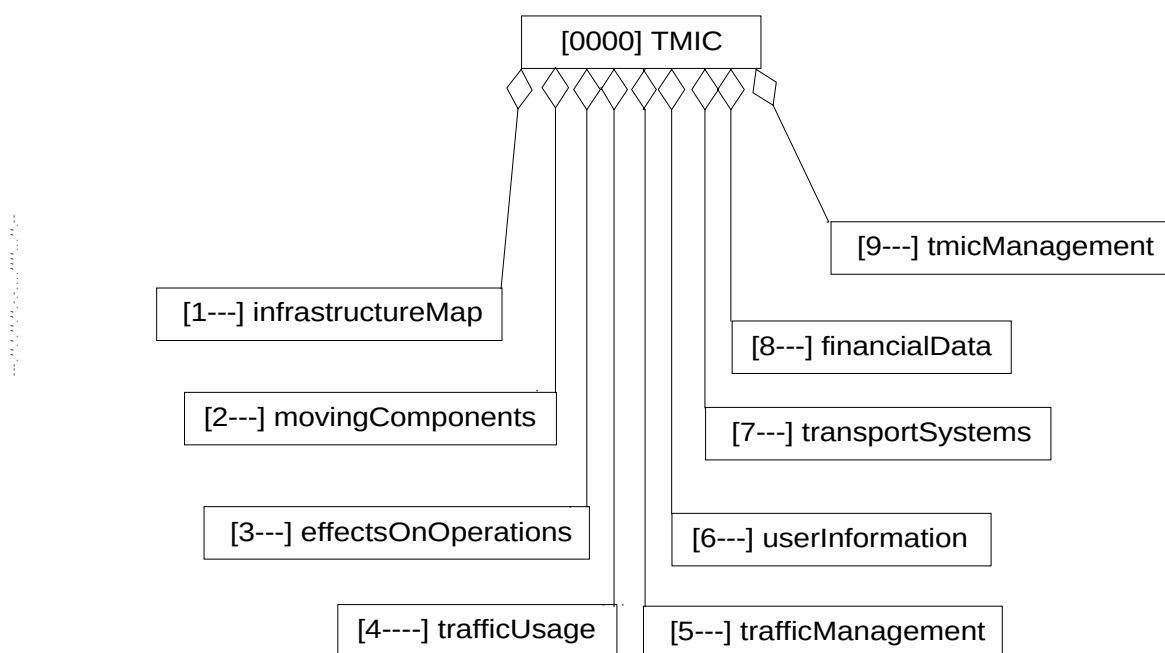


Figure 10 — Top level aggregation of Information classes derived from the 14827 naming tree

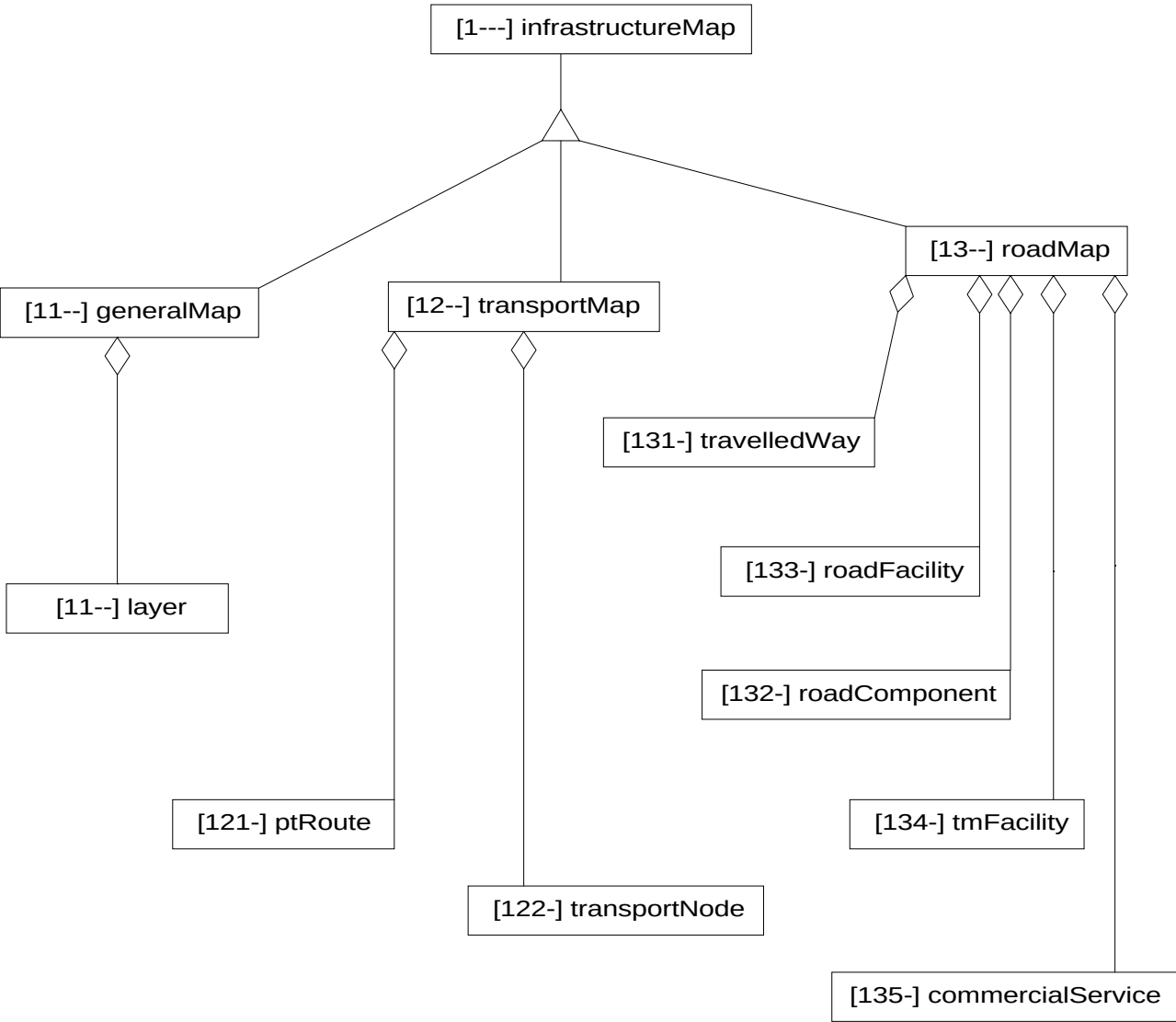


Figure 11 — Information classes in the aggregation of infrastructureMap

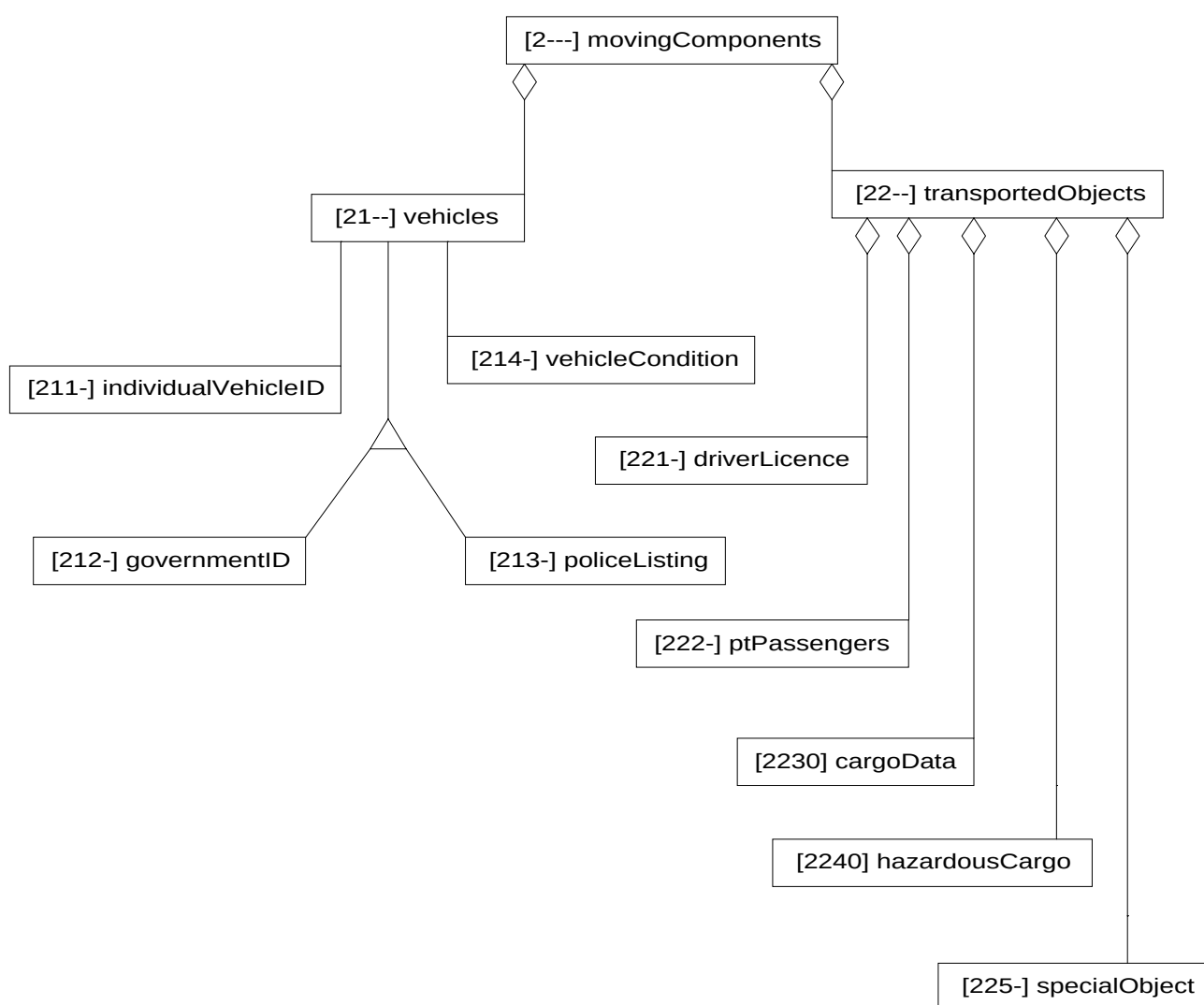


Figure 12 — Information classes in the aggregation of movingComponents

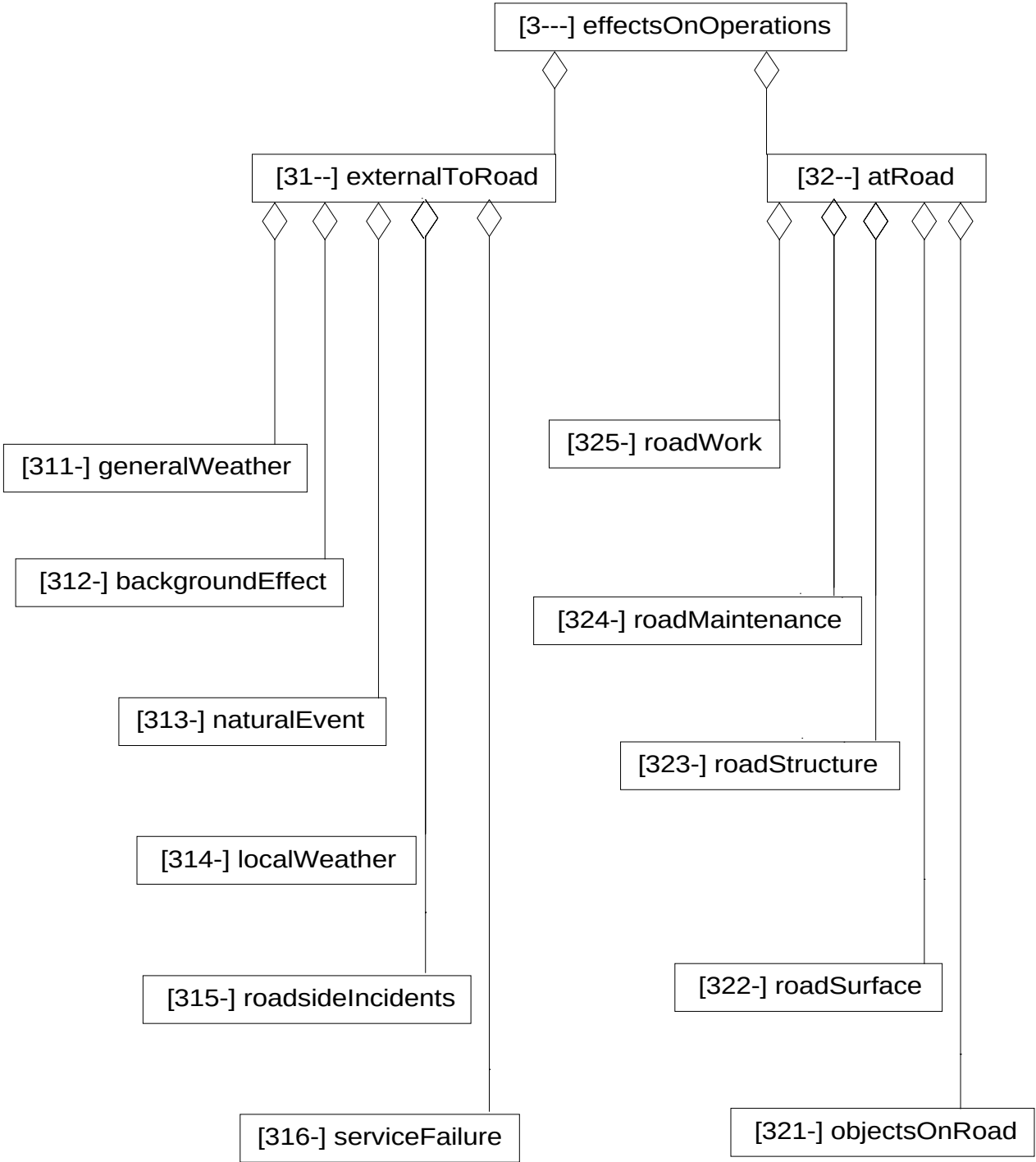


Figure 13 — Information classes in the aggregation of effectsOnOperations

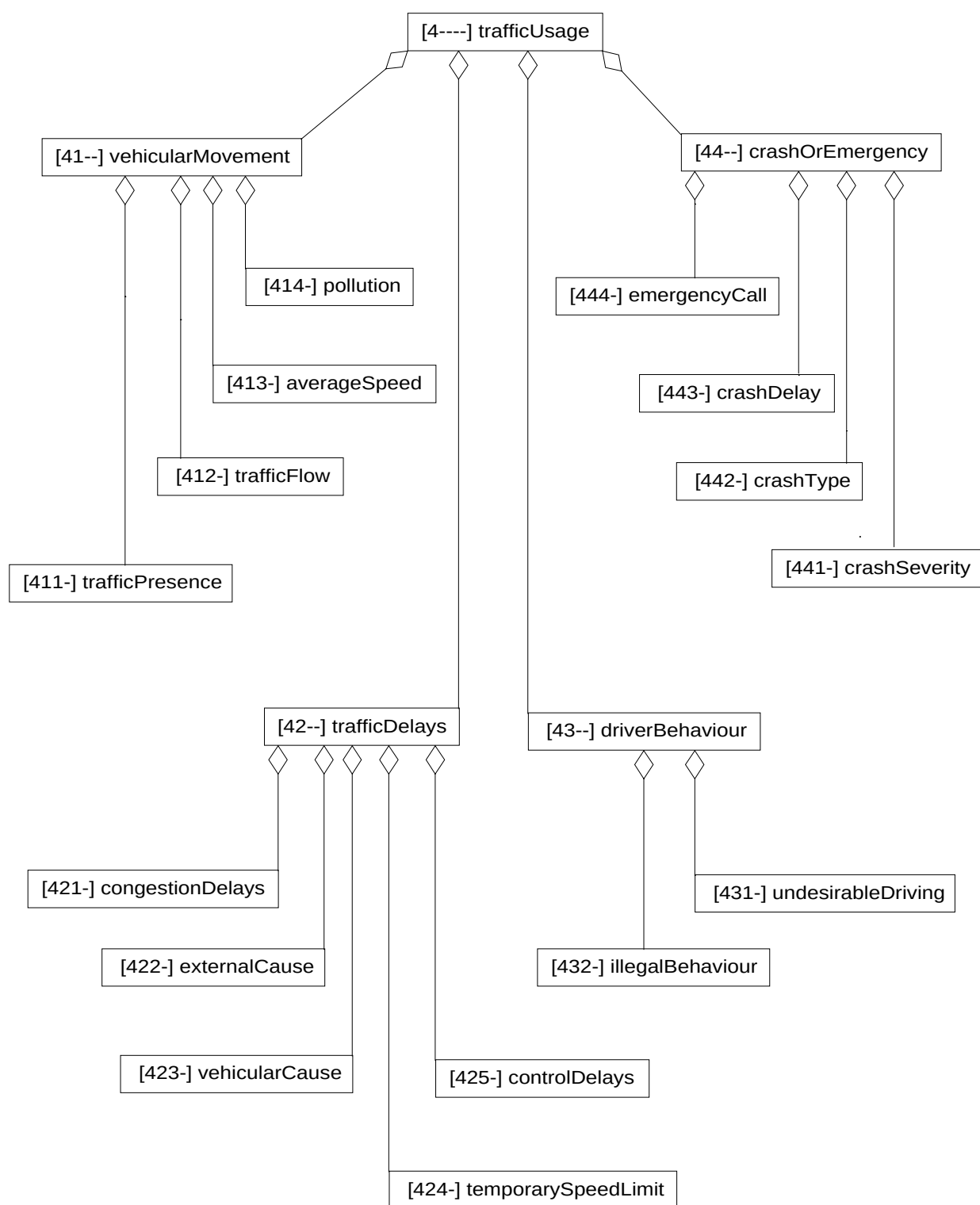


Figure 14 — Information classes in the aggregation of trafficUsage

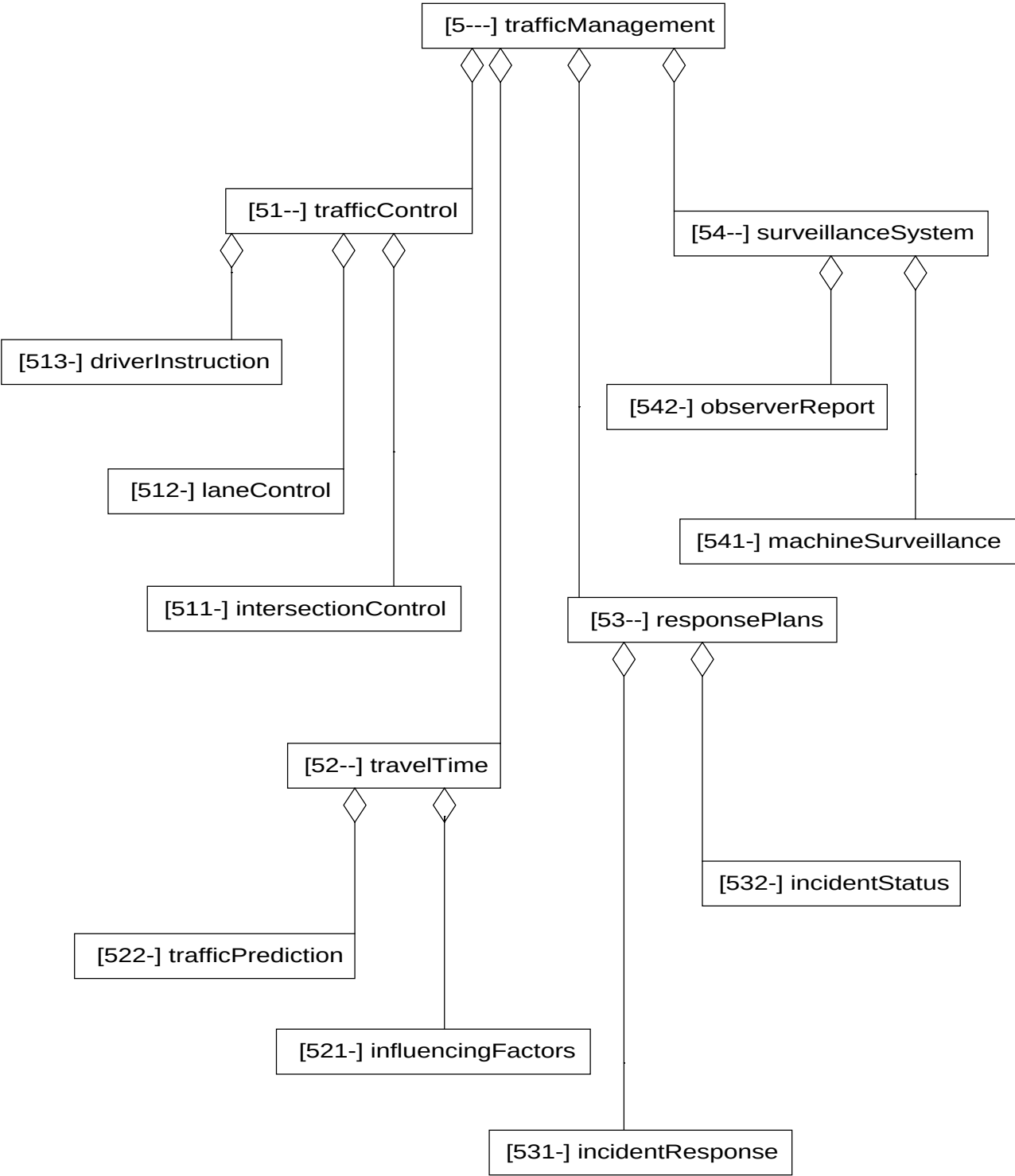


Figure 15 — Information classes in the aggregation of trafficManagement

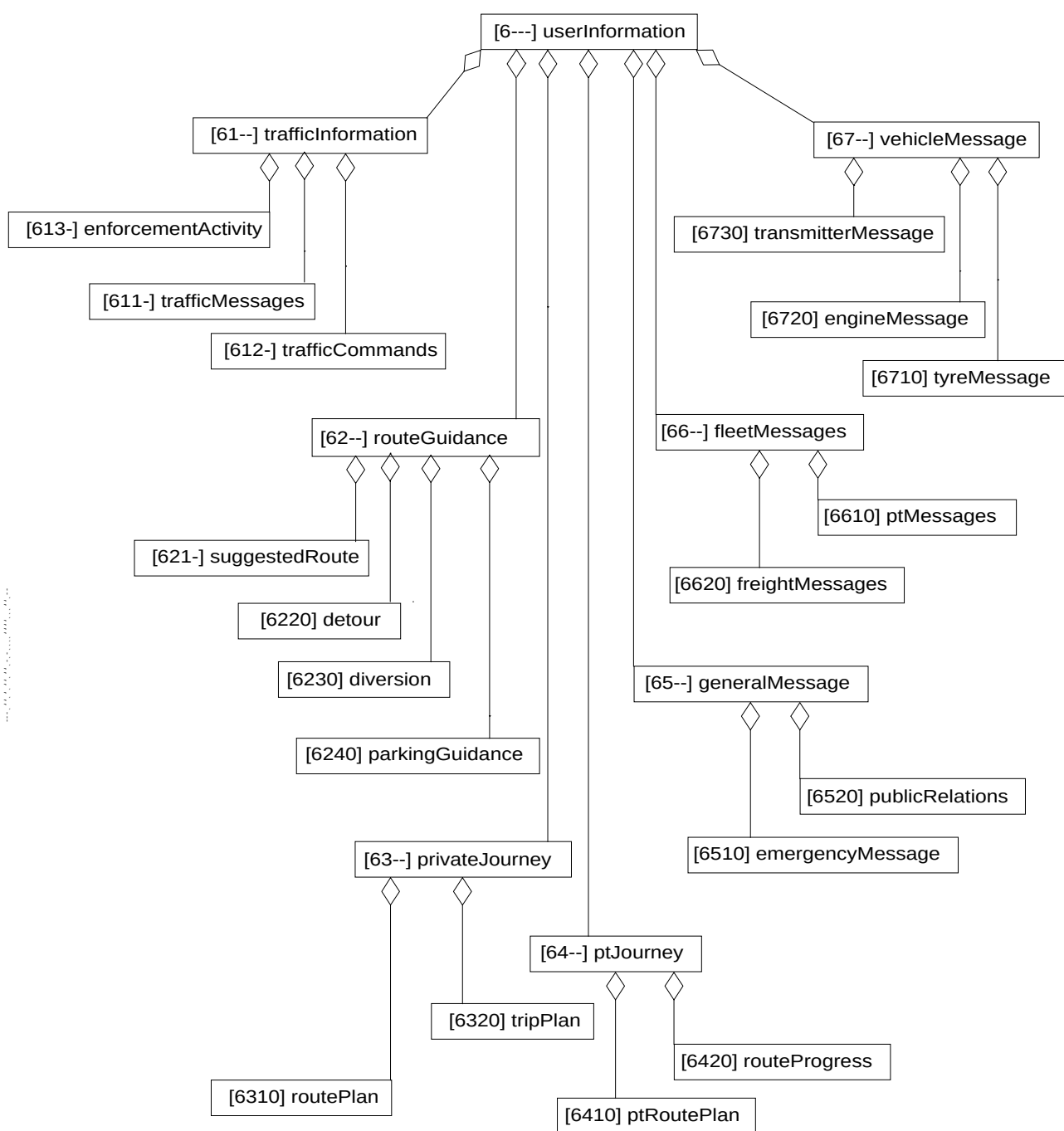


Figure 16 — Information classes in the aggregation of userInformation

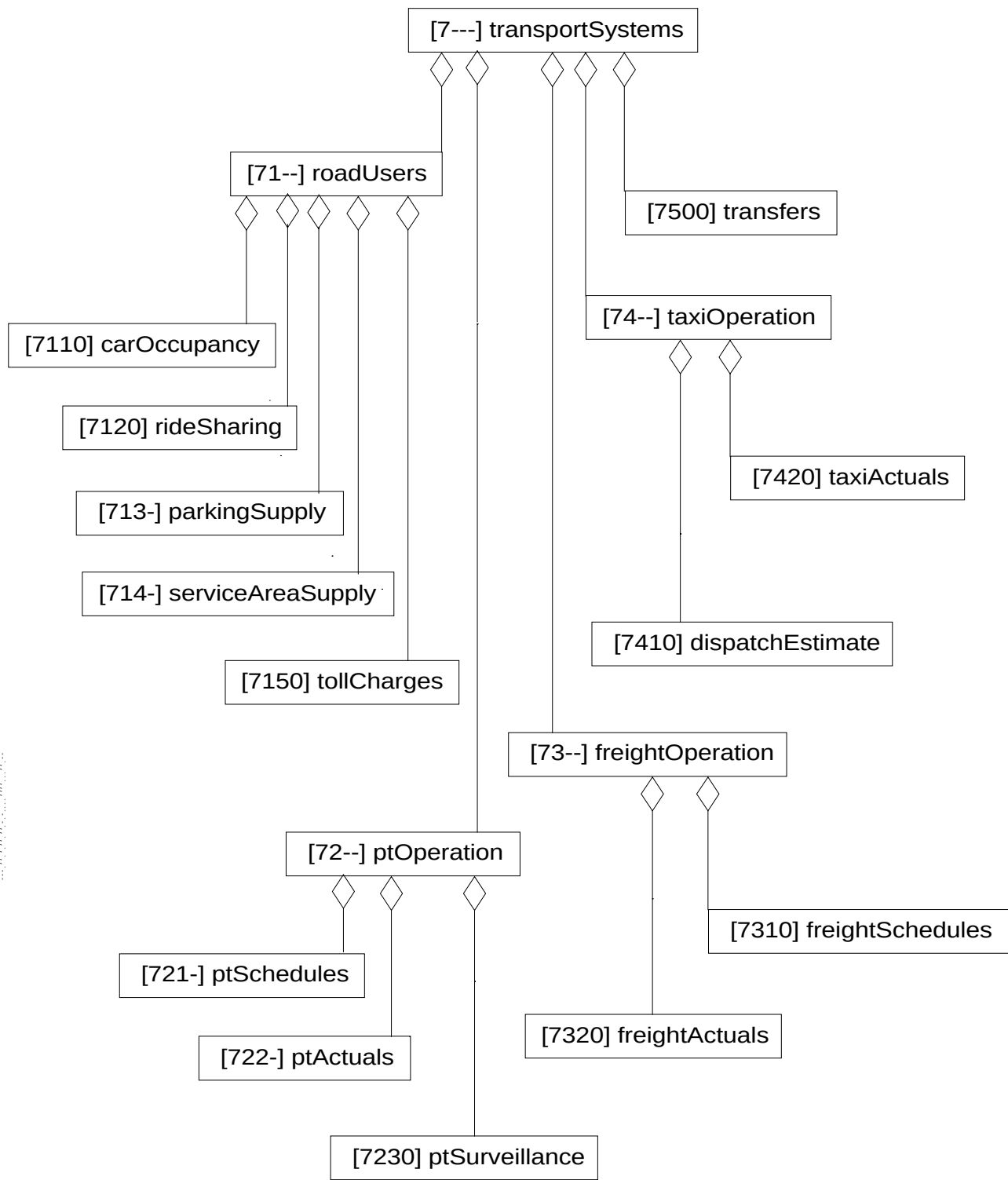


Figure 17 — Information classes in the aggregation of transportSystems

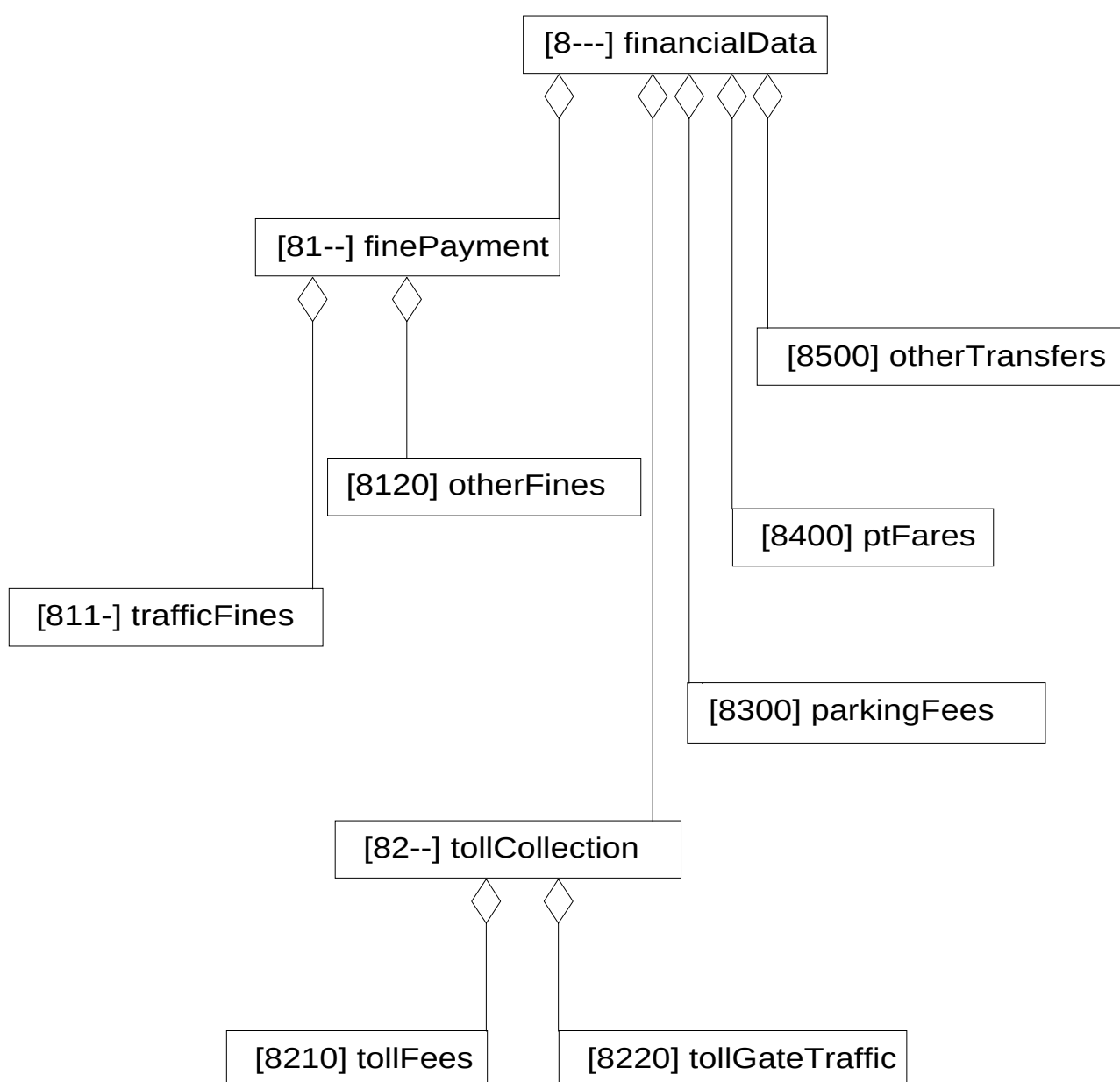


Figure 18 — Information classes in the aggregation of financialData

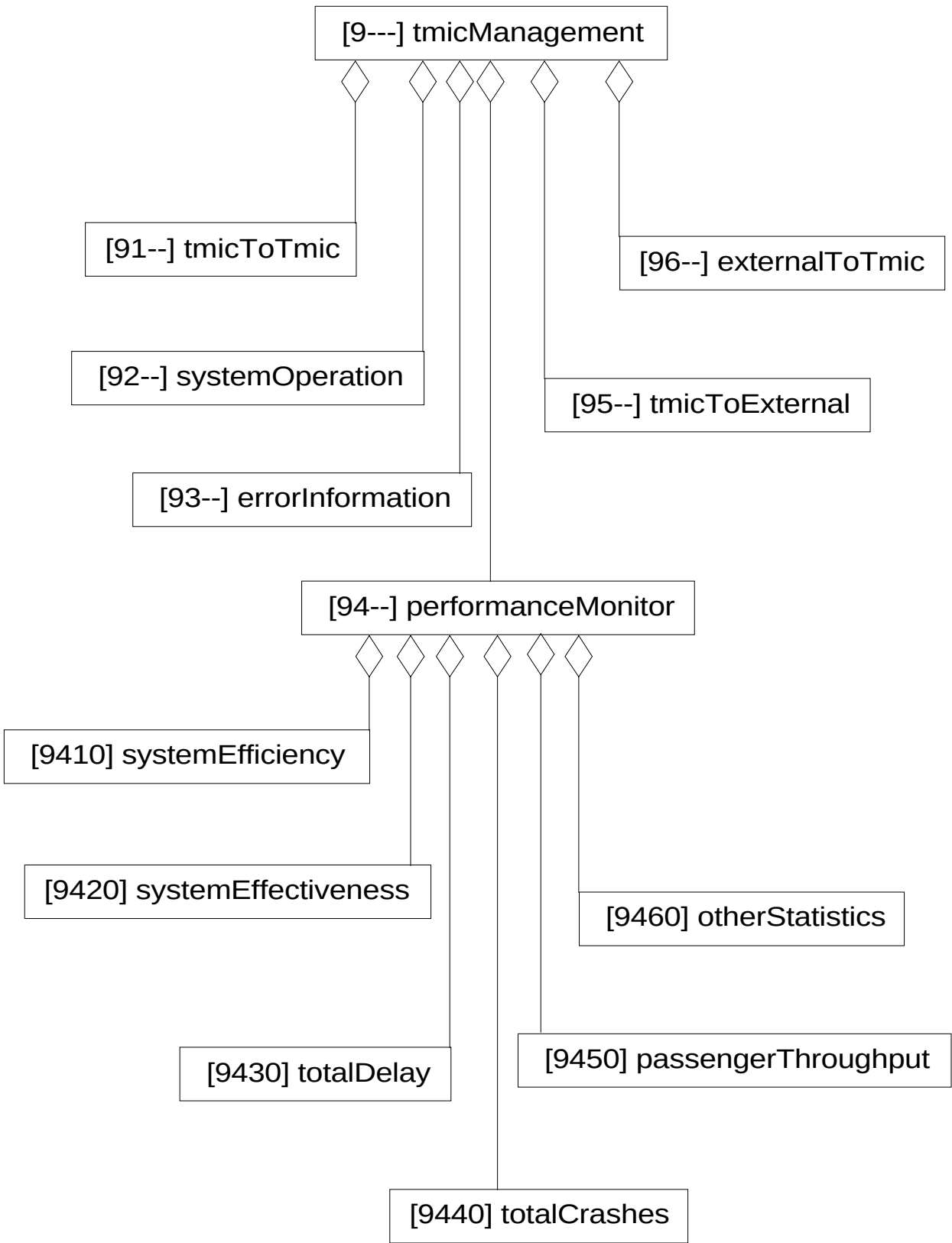


Figure 19 — Information classes in the aggregation of tmicManagement

7 Elaboration of the Sequence Diagrams

This clause presents the results of applying the elaboration method described in clause 5. The primary inputs are the sequence diagrams of the core architecture in ISO/TR 14813 Part 2 and the new classes derived in clause 6 of this document. The object interactions for each use case defined in the abstract sequence diagrams of ISO/TR 14813 Part 2 clause 8 are expanded by mapping each interaction to the expanded set of control classes. In this process the number of object interactions is increased and the class operations defined in ISO/TR 14813 Part 2 are replaced with a finer grained set of operations for the new control classes.

Since sequence diagrams document the logic of the use case, there is a one to one mapping with each use case of the core reference architecture in ISO/TR 14813 Part 2. All the classes involved in a sequence diagram are first identified including some new classes which will be explained as they arise. Any class that provides an operation for an object interaction in a sequence diagram is called a *key class* in this report. Thus key classes are distinguished from classes which only occur as parameters in an object interaction of a sequence diagram.

Within the sequence diagram, each object interaction is represented by a labelled directed line between two objects that are identified by their class name (see Figure 20). The number of pairings of objects is not specified, that is the mapping of an interaction to source and target objects may be one to one, one to many, many to one, or many to many. Also, many occurrences of the same sequence may occur in parallel. The label on the interaction is the name of the operation of the target class which is executed in order to perform that part of a use case transaction. The parameters of the operation (which are transmitted between objects, and which may be input parameters or output parameters or both) are identified in parenthesis after the operation name. In this elaboration the only parameters identified are those defined in 6.3. Thus no parameter list can be regarded as complete, particularly where there are no parameters listed.

Sometimes a parameter of an operation does not map exactly to an information class of 6.3. In these cases, only part of an aggregate object conforming to the specified information class may be relevant as a parameter. If, for example, [41--] is a parameter, this means that there could be one or more of the objects covered by that class aggregation, i.e. [411-] trafficPresence (and recursively any sub-aggregate objects as defined in Annex A), [412-] trafficFlow, [413-] averageSpeed, [414-] pollution. In order to avoid clutter only the digits occurring in the class name are used to identify parameters on the sequence diagram. These digits constitute a unique identifier.

The classes defined in 6.3 have been developed for applications in traffic management and information centres, consequently they are most relevant to the Traffic Management sequence diagrams (7.2) and partly to Traveller Information (7.1). Therefore the development of the sequence diagrams for Traffic Management (7.2) is treated in more detail than that of the other clauses. The object interactions defined for the sequence diagrams in other clauses are largely without parameters.

7.1 Traveller Information

7.1.1 Pre-journey Information

The abstract operations for Pre-journey Information in ISO/TR 14813 Part 2 are provided by the Transport Class. This leads to the following key classes defined in previous clauses.

- O-D Route
- Journey

The following new control and information classes are required:

- Journey Schedule: This class is invented to maintain the mode segments and service times for the trips generated for a journey. The class is associated with the Journey class.
- Public Transport Schedule: A control class which supports the planning and operational functions of a fixed route public transport system.

- Trip: This class is invented to control journey planning and execution. Journeys (for Travellers, Goods Items or Commercial Vehicle Tours) are made up of one or more single mode Trips, thus there is an aggregate association between the Trip and Journey classes. The operations of this class access route information, choose the most suitable route and provide dynamic information.

The sequence diagram for a journey planning exercise is shown in Figure 20. The first stage is to iteratively validate the Traveller's journey request:

- Travel Terminal.initiate journey request ([641-])

in the light of feedback information got by comparing requirements with current conditions recorded by objects of the O-D Route class which in turn interact with objects of the Public Transport Schedule class:

- O-D Route.access current conditions ([611-],[72--],[7120],[7131])
- Public Transport Schedule.access public transport running data ([72--])

The main stage is an iteration whereby the validated Journey specification is first defined:

- Journey.create ([641-])

Some transactions of Route Guidance and Navigation (Figure 21 at line 2.2) are then invoked which develop the individual Trip and Journey Schedule objects associated with the Journey object. The details of the journey plan are then returned to the Traveller through the nested sequence of operations:

- Journey.iterate over trips (...)
- Trip.access next segment (...)
- Trip.access route ([641-])
- Journey Schedule.access schedule ([72--])

If the Traveller commits to this plan the transactions of Journey Payment (7.1.4) are invoked otherwise the journey request is updated and the iteration repeated.

The final transaction is to update the travel statistics:

- O-D Route.update travel date([62--])

7.1.2 Route Guidance and Navigation

The abstract operations for Route Guidance and Navigation in ISO/TR 14813 Part 2 are provided by the Transport Class, the Roadway Class and the Event Class. This leads to the following key classes defined in previous clauses.

- O-D Route
- Journey
- Trip
- Public Transport Schedule
- Roadway Group
- Incident

The following new information class is required.

- Probe: This information class is invented to generate resource use statistics from individual travel records which are provided by the Resource control class.

The sequence diagram is shown in Figure 21. The first part is a maintenance iteration for geographic data:

- O-D Route.update map data ([131-])
- Journey.update map data ([12--])

for traffic data:

- O-D Route.update effects data ([3---])
- Roadway Group.access prediction data ([522-])
- Probe.access route travel time data ([131-],[52--])
- Roadway Group.access traffic data ([52--])
- Incident.access predicted incidents ([5213],[5214])

and public transport data:

- Public Transport Schedule.access public transport running data ([72--])

Related O-D Route objects exchange such data:

- O-D Route.exchange data(...)

and data is prepared for application:

- O-D Route.generate advice(...)

This supports any routine enquiry by Travellers:

- O-D Route.access advisory data ([611-],[7---])

The remainder of the sequence diagram is an iteration for servicing various types of planning, guidance or navigation application. If the request is from a vehicle the journey is defined:

- Journey.create([63--])

The Journey (for whatever payload) is then analysed into component trips and the navigation or guidance data generated:

- Trip.create ([631-],[641-])
- O-D Route.access route conditions([631-],[641-],[52--],[7---])
- Trip.generate route and guidance (...)

If the application is for navigation this is accessed as required:

- Trip.access route guidance ([62--])

If the application is for journey planning the transactions of Journey Schedule (7.1.3) are invoked.

7.1.3 Journey Schedule

The abstract operations for Journey Schedule in ISO/TR 14813 Part 2 are provided by the Transport Class. This leads to the following key classes defined in previous clauses.

- Journey
- Trip
- Journey Schedule

The sequence diagram for each journey planning exercise is shown in Figure 22. The first stage is to initialise a schedule:

- Journey Schedule.create (...)

The main iteration is repeated for each trip:

- Trip.access next segment (...)

For those trips which require reservations the action is taken. The Demand Responsive Public Transport transactions (7.5.3) are invoked if necessary. After each actor interaction the confirmed schedule is recorded:

- Journey Schedule.update time confirmation and price (...)

Once all trips have been processed the complete journey is checked:

- Journey Schedule.validate (...)

7.1.4 Journey Payment

The abstract operations for Journey Payment in ISO/TR 14813 Part 2 are provided by Transport Class and the Payment Class. This leads to the following key classes defined in previous clauses.

- Journey
- Journey Schedule
- Billing Record

The sequence diagram for one journey planning exercise is shown in Figure 23. The first stage is to capture the payment means and commit to payment:

- Travel Terminal.read payment card/tag ([2214])
- Journey.commit ([2214])

The main logic is an iteration whereby each trip in the journey is paid for by invoking the transactions of Payment Transaction (Figure 59 at line 4.4.1):

- Journey Schedule.access segment and cost (...)
- Billing Record.create ([2214])

The final action is to give confirmation:

- Travel Terminal.confirm(...)

7.1.5 On-trip Traveller Information

The abstract operations for On-trip Traveller Information in ISO/TR 14813 Part 2 are provided by the Payment Class. This leads to the following key classes defined in previous clauses.

- Yellow Pages Service
- Billing Record

The sequence diagram for the set of transactions is shown in Figure 24. The first part of the sequence is devoted to the maintenance of information services:

- Yellow Pages Service.update general information (...)
- Travel Terminal.provide yp advertisement (...)

The second part is devoted to the maintenance of yellow pages services:

- Yellow Pages Service.update construct yp service (...)
- Yellow Pages Service.update yp service (...)

The main logic is an iteration for the provision of a yellow pages service:

- Travel Terminal. yp request (...)
- Travel Terminal.read payment card/tag ([2214])
- Yellow Pages Service.yp transaction (...)
- Operating Interface.yp service order(...)
- Billing Record.create ([2214])

followed by use of the Payment Transaction (Figure 59 at line 3) and then

- Travel Terminal.confirm yp request (...)

Independent transactions arise in the case of a Traveller revising plans for the remaining trips of a journey. These are described in Pre-Journey Information (Figure 20 at line 3).

Independent transactions support a Driver on route. These are described in Route Guidance and Navigation (Figure 21 at line 2.3).

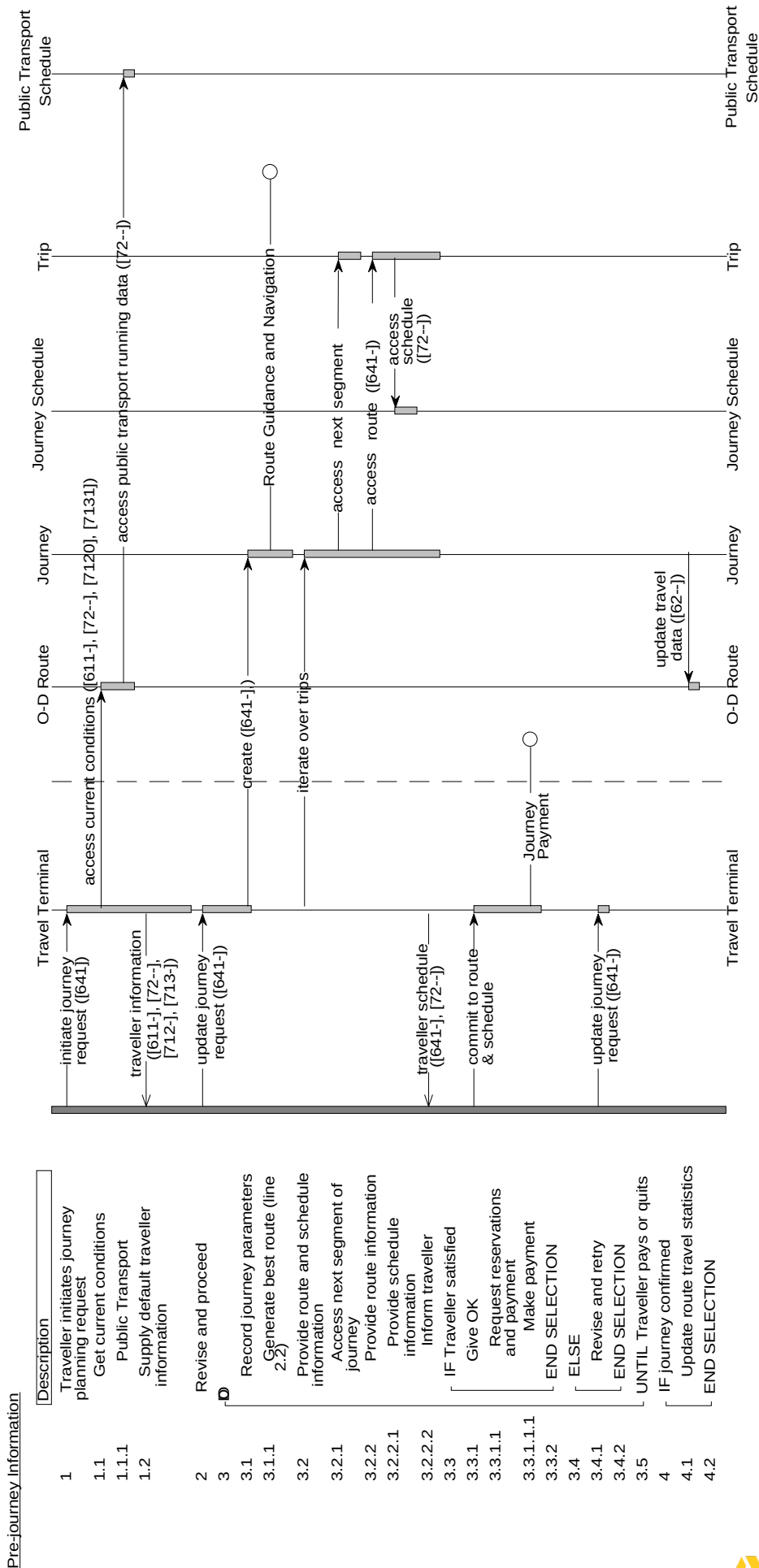


Figure 20 — Sequence diagram for Pre-journey Information

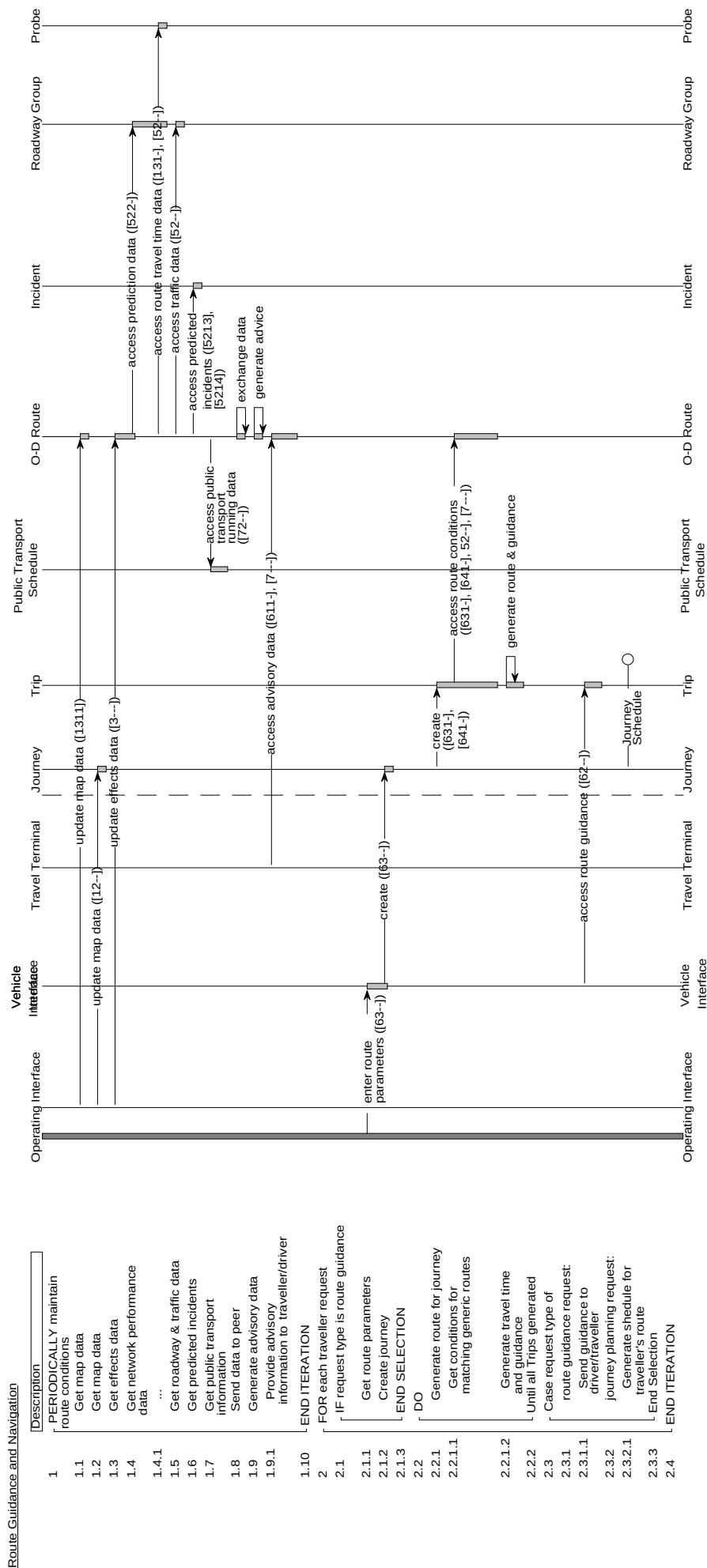


Figure 21 — Sequence diagram for Route Guidance and Navigation

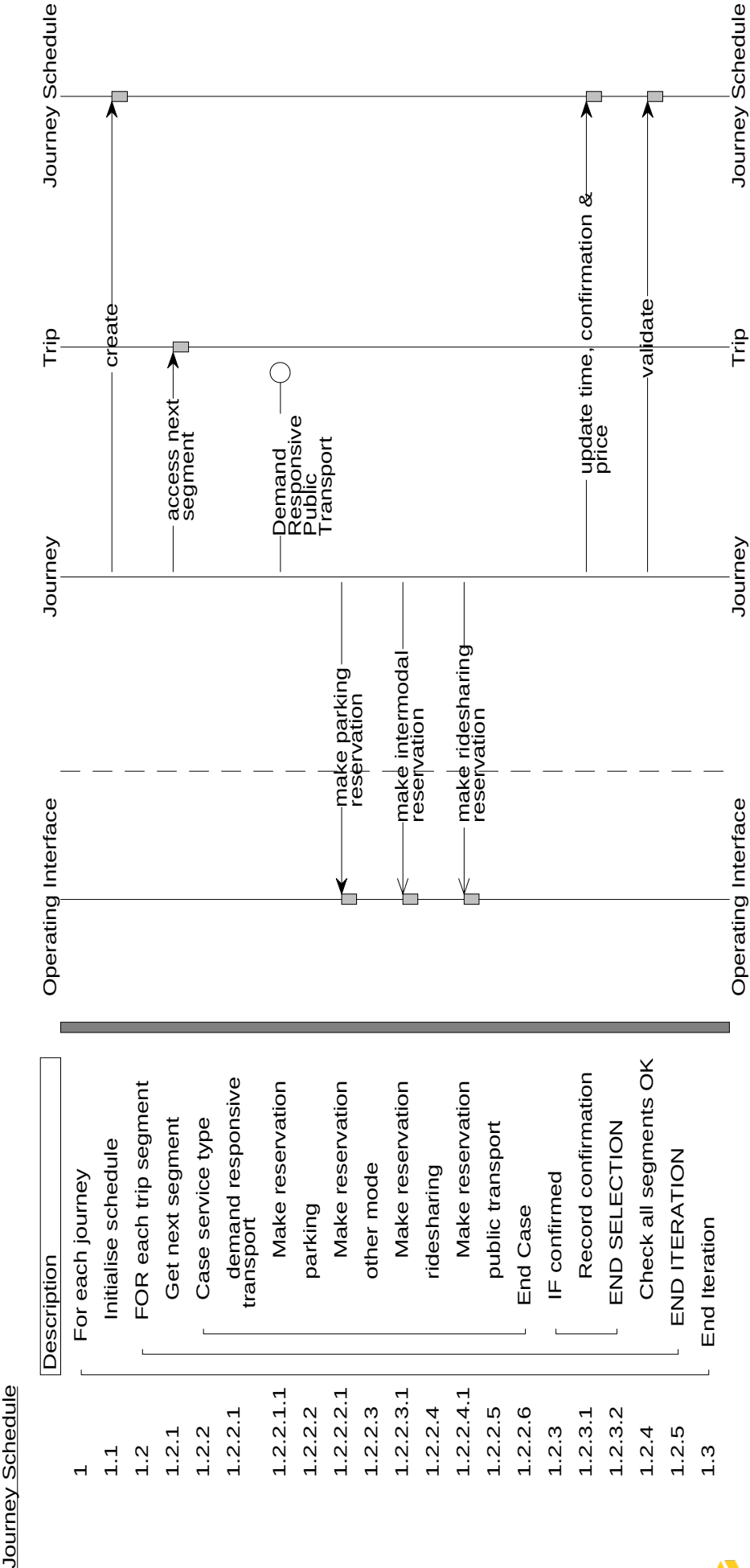


Figure 22 — Sequence diagram for Journey Schedule

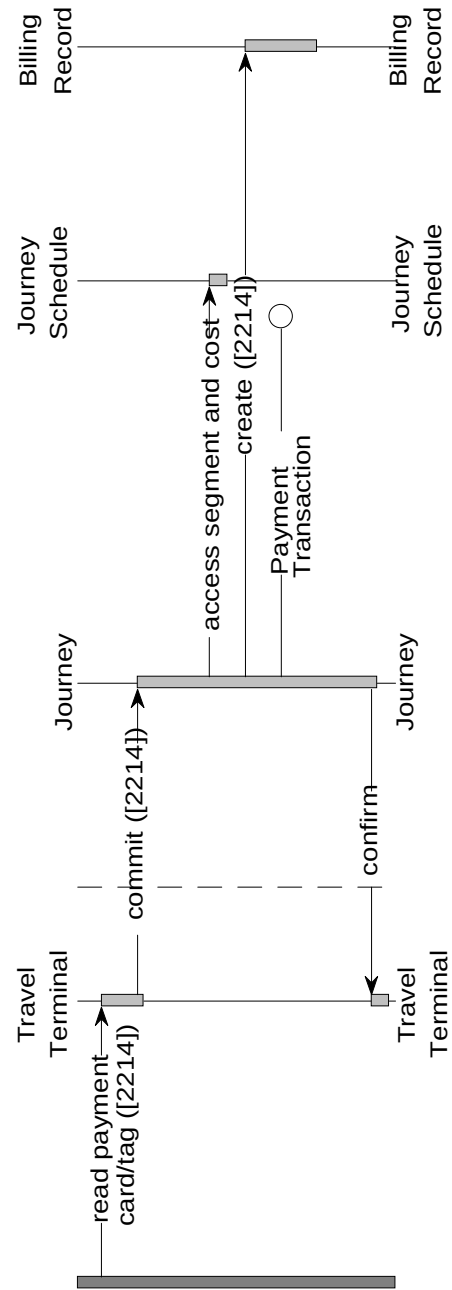


Figure 23 — Sequence diagram for Journey Payment

Journey Payment	
	Description
1	Initiate payment
2	Commit to trip
3	FOR each trip segment
3.1	Get trip schedule
3.2	Make payment
3.2.1	Payment Transaction (4.4.1)
3.3	END ITERATION
4	Confirm

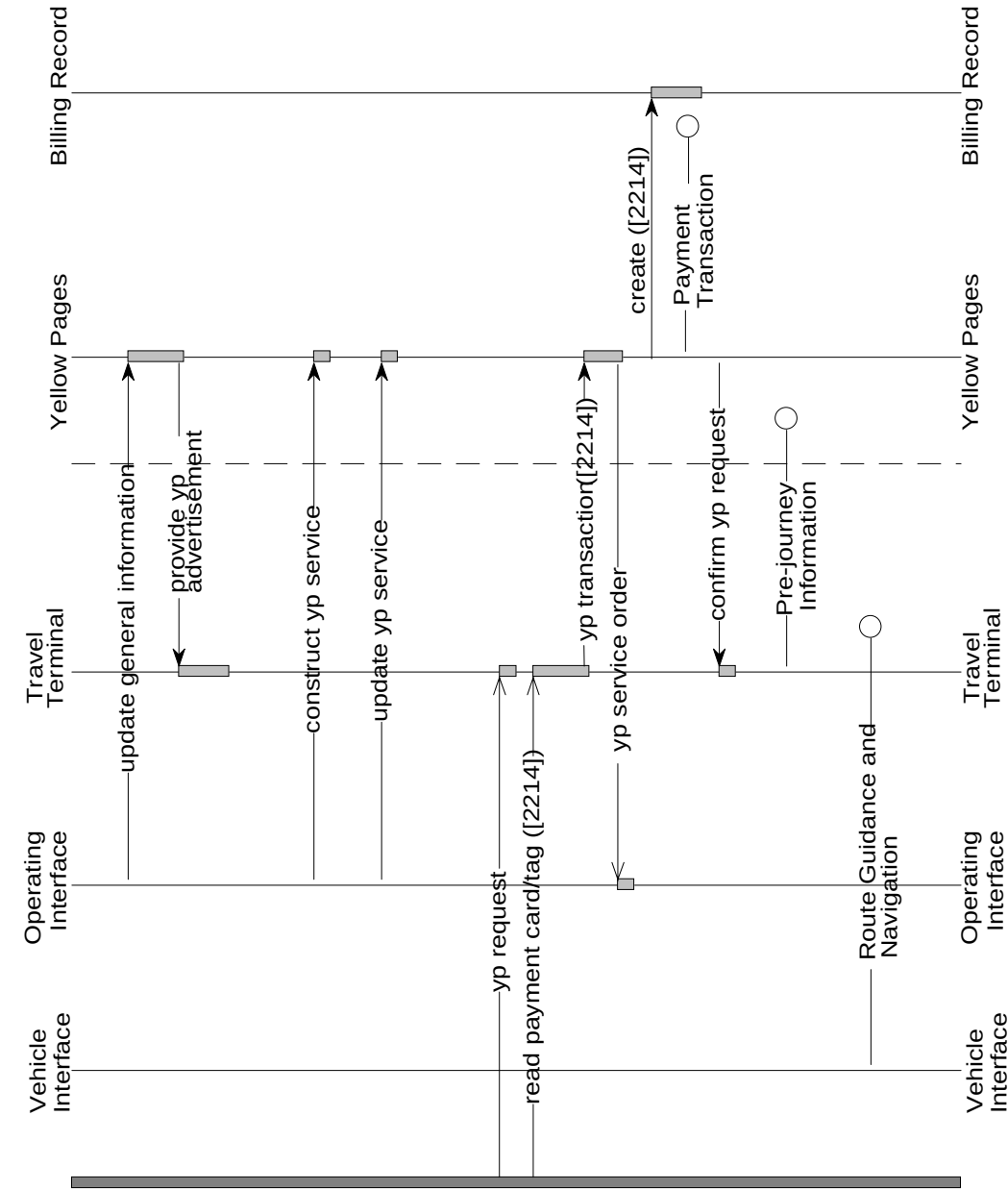


Figure 24 — Sequence diagram for On-trip Traveller Information

On-trip Traveller Information		Description
1	DD	
1.1		Update weather and media information
1.2		Broadcast yellow pages data, weather and media data
1.3		END ITERATION
2		IF new yellow pages service
2.1		Create yellow pages data
2.2		ELSE
2.3		Update yellow pages data
2.4		END SELECTION
3		FOR each yellow pages service request
3.1		Make yellow pages request
3.1.1		Provide payment instrument
3.2		initiate transaction
3.3		Place order
3.4		Make advance payment
3.4.1		Payment Transaction (3)
3.5		Confirm request
3.6		END ITERATION
4		Revise journey (Pre-journey Information (3))
5		WHILE driving
5.1		Get route guidance (Route Guidance & Navigation (2.3))
5.2		End Iteration

7.2 Traffic Management

The information classes identified in 6.3 were developed to meet the message and information contexts of traffic management and information centres. Thus they are highly relevant to this clause. Therefore the elaboration in this clause produces pairs of diagrams for each use case (e.g. Figures 25 and 26). There is one sequence diagram and one class diagram, the latter showing the key classes and parameter information classes which occur in the former diagram. The class diagram partitions the classes by package. In order to avoid cluttering the class diagrams it is usual to associate a parameter information class with a key class in only one package. There is always a key class with which each information class (parameter) can be associated.

7.2.1 Traffic and Pollution Measurement and Control

The abstract operations for Traffic and Pollution Measurement and Control in ISO/TR 14813 Part 2 are provided by the Roadway Class and the Event Class. This leads to the following key classes defined in 6.1.

- Local Control Group
- Roadway Group
- Violation

The following new information classes are required:

- Movement: This information class is invented to describe a stream of traffic and to record measurements about the associated traffic flow. The stream measured might be over a particular manoeuvre, or over a more loosely specified route beginning in a specified road element and ending in a specified road element. It is associated with the Local Control Group class.
- Parking: This information class is invented to record measurements about parking operations provided by Parking actors. It is associated with the Roadway Group class.
- Pollution: This information class is invented to record and process measurements about pollution. It is associated with the Incident class.

The following new interface classes are required:

- Roadside Peripheral: This interface class is invented to interface various signs, detectors and transceivers to TICS.
- Roadside Sensor: This interface class is invented to interface various detectors and transceivers to TICS. It is a subclass of Roadside Peripheral.
- Vehicle Sensor: This interface class is invented to interface special Automatic Vehicle Identification equipment to TICS. It is a subclass of Roadside Sensor.

The sequence diagram is shown in Figure 25. The main logic is a continuing iteration whereby Traffic is measured at the primitive level by sensors and then these measurements are integrated and aggregated in stages by objects of the Local Control Group class and the Roadway Group class to develop the strategic traffic measurements. In parallel with these transactions any violation of regulations by individual vehicles is detected and acted upon.

There are some less frequent maintenance transactions performed, dealing with reference geographic data, operator data display and processing violations. These are placed outside the main loop.

Comments on some of the interactions of the sequence diagram in Figure 25 are given below.

Objects of the interface classes Roadside Sensor and Vehicle Sensor interface between sensor hardware and an object of the Local Control Group class. They invoke two operations in the Local Control Group class:

- Local Control Group.apply traffic regulations ([411-], [412-], [414-])
- Local Control Group.update sensor measurements ([411-], [412-])

In addition an operation in the Pollution class is invoked to derive the area pollution:

- Pollution.update pollution data ([414-])

Annex A contains documentation on each of the parameters, which can also be identified in Figures 11 to 19. If the operation, Local Control Group.apply traffic regulations, determines that an infringement has occurred then the following operations are invoked:

- Violation.create ([411-], [412-], [414-])
- Vehicle Interface.warn driver ([6132])

Objects of the Local Control Group class invoke operations in an object of the Roadway Group to process measurements at the higher level:

- Roadway Group.integrate measurements ([41--])
- Roadway Group.update current data ([412-], [413-])
- Movement.update measurements ([412-])

The Pollution class updates its higher level measurements:

- Pollution.update current data ([414-])

At the end of the iteration any adjustment to the attributes of objects of the Local Control Group or Sensor class is requested:

- Local Control Group.update control data ([51--])
- Roadside Sensor.update control data (...)
- Roadside Peripheral.update control data (...)

The above operations are just two of numerous examples of missing parameters, with no requisite data elements in draft ISO 14827.

Traffic operators may select and display data:

- Roadway Group.support operator access ([4---])

The final stage is used to process any violation and notify the conformance agency:

- Violation.process violation ([81--])
- Operating Interface.report violation ([81--])

The elaboration of the classes of the Roadway and Events packages developed in the above process is summarised in Figure 26, with the key classes denoted by double rectangles. Associations already shown in diagrams in clause 5 may not be included.