SIRI Handbook
& Functional Service Diagrams

Version 0.13
2008/01/10 Njsk Kizoom

DRAFT
## Control sheet

### Version control

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Version</th>
<th>Description of changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006/03/02</td>
<td>Nick Knowles,</td>
<td>V.01</td>
<td>Previous Draft</td>
</tr>
<tr>
<td>2007/04/15</td>
<td>Nick Knowles,</td>
<td>V.02</td>
<td>Word Draft</td>
</tr>
<tr>
<td>2007/10/11</td>
<td>Nick Knowles,</td>
<td>V.07</td>
<td>Word Draft</td>
</tr>
<tr>
<td>2008/01/11</td>
<td>Nick Knowles,</td>
<td>V.12</td>
<td>Revised Draft</td>
</tr>
<tr>
<td>2008/01/11</td>
<td>Nick Knowles,</td>
<td>V.14</td>
<td>Revised Draft</td>
</tr>
</tbody>
</table>
1. Introduction ........................................................................................................................................... 8

Acknowledgements .................................................................................................................................. 8

References ................................................................................................................................................. 8

2. Introduction to SIRI ............................................................................................................................... 10

What is SIRI? .............................................................................................................................................. 10

What Functional Services are available? ..................................................................................................... 11

3. Use of SIRI Services .................................................................................................................................. 13

Separation of Concerns ............................................................................................................................... 14

Communications & message transport ......................................................................................................... 14

Roles & Patterns of Interaction ..................................................................................................................... 14

Request/Response for a SIRI Functional SERVICE ...................................................................................... 15

Publish/Subscribe for a SIRI Functional SERVICE ....................................................................................... 15

Publish/Subscribe with Separate Notification & Fetched Delivery ............................................................... 16

XML Example of a SIRI Functional Service Request & Delivery ......................................................................... 16

Stop Monitoring Request Example ........................................................................................................... 17

Stop Monitoring Request Delivery Example ............................................................................................... 17

Stop Monitoring Subscription Request Example .......................................................................................... 18

Stop Monitoring Subscription Delivery Example .......................................................................................... 18

SIRI & Transmodel ..................................................................................................................................... 20

Relationship of SIRI to TRANSMODEL ....................................................................................................... 20

Transmodel entities relevant for SIRI ............................................................................................................ 21

Stop Places - SIRI & IFOPT ......................................................................................................................... 23

Participant identifiers, Message & Subscription Management .................................................................... 24

Request Response Entities & Message Elements .......................................................................................... 24

publish Subscribe message elements .......................................................................................................... 26

Recovery & Restart ..................................................................................................................................... 28

What does a SIRI Implementation require? .................................................................................................. 28

4. Service Descriptions ............................................................................................................................... 29

© Copyright Kizoom 2006-2007
SIRI Functional Services - UML Diagrams

Organisation.................................................................................................................. 29

Functional Requests & responses.................................................................................. 29

5. SIRI Stop Monitoring (SM)......................................................................................... 31

SIRI-SM: Subscription & Request.................................................................................. 31

  StopMonitoringRequest Summary ................................................................................ 31
  StopMonitoringRequest Detail ..................................................................................... 32

SIRI-SM: Delivery............................................................................................................. 33

  StopMonitoringDelivery Summary ............................................................................... 33
  StopMonitoringDelivery Detail ..................................................................................... 34

Use of MonitoredVehicleJourney Element for SIRI-SM................................................. 36

Examples of Live services that Use SIRI-SM ................................................................. 40

  Map based web departure boards ................................................................................. 40
  WAP mobile departure boards...................................................................................... 40
  SMS departure boards.................................................................................................. 40

6. SIRI Vehicle Monitoring (VM).................................................................................... 41

SIRI-VM: Subscription & Request ................................................................................ 41

  VehicleMonitoringRequest Summary ........................................................................... 41
  VehicleMonitoringRequest Detail ................................................................................ 42

SIRI-VM: Delivery........................................................................................................... 43

  VehicleMonitoringDelivery Summary .......................................................................... 43

Use of MonitoredVehicleJourney Element for SIRI-VM................................................. 44

  VehicleMonitoringDelivery Detail ................................................................................. 45

7. SIRI Production Timetable (PT).................................................................................. 46

SIRI-PT: Subscription & Request .................................................................................. 46

  ProductionTimetableRequest Summary ....................................................................... 46
  ProductionTimetableRequest Detail .............................................................................. 47

SIRI-PT: Delivery............................................................................................................. 48

  ProductionTimetableDelivery Summary ..................................................................... 48
  ProductionTimetableDelivery Detail ............................................................................. 49
SIRI Functional Services - UML Diagrams

8. SIRI EstimatedTimetable (ET) ........................................................................................................ 50

SIRI-ET: Subscription & Request ........................................................................................................ 50

EstimatedTimetableRequest Summary ............................................................................................... 50

EstimatedTimetableRequest Detail ...................................................................................................... 51

SIRI-ET: Delivery ................................................................................................................................... 51

EstimatedTimetableDelivery Summary ................................................................................................. 52

EstimatedTimetableDelivery Detail ...................................................................................................... 53

9. SIRI Stop Timetable (ST) .................................................................................................................. 54

SIRI-ST: Subscription & Request .......................................................................................................... 54

StopTimetableRequest Summary ........................................................................................................... 54

StopTimetableRequest Detail ................................................................................................................ 55

SIRI-ST: Delivery ................................................................................................................................... 56

StopTimetableDelivery Summary ........................................................................................................... 56

StopTimetableDelivery Detail ................................................................................................................ 57

10. SIRI ConnectionTimetable (CT) ...................................................................................................... 58

SIRI-CT: Subscription & Request ........................................................................................................... 58

ConnectionTimetableRequest Summary ............................................................................................... 58

ConnectionTimetableRequest Detail ..................................................................................................... 59

SIRI-CT: Delivery ................................................................................................................................... 60

ConnectionTimetableDelivery Summary ............................................................................................... 60

ConnectionTimetableDelivery Detail ..................................................................................................... 61

11. SIRI ConnectionMonitoring (CM) .................................................................................................... 62

SIRI-CM: Subscription & Request ......................................................................................................... 62

ConnectionMonitoringRequest Summary .............................................................................................. 62

ConnectionMonitoringRequest Detail .................................................................................................. 63

SIRI-CM: Delivery .................................................................................................................................. 63

ConnectionMonitoringDelivery Summary .............................................................................................. 64

ConnectionMonitoringDelivery Detail .................................................................................................. 65

12. SIRI GeneralMessage (GM) .............................................................................................................. 66
SIRI Functional Services - UML Diagrams

SIRI-GM: Subscription & Request ................................................................. 66
  GeneralMessageRequest Summary ......................................................... 66
  GeneralMessageRequest Detail ............................................................... 67
SIRI-GM: Delivery .................................................................................... 68
  GeneralMessageDelivery Summary .......................................................... 68
  GeneralMessageDelivery Detail ............................................................... 69
13. SIRI FacilityMonitoring (FM) DRAFT .................................................... 70
SIRI-FM: Subscription & Request ............................................................... 70
  FacilityMonitoringRequest Summary ....................................................... 70
  FacilityMonitoringRequest Detail ............................................................ 70
SIRI-FM: Delivery .................................................................................... 70
  FacilityMonitoringDelivery Summary ...................................................... 70
  FacilityMonitoringDelivery Detail ........................................................... 70
14. SIRI SituationExchange (SX) DRAFT ..................................................... 71
SIRI-SX: Subscription & Request ............................................................... 71
  SituationExchangeRequest Summary ....................................................... 71
  SituationExchangeRequest Detail ............................................................ 72
SIRI-SX: Delivery .................................................................................... 72
  SituationExchangeDelivery Summary ...................................................... 73
  SituationExchangeDelivery Detail ........................................................... 74
  Situation Model ..................................................................................... 75
15. SIRI Common Data Types ................................................................. 80
Common SIRI Data Types – XML Simple Types ........................................ 80
Common SIRI Data Types ...................................................................... 80
Common SIRI SimPLE Data Types – Codes & Identifiers ........................... 81
Common General SIRI Enumerations ....................................................... 82
SIRI-SX Enumerations ........................................................................... 83
IFOPT Enumerations for SIRI-SX ........................................................... 84
TPEG Miscellaneous Enumerations for SIRI-SX .................................... 85
1. INTRODUCTION

This SIRI Handbook provides a short overview of the CEN SIRI (Service Interface for Real Time Information) technical specification and of the individual SIRI functional services.

It is intended to provide both an introduction to SIRI for technical readers and a quick reference guide for developers who are already familiar with SIRI, but wish to check particular points of detail. It does not cover a number of more detailed aspects of SIRI (for example capability discovery, access controls, etc).

A particular purpose of the handbook is to illustrate the relationship between SIRI and Transmodel by diagramming the SIRI message content in such a way that the reference to Transmodel elements is explicit.

Another goal of this handbook is make understanding a simple use of SIRI for common applications as accessible as possible. As a technical standard intended to support many different implementations and countries, and many different applications and usages, SIRI has a large number of features and optional capabilities. Consequently it requires a large specification. Nonetheless, commonly required basic real-time services, such as stop departures or timetable exchange can be understood and implemented simply in SIRI, and this guide is intended to highlight such straightforward uses.

The guide assumes a technical knowledge of XML and web services and of data modelling with UML notation.

ACKNOWLEDGEMENTS

This document is based on the SIRI technical specification which was developed with sponsorship from VDV, RTIG, Department of Transport and other organizations.

This handbook & diagrams have been prepared & provided by Kizoom.

REFERENCES

- SIRI Specification Service interface for real-time information relating to public transport operations
  - CEN/TS 00278181-1 — Part 1: Introduction
  - CEN/TS 00278181-2 — Part 2: Communications infrastructure
  - CEN/TS 00278181-3 — Part 3: Functional Service Interfaces
- SIRI Schema
  - http://www.siri.org.uk/schema/1.2/siri.xsd
- SIRI White Paper
  - http://www.siri.org.uk/
SIRI Functional Services - UML Diagrams

- Transmodel
  - CEN TC 278 ENV 12896 Reference Data Model for Public Transport

- IFOPT
  - CEN/TS 00278207 – Identification of Fixed Objects in Public Transport

- TPEG
2. INTRODUCTION TO SIRI

WHAT IS SIRI?

SIRI specifies a European data interface standard for exchanging information about the planned, current or projected performance of real-time public transport operations between different computer systems.

- SIRI comprises a modular set of discrete functional services for exchanging data for public transport information systems. Services cover planned and real time timetable exchange; vehicle activity at stops; vehicle movement; and information to assist in the provision of reliable connections between vehicles. Services may be implemented and used individually.

- SIRI aims to incorporate the best of various national and proprietary standards from across Europe and deliver these as web services using a modern XML schema.

- The services assume a standard conceptual model for the data to be exchanged, based on the CEN Transmodel data reference model. Element names and data structures are based on this model. SIRI was developed for bus data, but can be used just as well for other modes of transport such as Rail and Air.

- All SIRI services are provided over a standardised Communications layer, based on a Web Services Architecture.
  - The Communications layer upholds a consistent approach for all the functional services to Security, Authentication, Version Negotiation, Recovery/Restart, and Access Control/Filtering.
  - To support different operating requirements, two main patterns of interactions are supported: an immediate Request/Response protocol; and an asynchronous Publish/Subscribe protocol. The Publish/Subscribe can be further elaborated with a fetched delivery interaction to optimise the use of bandwidth.

Figure 2-1 SIRI Services
WHAT FUNCTIONAL SERVICES ARE AVAILABLE?

SIRI currently comprises the following functional services (See Figure 2-1):

- **Stop Timetable (ST) and Stop Monitoring services (SM)** provide stop-centric information about current and forthcoming vehicle arrivals and departures at a nominated stop or Monitoring Point, typically for departures within the next 20-60 minutes for display to the public.
  - The SM service is suited in particular for providing departure boards for web services and on all forms of device.

- **Vehicle Monitoring service (VM)** provides information about the current location and expected activities of a particular vehicle, and can give the current and subsequent Journey and the Calling points on each journey, together with the scheduled and expected arrival times.
  - The VM service is suited in particular for onboard displays, and visualisation of vehicle movement, and for exchanging information on roaming vehicles between different control systems. It can be used for example to support a moving image showing the position of a bus on a map.
  - It also constitutes a detailed logging feed suitable for collecting historic about performance against schedule.

- **Production Timetable** service (PT) exchanges information about the expected operation of a transport network for a specified day in the near future.
  - Typically this is produced a few hours or days before the day in question and incorporates any changes to the timetables known at that stage. It can of course also be used to distributed timetables long in advance.
  - A Production Timetable can be filtered by Operator, Line and Date Range, allowing only the section of the timetable of interest to be selected.
  - Suited for provisioning AVL systems and smart devices with base timetables.

- **Estimated Timetable** service (ET) provides details of the operation of the transport network for a period within the current day, detailing real time deviations from the timetables and control actions affecting the Timetable (cancellations, additional Journeys and Detours).
  - An estimated timetable can be filtered by Operator or by Line, allowing only the section of the timetable that is of interest to be selected.
  - Suited for provisioning AVL systems and smart devices with real-time timetables.
  - Also suitable for the bulk exchange of data.

- **Connection Timetable (CT) and Connection Monitoring** service (CM) allow transport operators to exchange information about the real-time management of interchanges between feeder and distributor vehicles arriving and departing at a connection point, for example, to let passengers on a delayed train know that a local bus service will wait for them.
SIRI Functional Services - UML Diagrams

- It can be used in particular for Guaranteed Interchange (‘Connection protection’) services.

- **General Messaging Service (GM)** provides a structured way to exchange arbitrary informative messages between participants, such as travel news, or operational advice.
  - It can be used to link together incident management systems in a store and forward architecture.

Two Further services are under development as SIRI version 1.3

**Facilities Monitoring Service (FM)** provides a information about changes in availability of equipment.

It can be used to link together equipment and incident management systems in a store and forward architecture.

**Situation Exchange Service (SE)** provides a fully featured service for exchanging planned and unplanned incidents between control centres and distribution systems.

It includes a structured incident model can be used to integrate incident management systems with other SIRI services and with external services.
3. USE OF SIRI SERVICES

All the SIRI Functional services follow a common design pattern.

- SIRI is for the exchange of data between participant services. The server which provides the data is the **Producer**. The server which obtains the data is the **Consumer**.

- The same servers may sustain more than one different SIRI functional service, and support either or both Consumer and Producer roles as appropriate. The Producer service itself may be broken down into further roles (e.g. publisher, notification producer) though a Consumer does not need to be aware of these constituent roles.

- Each functional service has unique **endpoints**, the addresses at which the service can be found.

- Each **Participant**, whether consumer or producer, has a unique participant identifier.

- **Payload** content is separated from message management content.

- Content may be obtained from a Producer either by an immediate direct request/response interaction, or by an asynchronous publish/subscribe interaction. In both cases the same set of request parameters are supported to specify the desired response content; for a subscription the request is wrapped in an additional subscription request element.

- Data exchanged by SIRI services may either be used as input to other computations, or be rendered into end user information views in any one of a number of media: e.g. **web** (HTML, javascript, etc), **WAP** (xHTML), **SMS, Smartphones** (j2ME, etc) or **Voice** (VML etc).

Figure 3-1 illustrates the concept of using multiple SIRI services to connect up different types of PT Information application, for example AVMS, Journey Planners, Incident Capture systems, Web delivery of stop departure and incidents, alerts, etc.
SIRI Functional Services - UML Diagrams

Another common use of SIRI would be to separate the load of the real-time systems that manage and operate the public transport from the wider passenger information systems that distribute the data. Thus for example, a SIRI push service could send real-time predictions from the AVMS system to a separate service that supports passenger queries by web and mobile. This allows the latter to be scaled without affecting the reliability of the operational system.

SEPARATION OF CONCER NS

SIRI is designed to separate different software architectural concerns, in particular for communications/content and for scaling.

COMMUNICATIONS & MESSAGE TRANSPORT

SIRI separates message content from message transport related elements so that it may be used with different methods of communication, for example XML over http, XML over SOAP/WSDL. Although a default method of http is currently used, it would be possible to use the SIRI content model, with quite different transport – for example TCP IP sockets.

In order to specify an individual SIRI functional service (or indeed any web service), three different aspects of the service need to be specified; (i) the data content model – for SIRI based on Transmodel; (ii) the Communications Transport protocol used to serialise and exchange messages (for example, http & XML) and; (iii) the exchange behaviour – the rules for use of the each functional protocol as a dynamic sequence of messages subject to certain application dependent behaviour. Figure 3-2 illustrates the concept of separation of concerns. In SIRI the Data content is described as XML messages. In addition, the variables controlling the parameterised parts of the exchange behaviour are specified as XML values in a configuration or capability description.

ROLES & PATTERNS OF INTERACTION

SIRI can be regarded as a software framework: all the SIRI Functional Services are based on common patterns of interaction, populated with specific messages and data elements specific to a particular service. These patterns distinguish the various software roles which occur in real-time data exchange, for example Consumer, Producer, Notification Producer, Subscriber - even if these are combined on a single server for a particular implementation. The main patterns can be summarized as follows:

- Functional Service Request Response.
SIRI Functional Services - UML Diagrams

- Functional Service Subscription with asynchronous Direct Delivery.
- Functional Service Subscription with asynchronous Notification & Fetched Delivery.

REQUEST/RESPONSE FOR A SIRI FUNCTIONAL SERVICE

For the Request / Response interaction, the Consumer makes a Functional Service Request to the Producer, and receives a Functional Service Delivery message in response. The Functional Service Request may state the filtering criteria as Topics (the desired domain element references) and Policies (criteria as to how the results are to be filtered. The Topics and Policies depend on the specific functional service.

![Sequence Diagram of Request Response Interaction](image)

Figure 3-3 Sequence Diagram of Request Response Interaction

A Functional Service Request is always for a specific SIRI service, e.g. Stop Monitoring, General Message, etc. Note that the request / response pattern is also widely in SIRI for the exchange of information as pairs of messages, for example for status checking, to create subscriptions, etc.

PUBLISH/subscribe FOR A SIRI FUNCTIONAL SERVICE

For the Publish / Subscribe interaction, a Subscriber makes a Subscription Request to the Producer, which creates a Subscription for a Consumer (usually but not necessarily the same server as the Consumer) for the requested lease period for a specified Functional Service Request and Subscription Policy. The Producer will subsequently send one or more asynchronous Delivery messages if the Request criteria are met and continue doing so in accordance with the subscription policy, up to the end of the lease. The Filtering criteria can include change thresholds or ‘hysteresis’ values. The Subscription will eventually expire or be terminated.
In the simplest use, SIRI services follow of the normal web service paradigm of **direct delivery**, returning content in response to a request or a subscription. To allow additional optimisation of bandwidth and device queuing, and to provide compatibility with legacy systems, SIRI also supports an additional delivery variant of **fetched delivery** in which data is returned in two steps: first a notification message from Producer to Consumer that data is available, then a Request/Response interaction from Consumer to Producer to fetch the data using a delivery message. The payload content of the delivery message is the same as for the delivery of a direct response.

**XML EXAMPLE OF A SIRI FUNCTIONAL SERVICE REQUEST & DELIVERY**
The following two XML code fragments illustrate a request/response message pair for a Functional Service - the SIRI-SM Stop Monitoring service, populated to support a simple departure board.

**STOP MONITORING REQUEST EXAMPLE**

The following is an example of a StopMonitoringRequest. The request consists of a standard header, which is similar for all SIRI requests, and then Topics and Policies that are specific to the SIRI-SM functional service. The request asks for the departures for a stop ‘HLTST011’. The response is to contain up to seven vehicle journeys at the stop, as set by the policy. If there are more than seven available, then only the first two for each line will be shown.

```xml
<ServiceRequest>
  <RequestorRef>NADER</RequestorRef>
  <RequestTimestamp>2004-12-17T09:30:47-05:00</RequestTimestamp>
  <StopMonitoringRequest version="1.0">
    <!-- All LINE77 services from stop EH00001 to destination PLACE457 in the next 30 minutes -->
    <RequestTimestamp>2004-12-17T09:30:47-05:00</RequestTimestamp>
    <MessageIdentifier>NDR06756</MessageIdentifier>
    <!-- TOPIC ================ -->
    <PreviewInterval>P30M</PreviewInterval>
    <MonitoringRef>HLTST011</MonitoringRef>
    <!-- POLICY =================== -->
    <MaximumStopVisits>7</MaximumStopVisits>
    <MinimumStopVisitsPerLine>2</MinimumStopVisitsPerLine>
    <StopMonitoringDetailLevel>normal</StopMonitoringDetailLevel>
  </StopMonitoringRequest>
</ServiceRequest>
```

**STOP MONITORING REQUEST DELIVERY EXAMPLE**

The following is an example of a StopMonitoringDelivery in response to a StopMonitoringRequest. It shows a single StopVisit for a detail level of Normal. The delivery consists of a standard header, which is similar for all SIRI requests, and then payload that is specific to the functional service.

```xml
<ServiceDelivery>
  <!-- HEADER =============== -->
  <ResponseTimestamp>2004-12-17T09:30:46-05:00</ResponseTimestamp>
  <ProducerRef>KUBRICK</ProducerRef>
  <!-- PAYLOAD =========== -->
  <StopMonitoringDelivery version="0.1d">
    <RequestMessageRef>NDR06756</RequestMessageRef>
    <!-- FIRST ARRIVAL ================ -->
    <MonitoredStopVisit>
      <RecordedAtTime>2004-12-17T09:25:46-05:00</RecordedAtTime>
      <MonitoringRef>HLTST011</MonitoringRef>
      <MonitoredVehicleJourney>
        <!-- JOURNEY IDENTITY GROUP -->
        <LineRef>Line123</LineRef>
        <DirectionRef>Out</DirectionRef>
        <FramedVehicleJourneyRef>
          <DataFrameRef>2004-12-17</DataFrameRef>
          <DatedVehicleJourneyRef>03567</DatedVehicleJourneyRef>
        </FramedVehicleJourneyRef>
      </MonitoredVehicleJourney>
    </MonitoredStopVisit>
  </StopMonitoringDelivery>
</ServiceDelivery>
```
STOP MONITORING SUBSCRIPTION REQUEST EXAMPLE

The following further example shows a StopMonitoringSubscriptionRequest to set up a subscription to be sent updates for a stop asynchronously. It embeds a StopMonitoringRequest which specifies the selection criteria - to see all departures for LINE23 from stop POIT5678 for up to 30 minutes ahead, including the calling pattern. It adds some values to set the subscription policy: updates are only sent if the values have changed by more than 2 Minutes. The producer would return

STOP MONITORING SUBSCRIPTION DELIVERY EXAMPLE

The following is an example of a StopMonitoringDelivery in response to a StopMonitoringSubscriptionRequest. It is almost identical to the earlier response to a StopMonitoringRequest, but also includes subscription identifier data.
<ResponseTimestamp>2004-12-17T09:30:46-05:00</ResponseTimestamp>
<ProducerRef>KUBRICK</ProducerRef>
<!-------------------PAYLOAD--------------------------------------------------->
<StopMonitoringDelivery version="0.1d">
  <ResponseTimestamp>2004-12-17T09:30:47-05:00</ResponseTimestamp>
  <SubscriberRef>NADER</SubscriberRef>
  <SubscriptionRef>2004-12-17T09:30:47-05:00</SubscriptionRef>
  <ValidUntil>2004-12-17T09:30:47-05:00</ValidUntil>
  <!----------------FIRST ARRIVAL ----------------------------------------------->
  <MonitoredStopVisit>
    <RecordedAtTime>2004-12-17T09:25:46-05:00</RecordedAtTime>
    <MonitoringRef>HLTST011</MonitoringRef>
    <MonitoredVehicleJourney>
      <!-- JOURNEY IDENTITY GROUP -->
      <LineRef>Line123</LineRef>
      <DirectionRef>Out</DirectionRef>
      <FramedVehicleJourneyRef>
        <DataFrameRef>2004-12-17</DataFrameRef>
        <DatedVehicleJourneyRef>03567</DatedVehicleJourneyRef>
      </FramedVehicleJourneyRef>
      <!-- JOURNEY PATTERN INFO GROUP -->
      <PublishedLineName>123</PublishedLineName>
      <DestinationName>Paradise Park</DestinationName>
      <VehicleRef>VEH987654</VehicleRef>
      <!-- CALL AT STOP -->
      <MonitoredCall>
        <VisitNumber>0014</VisitNumber>
        <VehicleAtStop>false</VehicleAtStop>
      </MonitoredCall>
    </MonitoredVehicleJourney>
  </MonitoredStopVisit>
</StopMonitoringDelivery>
The ‘payload’ content of SIRI messages is intended to be compatible with Transmodel, the CEN Reference Data Model for Public Transport, which sets out common terminology and data abstractions for representing public transport information systems. Thus it should be possible to map data readily from a Transmodel compliant database or application into SIRI messages for data exchange and vice versa, and someone familiar with Transmodel should be able to recognize the application payload elements of the SIRI content model.

- SIRI uses Transmodel terms and data elements wherever they exist, and assumes the same cardinalities as Transmodel for any associations between elements.
- In some cases, SIRI introduces additional elements or attributes for additional concepts not yet in Transmodel. Mostly this consists of adding additional attributes to existing Transmodel concepts.
- In certain other cases, additional elements are introduced in SIRI for convenience to group existing elements, for example, a SIRI ‘Call’ element groups the Transmodel entities STOP IN SEQUENCE, various PASSING TIMES, and other additional attributes. These additional elements do not break conformity with Transmodel but can be regarded as “implementation views.”

Although SIRI is based on Transmodel, there is nonetheless some subtlety involved in understanding the mapping of SIRI content onto Transmodel. Most SIRI functional services are only concerned with exchanging small parts of the information models described by Transmodel - and need to perform this exchange efficiently in real-time. The serialisation of the abstract Transmodel content model into concrete SIRI messages is therefore selective: only those specific elements that are needed for each particular functional service are included; often these will be just the changes to a previously established data set, rather than the full data set. As a consequence, SIRI message content cannot and should not be read as a full or canonical representation of Transmodel, even for the subset of it with which it is concerned. Nor can all the associations of Transmodel be discerned from just their use in SIRI messages.

- When exchanging real-time information, there is frequently a need to exchange just one or two properties of an associated entity without exchanging the full definition of that entity. For example, when exchanging information about a vehicle journey it may be useful to know certain attributes of its journey pattern - or of the attributes that can be found though knowledge of that journey pattern by navigating the associations from the vehicle journey to its related entities and onwards. Similarly when exchanging a real-time data for a ‘Call’ at a stop, it may be useful to include the stop name– even though this is properly part of a separate Stop Point entity and could be found through the with a stop point identifier on the call element. To handle this practical requirement, SIRI makes use of groups of elements that include the necessary identifiers of the referenced entities, annotated with selected attributes. In a few cases in SIRI these groupings are implemented as additional entities, for example a SIRI Call. In most other cases the SIRI XMI schema simply includes groups of attributes from associated elements at the appropriate point in payload content (indicated in diagrams by a <<group>> stereotype). Again, such groups should be regarded as implementation artefacts, i.e. views of small parts of the Transmodel model that are...
introduced for convenience, derived through Transmodel associations and which can be mapped back to a full Transmodel representation on source and destination systems.

- There is also a need to support **different levels of detail** for population of the message content for different applications and different implementations. For example, some departure boards will show just a single departure time for each service, others will wish to show for each service a full calling pattern for all previous and onwards stops, with arrival and departure times at each destination. Most attributes and associations are therefore optional in SIRI messages, even if they are mandatory in the underlying data model.

- Some SIRI services are designed to exchange or differences in values to previously exchanged values, or in some cases differences that differ by more than an agreed amount.

- To some extent the ordering and nesting of elements has been constrained by a requirement to enable a straightforward migration from legacy standards, in particular VDV453 and VDV454, upon which many of the SIRI services are based.

Since SIRI was developed as a harmonisation of existing working national standards that reflect many actual real-time implementation requirements, SIRI is not perfectly uniform in its design or its expression of Transmodel concepts. Nonetheless it achieves a close and useful correspondence.

### TRANSMODEL ENTITIES RELEVANT FOR SIRI

Figure 3-6 shows some key Transmodel schedule related entities that are relevant for SIRI message content, including a couple of SIRI artefacts (for example, **DatedCall**, and **VehicleJourneyInfo**).
SIRI Functional Services - UML Diagrams

See the Transmodel specification for full definitions. The following notes outline the relevance of some key elements:

- A VEHICLE JOURNEY describes a timetabled journey on a public transport route that runs regularly at a particular time: a DATED VEHICLE JOURNEY is a planned journey run on a specific date.

- A VEHICLE JOURNEY follows a JOURNEY PATTERN, which specifies a particular sequence of stops and timings for covering the links between them. A JOURNEY PATTERN may be a specific instance of a more general SERVICE PATTERN associated with a ROUTE, a particular sequence of stops independent of timing. The sequence of STOP POINTs IN SEQUENCE of such patterns identifies the STOP POINTs and order of visit of stops of the vehicle journey. A VEHICLE JOURNEY will have PASSING TIMEs at each stop point. Most SIRI services are concerned primarily with the individual VEHICLE JOURNEYS (in effect a column of a timetable) rather than with the JOURNEY PATTERN, which is typically shared between many journeys of a timetable. However it is sometimes useful to be able to determine the JOURNEY PATTERN in order to access common shared properties of a service, so SIRI messages generally allow the specification of the identifier of the JOURNEY PATTERN of the VEHICLE JOURNEY.

- SIRI groups together the set of passing times and other real-time attributes of a vehicle calling at a stop point as part of a vehicle journey as a Call.

- A dated vehicle journey that is being tracked and monitored by an AVL system (and so for which real time data may be available) is a MONITORED VEHICLE JOURNEY.

- An important point to note is that the labelling of groups of VEHICLE JOURNEYs as belonging to a named LINE (as marketed to the public) and that make up a timetable is independent of any individual journey and may be quite arbitrary: different journeys of the LINE may follow different journey patterns that follow different branches, or that omit stops on some journeys.

- For vehicle and crew scheduling, a DATED VEHICLE JOURNEY may be linked up end-to-end with other dated vehicle journeys to make a BLOCK, comprising one or more runs – or in Transmodel terminology, COURSE OF JOURNEYS. Real-time messages may want to include these references to support operational management systems.

- The SIRI VehicleJourneyInfo group is a named subset of particular VEHICLE JOURNEY attributes which are often required in SIRI content to give context to real time data for a journey. It is shown with a <<group>> stereotype to indicate that in concrete implementations they are embedded in-line. Similar groupings exist for JourneyPatternInfo etc.

- Other variant views of DATED VEHICLE JOURNEY used in SIRI, not shown in Figure 3-6, each including particular subsets of timetabled or real-time attributes include TargetedVehicleJourney, EstimatedVehicleJourney, and InterchangeJourney.
The CEN IFOPT (Identification of Fixed objects in Public Transport) technical specification extends Transmodel to describe the structure of STOP PLACEs – that is, stations, bus stops, airports, and other points of access to transport. One point it clarifies is the distinction between a reference to a stop in a timetable (as a point in a journey pattern of a vehicle journey) – a SCHEDULED STOP POINT, and its possible meanings for a real time application, which may be concerned variously with the STOP PLACE as (i) a group of stops; (ii) an individual QUAY i.e. platform, bay, berth, stop pole, etc. within the STOP PLACE; or (iii) a BOARDING POSITION within a QUAY. This is summarised in Figure 3-7, which shows that a stop point in the timetable may be variously associated with different levels of detail within a stop place. In practice assignment of a SCHEDULED STOP POINT to a STOP PLACE is very done implicitly – by using the same identifier for both, but it might be done more explicitly by an PASSENGER STOP POINT assignment.

**Figure 3-7 IFOPT Stop Assignment**

IFOPT was developed after the SIRI specification, but aspects of it are useful as a conceptual model for understanding SIRI’s use of reference data. It will typically be up to specific applications and implementations to assign appropriate identifiers of stop points to SIRI message elements to reflect the groupings and correspondences for their own configuration. For example, a MonitoringRef of the SIRI Stop Monitoring service might refer to the STOP PLACE if it shows the whole departure board for a station or pair of stops, or to a QUAY if it shows just departures for a particular platform.

The SIRI-SX service was developed after the IFOPT specification and does make explicit use of IFOPT concepts to distinguish components of a STOP PLACE.
PARTICIPANT IDENTIFIERS, MESSAGE & SUBSCRIPTION MANAGEMENT

The requests and responses of SIRI functional services use a common set of header elements to track messages and subscriptions - see Figure 3-8.

- Each Participant system is given a unique identifier. This **ParticipantCode** is used to provide uniqueness of reference for data entities from each system. It is used as the **RequestorRef** on request messages and the **ProducerRef** on response messages.

- Each request message can be given an identifier by the requestor (**MessageIdentifier**) that is echoed back in the response (**RequestMessageRef**). This allows a requestor to match request and responses.

- The Requestor needs to be configured to know the address, i.e. url to which new requests for a SIRI functional service are to be sent. The request can contain an **EndPointAddress** to which responses are to be sent.

- All messages are timestamped with an identifier in UTC.

REQUEST RESPONSE ENTITIES & MESSAGE ELEMENTS

Figure 3-8 Illustrates the common elements involved in SIRI request/response interactions.

- SIRI functional service requests are sent from a **ConsumerService** to a **ProducerService**, which returns a delivery which satisfies the request **Topics** and **Policies**

- A **ServiceRequest** contains one or more SIRI functional requests for specific functional services; each concrete SIRI functional service request (**SiriFsXXXRequest**) inherits common properties from an **AbstractFunctionalServiceRequest**.

- A **ServiceDelivery** contains one or more delivery elements for a specific SIRI functional service (**SiriXxxFsDelivery**) - or **ErrorConditions** if responses cannot be returned. Each SIRI functional service delivery inherits common properties from an **Abstract-FunctionalServiceDelivery**. (Note: Subscription attributes are not populated for deliveries for request/response).

- The **RequestMessageRef** on the **ServiceDelivery** messages references the **MessageIdentifier** of a **ServiceRequest** to which it responds. The **RequestMessageRef** on each individual functional service delivery (**SiriXxxFsDelivery**) within the **ServiceDelivery** references a **MessageIdentifier** on the corresponding functional service request (**SiriXxxFsRequest**).
Figure 3-8 SIRI Request/Response entities
PUBLISH SUBSCRIBE MESSAGE ELEMENTS

Figure 3-8 Illustrates the common elements involved in SIRI publish/subscribe interactions.

- The **SubscribingService** requests subscriptions from a **SubscriptionManager** associated with the **ProducerService** which creates and manages subscriptions.

- Each subscription is given a unique identifier (**SubscriptionCode**) by the subscriber, which is specified on the subscription request for a specific SIRI functional service (**SiriXxxFs-SubscriptionRequest**) as the **SubscriptionIdentifier**, and then included on all deliveries sent back in response to that subscription. The identifier must be unique within the **ParticipantCode** of the subscriber (**SubscriberRef**).

- One or more functional service subscription requests may be submitted at a time as part of a single **SubscriptionRequest**. All must be for the same type of service. In response to the subscription request, a **SubscriptionResponse** message is returned with a **ResponseStatus** containing the identifier of each functional service subscription (**SubscriptionRef**) and whether it has been successfully created.

- A subscription request includes an **InitialTerminationTime** which specifies a desired lease period. This will be echoed back as the **ValidUntil** time on responses.

- The **ProducerService** subsequently monitors the real-time feed and if the request criteria of the subscription are matched, creates deliveries that are sent by the **ProducerService** to the **ConsumerService** nominated on the subscription request. Each delivery message includes the subscription identifier (**SubscriptionRef**) and the **ParticipantCode** of the subscriber, termed the **SubscriberRef**.

- The subscriber, consumer and producer service keep mirrored representations of the active subscriptions. Recovery from system failure or loss of connection is discussed separately below.

- The **SubscribingService** requests needs to be configured to know the address i.e. url of the system to which new subscription requests are to be sent. The request can contain a **ConsumerEndPointAddress** to which delivery responses are to be sent and a distinct **SubscriberEndPointAddress** to which an acknowledgement for the subscription request is to be sent.

- An optional capability of SIRI is to have a **SubscriptionFilter** that will group subscriptions from the same participant for the same type of service so that notifications and delivery data can be sent as a single message. Otherwise each subscription will give rise to separate notification and delivery messages.

- All messages are timestamped with an identifier in UTC.
SIRI Functional Services - UML Diagrams

Figure 3-9 SIRI Publish/Subscribe entities
RECOVERY & RESTART

A SIRI Publish/subscribe interaction involve a long running continuous flow of data from a Producer to a Consumer. Mechanisms are needed to allow recover from system failure of either producer, consumer or communications link. These are described in Part 2 of the SIRI specification. In brief:

A CheckStatus request can be sent between any two participants to see if the other is still working. The response includes a ServiceStartedTime which can be used to detect if the producer has failed and been restarted.

A heartbeat message can be sent from Producer to Consumer at regular intervals to indicate the service is still functioning, even if there is no new real-time data. This includes a ServiceStartedTime that can be used to detect interruption of service.

it is up to the subscribing service to detect interruptions of service and if necessary to terminate and restart subscriptions.

WHAT DOES A SIRI IMPLEMENTATION REQUIRE?

The deployment of a concrete SIRI implementation entails the following:

- The choice of one or more appropriate SIRI Functional Services to meet the application needs.
- A country and or local profile that will set out delivery method, the capabilities to be supported, the data reference system for stop identifiers, etc.
- Allocation of unique Participant Identifiers to identify the different systems.
- A Producer application for the functional service deployed to a server at a known endpoint URL.
- A Consumer application for the functional service deployed to a client.
- A Status service to check that the service is available and working correctly and that there has not been an interruption.
4. SERVICE DESCRIPTIONS

The rest of this guide presents the Subscription, Request & Delivery elements of the SIRI Functional Services described in Part 3 of the SIRI specification as UML Class diagrams.

ORGANISATION

The UML Notation and use of colour is summarised in ANNEX A. Note the use of a <<group>> stereotype on classes that represent a reusable group of related attributes.

For each SIRI Functional service we show:

1. A brief introduction describing the purpose and typical use of the service.
2. A simplified summary of the Subscription & Request elements for the functional service. Note that both are shown on the same diagram, since a subscription embeds a request, though in practice they will be used separately.
3. A more detailed view of the Subscription & Request elements, including data types and abstract supertypes.
4. A simplified summary of the Delivery elements, including payload content that bases relevant parts of Transmodel.
5. A more detailed view of the Delivery elements, including data types and abstract supertypes.

For certain Services (SIRI-SM, SIRI-VM and SIRI-SX) we provide additional summary diagrams of the model elements contained within the delivery.

Note that to allow for compatibility with VDV, for some services the XML schema supports two representations – a nested one and a “flattened” one that omits the nesting tags. The diagrams in this handbook show only the nested representation, which is the normative CEN one.

Services are discussed in the following order:

1. SM Stop Monitoring, VM Vehicle Monitoring.
2. PT Production Timetable, ET Estimated Timetable, ST Stop Timetable.
3. CT Connection Timetable, CM Connection Monitoring.
4. FM Facility Monitoring.
5. GM General Message, SX-Situation Exchange

FUNCTIONAL REQUESTS & RESPONSES

Figure 4-1 shows a summary of the various SIRI functional service requests. One or more functional service requests can be contained within a SIRI ServiceRequest. One or more functional service subscription requests can be contained within a SIRI SubscriptionRequest; each also includes a corresponding functional service request.
Figure 4-1 Summary of SIRI Functional Service Requests

Figure 4-2 shows a summary of the various SIRI functional service delivery messages produced in response to functional service requests. One or more functional service delivery can be contained within a SIRI ServiceDelivery.

Figure 4-2 Summary of SIRI Functional Service Deliveries
SIRI Functional Services - UML Diagrams

5. SIRI STOP MONITORING (SM)

The SIRI Stop Monitoring Service provides a stop-centric view of vehicle arrivals and departures at a designated stop. It can be used by displays and other delivery services to provide departure board and other presentations of timetabled and real-time journey information both at stops and remotely. The choice of data elements to display and the presentation format is up to the client system. The service can be used in conjunction with the SIRI Stop Timetable service, which provides scheduled data.

- The Topics allow a Consumer system to specify that only stop departures and arrivals for a specific Monitoring point, Operator, Line, or Direction are to be returned.
- The Request Policies allow a Consumer system to control the amount of data returned, both in terms of the number of elements and level of detail.
- The Subscription Policies allow a Consumer system to control the change threshold for returning an update, so that for example an update is only sent if prediction has changed by more than a given amount.
- The Delivery message can include not only service arrival and departure times, but also text notices and the identifiers of Situation elements associated with the stop and/or line.

SIRI-SM: SUBSCRIPTION & REQUEST

STOPMONITORINGREQUEST SUMMARY

SIRI-SM Summary StopMonitoringSubscription & StopMonitoringRequest

Figure 5-1 StopMonitoringRequest - Summary
The **StopMonitoringRequest** is used to request stop arrival and departure data for a given stop or stops identified by a **MonitoringRef** - which will normally correspond to the identifier of a physical stop or platform, but might also be a “logical stop” which groups several departure boards.

Other criteria that can be used to filter results include Line (e.g. ‘N45’), Direction, Operator and Destination.

---

**Figure 5-2 StopMonitoringRequest - Detail**
SIRI Functional Services - UML Diagrams

SIRI-SM: DELIVERY

The StopMonitoringDelivery (Figure 5-3) is returned by the SIRI-SM service and normally contains one or more MonitoredStopVisit elements for each monitored stop, each recording a MonitoredVehicleJourney with a MonitoredCall at the stop with passing times. The level of detail to include may vary considerably; for example the full calling pattern of previous and onward calls may be included for each journey. It may also include StopLineNotices associated with the stop. Previous stop visits or line notices may be retracted with a StopVisitCancellation or StopLineNoticeCancellation.

STOPMONITORINGDELIVERY SUMMARY
SIRI SM Summary

StopMonitoringDelivery

Figure 5-3 StopMonitoringDelivery - Summary

STOPMONITORINGDELIVERY DETAIL

Figure 5-4 shows detailed attributes of a Stop Monitoring delivery – further details of the MonitoredVehicleJourney are described in further diagrams below.
Figure 5-4 StopMonitoringDelivery – Detail
**USE OF MONITOREDVEHICLEJOURNEY ELEMENT FOR SIRI-SM**

**LEVELS OF DETAIL**

The `MonitoredVehicleJourney` element represents an individual vehicle journey that is being monitored: it can be populated with different amounts of data for different applications. For example, for the Stop Monitoring service, it might contain just information about the time of the vehicle at the monitored stop (if say the requested `DetailLevel=normal`), or it might also include predicted times at all the subsequent stops along the route (if say `DetailLevel=calls`).

<table>
<thead>
<tr>
<th>Detail Level</th>
<th>Definition.</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum</td>
<td>Return only minimum data: a single <code>MonitoredCall</code> with a stop ids and passing time</td>
<td>Used for exchanging just real-time data between systems that are not concerned with other details.</td>
</tr>
<tr>
<td>basic</td>
<td>Return useful basic minimum data. A single <code>MonitoredCall</code> for each <code>MonitoredVehicleJourney</code> visiting the Stop, with passing times, and line id.</td>
<td>Used to drive simple displays with limited space that show just lines and passing times and line numbers.</td>
</tr>
<tr>
<td>normal</td>
<td>Return a single <code>MonitoredCall</code> at the stop for each <code>MonitoredVehicleJourney</code>, populated with with times, stop id, Stop name, <code>DestinationDisplay</code> values (i.e. headings), status etc.</td>
<td>Use for normal departure boards that show arrival/departure times for each vehicle along with names (but not a calling pattern).</td>
</tr>
<tr>
<td>calls</td>
<td>Return additional information including <code>Onwards</code> and <code>Previous</code> elements with the previous and onward calling pattern, modulated by the <code>NumberOfCalls</code> on the request.</td>
<td>Use for departure boards that need to show the full calling pattern or the onward calling pattern.</td>
</tr>
<tr>
<td>full</td>
<td>Return all information including full calling pattern, operational data (<code>BlockRef</code>, <code>VehicleRef</code>, etc), progress data, journey pattern references etc</td>
<td>Used for data exchange between operational services, or for advanced display services.</td>
</tr>
</tbody>
</table>

**Table 5-1 StopMonitoringRequest Detail Levels**

For train services, where there may be different carriages going to different places representing different vehicle journeys, train block parts can be used to identify which section of the train is involved. There will be separate deliveries for each vehicle journey.

**PASSING TIMES**

The times that are included on each call may include an arrival or departure time or both (depending whether it is the start, end or middle of a route), and planned (`Aimed`), predicted (`Expected`) or observed (`Actual`) - (depending whether it has reached the stop yet. Times are always absolute (i.e. in UTC): it is up to the delivery system to render these – often they will be shown on signs as relative e.g. “in two minutes”. Some systems only return average or predicted headway intervals instead of times.

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Arrival Time</th>
<th>Departure Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>Aimed Arrival Time</td>
<td>Aimed Departure Time</td>
</tr>
<tr>
<td>Estimated</td>
<td>Expected Arrival Time</td>
<td>Expected Departure Time</td>
</tr>
<tr>
<td>Observed</td>
<td>Actual Arrival Time</td>
<td>Actual Departure Time</td>
</tr>
</tbody>
</table>
STOP IDENTIFIERS

Each point for which a `MonitoredStopVisit` is returned is given a unique identifier, the `MonitoringRef`. Normally this will be the same as the stop point, but may be different.

For at stop use, the service can support cleardown identifiers to drive direct wireless cleardown of the displays signalled by the vehicle in proximity.

LOCATION

The `ProgressInfo` group of a `MonitoredVehicleJourney` can include a vehicle location in a set of coordinates of a specified data reference system, e.g. WGS84. The static coordinates of stops are part of the STOP PLACE model and are not normally included in messages.

QUALITY OF AVMS DATA

The `ProgressInfo` group of a `MonitoredVehicleJourney` also includes various attributes that can be used to judge the certainty level of any real-time predictions.
The **MonitoredVehicleJourneyElement** (Figure 5-5) represents a vehicle journey that is being monitored in real-time and for which predictions are being generated. It is used in both the SIRI-SM and SIRI-VM services. Not all elements will usually be populated.

- **MonitoredCall** records the time at the monitored point. **PreviousCall** and **OnwardsCall** can be used optionally to record the times at previous and onward stops.

- **JourneyPatternInfo**, **VehicleJourneyInfo** provide information describing the journey.

- **Progress Info** groups real-time properties such as location, congestion status, occupancy etc.
Figure 5-6 shows the detailed data attributes for a **MonitoredVehicleJourney**.

**MonitoredVehicleJourney**

© 2005 SIRI

Figure 5-6 Monitored vehicle Journey detail
EXAMPLES OF LIVE SERVICES THAT USE SIRI-SM

MAP BASED WEB DEPARTURE BOARDS

http://www.leicestertravel.info.

WAP MOBILE DEPARTURE BOARDS

SMS DEPARTURE BOARDS

UK National SMS service SMS to 84268
6. SIRI VEHICLE MONITORING (VM)

The SIRI Vehicle Monitoring Service reports the position and other real-time properties of a vehicle or group of vehicles making monitored journeys in real-time. It can be used for example to monitor the progress of vehicles; to provide information for systems which present visualisations of the movement of vehicles, for instance on maps, lists or line diagrams; and to exchange information about roaming vehicles between control centres. A particular use is to log data.

SIRI-VM: SUBSCRIPTION & REQUEST

VEHICLEMONITORINGREQUEST SUMMARY

The VehicleMonitoringRequest (Figure 6-1) is used to request real-time data for a vehicle or vehicles. The vehicles can be identified by an arbitrary prearranged grouping, a VehicleMonitoringRef—or other criteria such as a Line (e.g. ‘N45’), Direction, or individual VehicleRef.

- The Topics allow a Consumer system to specify that only vehicle movements for a specific Vehicle Monitoring group, Line, or Line Direction or Vehicle are to be returned.
- The Request Policies allow a Consumer system to control the amount of data returned.
- The Subscription Policies allow a Consumer system to control the change threshold for update.

Figure 6-1 VehicleMonitoringRequest - Summary
Figure 6-2 shows detailed attributes for a **VehicleMonitoring** request.

**Figure 6-2 VehicleMonitoringRequest - Detail**
The **VehicleMonitoringDelivery** (Figure 5-3) is returned by the SIRI-VM service and normally contains one or more **VehicleActivity** elements for each monitored vehicle, each recording a **MonitoredVehicleJourney** with at least some **ProgressInfo**, such as the current **Location**. The level of detail to include may vary considerably; for example, the calling pattern of previous, current and onward calls may be included for each journey. Previously active vehicles may be removed with a **VehicleActivityCancellation**.

---

**Figure 6-3 VehicleMonitoringDelivery - Summary**

© Copyright Kizoom 2006-2007
USE OF MONITOREDVEHICLEJOURNEY ELEMENT FOR SIRI-VM

See also discussion of Passing times and Stop Identifiers under the SIRI-SM service

LEVELS OF DETAIL

The MonitoredVehicleJourney element represents an individual vehicle journey that is being monitored: it can be populated with different amounts of data for different applications. For example, for a simple tracking use Vehicle Monitoring service, it might contain just information about the location of the vehicle at the monitored stop (if say the requested DetailLevel=normal), or it might also included predicted times at all the subsequent stops along the route (if say DetailLevel=calls).

<table>
<thead>
<tr>
<th>DetailLevel</th>
<th>Definition.</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum</td>
<td>Return only minimum data: Location</td>
<td>Used for exchanging just real-time data between systems that are not concerned with other details.</td>
</tr>
<tr>
<td>basic</td>
<td>Return useful basic minimum data to identify the vehicle journey – e.g. Line id - and supply simple ProgressInfo such as a current location,</td>
<td>Used to drive simple movement displays displays with limited space that show just lines and passing times and line numbers.</td>
</tr>
<tr>
<td>normal</td>
<td>Return a single MonitoredCall at the current or next stop for each MonitoredVehicleJourney, populated with times, stop id. Stop name, DestinationDisplay values (i.e. headings), status etc.</td>
<td>Use for on and off board next stop displays and visualisations.</td>
</tr>
<tr>
<td>calls</td>
<td>Return additional information including current MonitoredCall, Onwards, and Previous elements with the previous and onward calling pattern, modulated by the NumberOfCalls on the request.</td>
<td>Use for exchanging data for roaming, or for replicating current vehicle prediction data to distribution systems. Also can be used to feed historical logging services that want to record all the predictions at a given moment in time.</td>
</tr>
<tr>
<td>full</td>
<td>Return all information including full calling pattern, operational data (BlockRef, VehicleRef, etc), progress data, journey pattern references etc</td>
<td>Used for sending data to consumer systems that do not have all the reference data already</td>
</tr>
</tbody>
</table>

Table 6-1 VehicleMonitoringRequest Detail Levels

LOCATION

The ProgressInfo group of a MonitoredVehicleJourney can include the vehicle’s current location as a set of coordinates of a specified data reference system, e.g. WGS84.

The static coordinates of stops are part of the STOP PLACE model and are not normally included in messages.
SIRI Functional Services - UML Diagrams

Figure 6-4 shows more detailed attributes of VehicleMonitoringDelivery, many of which are concerned with supplying references to associated entities. The detailed model of a MonitoredVehicleJourney is the same as for the StopMonitoring service, but will be populated differently.

© Copyright Kizoom 2006-2007
The SIRI Production Timetable Service transmits daily timetables that include any planned updates that are known about at the time of transmission. The service is used typically to communicate between Scheduling systems and AVMS systems, and also between AVMS systems and intelligent clients of the AVMS system to distribute the latest timetables. The timetables exchanged should cover all lines covered by the AVMS system. SIRI-PT may be used in conjunction with the SIRI Estimated Timetable service to provide a base timetable.

The SIRI Production Timetable Service is also used to transmit the planned interchanges between journeys, including information about the linking of vehicle parts through the interchange, such as whether passengers are able to remain seated in the vehicle.

- The Request Topics allow a Consumer system to specify that only timetables for a specific timetable version, Operator, line, or direction are to be returned.
- The Delivery can include the times at stops, block and journey pattern information, and information about available connections.

**SIRI-PT: SUBSCRIPTION & REQUEST**

**PRODUCTIONTIMETABLEREQUEST SUMMARY**

Figure 7-1 ProductionTimeTableRequest - Summary
PRODUCTION Timetable Request Detail

Figure 7-2 shows the detailed attributes of the ProductionTimetableRequest which is used to request the timetable of a given service or services.

Timetables may be requested by operator, Line Direction, Operator and Destination.
In essence the ProductionTimetableDelivery returns a timetable as a Versioned set of DatedVehicleJourney instances, each having two or more DatedCalls, as shown in Figure 7-3 which also shows attributes and associated entity references, for example that a DatedVehicleJourney may have JourneyPatternInfo and ServiceInfo. DatedCalls may also have TargetedInterchange elements giving information about timetabled connections.
Figure 7-4 shows attribute types for a ProductionTimetableDelivery as well.

![UML Diagram](image-url)

---

**Figure 7-4 ProductionTimetableDelivery - Detail**
The SIRI Estimated Timetable service is used by an AVMS or real-time hub to inform interested systems of the current status of all known vehicle journeys. This enables schedule information systems to provide up-to-the-minute information for short-term journey planning. It can also be used to support intelligent displays that calculate the deviations from the timetables themselves using a timetable and a real time difference delay by the SIRI Stop Monitoring Service. A further use is the historic logging of changes to journey times. It can be used to exchange changes to a timetable established by the SIRI Production timetable service.

- The Request Topics allow a Consumer system to specify that only timetables for a specific timetable version, Operator, Line, or Direction are to be returned.
- The Subscription Policies allow a subscriber to specify the amount of change to allow before sending an update.
- The Delivery returns predicted real-time changes to the timetable.

**SIRI-ET: SUBSCRIPTION & REQUEST**

**ESTIMATEDTIMETABLEREQUEST SUMMARY**

![EstimatedTimetableRequest - Summary](image)

Figure 8-1 EstimatedTimetableRequest - Summary
Figure 8-2 shows the detailed attributes of the **EstimatedTimetableRequest** which is used to request the real-time timetable of a given service or services.

Timetables may be requested by operator, **Line, Direction, Operator** and **Destination**.

---

**Figure 8-2 EstimatedTimetableRequest - Detail**

---

**SIRI-ET: DELIVERY**
Figure 7-3 shows the attributes of an EstimatedTimetableDelivery, which the SIRI-ET service uses to return a timetable as a versioned set of EstimatedVehicleJourney instances, each having an ordered collection of EstimatedCalls. These structures are similar to those of a TargetedVehicleJourney elements of the SIRI-PT service but additionally include expected times i.e. real-time predictions as well.

Figure 8-3 EstimatedTimetableDelivery - Summary
Figure 7-4 shows the detailed attributes of an EstimatedTimetableDelivery.

Figure 8-4 EstimatedTimetableDelivery - Detail
9. **SIRI STOP TIMETABLE (ST)**

The SIRI Stop Timetable Service provides a timetabled for vehicle arrivals and departures at a designated stop. It can be used to reduce the amount of information that needs to be transmitted in real-time to stops and displays, as reference data for a Stop Monitoring Service; and provides a data feed of the static timetables.

- The Request Topics allow a Consumer system to specify that only stop timetables for a specific monitoring point, line, or direction are to be returned.
- The Delivery returns departures from the stop for a specified window.

**SIRI-ST: SUBSCRIPTION & REQUEST**

**STOPTIMETABLEREQUEST SUMMARY**

![Figure 9-1 StopTimetableRequest - Summary](image)

© Copyright Kizoom 2006-2007
The **StopTimetableRequest** (Figure 9-1) is used to request stop arrival and departure data for a given stop or stops identified by a **MonitoringRef** - which will normally correspond to the identifier of a physical stop or platform, but might also be a “logical stop” which groups several departure boards.

Other criteria that can be used to filter results include Line (e.g. ‘N45’), Direction, Operator and Destination.

---

**Figure 9-2 StopTimetableRequest - Detail**
The **StopTimetableDelivery** (Figure 5-3) is returned by the SIRI-ST service and normally contains one or more **TargetedStopVisit** elements for each monitored stop, each recording a **TargetedVehicleJourney** with a **TargetedCall** at the stop with passing times.

**STOPTIMETABLEDELIVERY SUMMARY**

![Diagram of StopTimetableDelivery](image)

Figure 9-3 StopTimetableDelivery - Summary
Figure 9-4 Figure 7-4 shows attribute types for a StopTimetableDelivery as well.
10. SIRI CONNECTION TIMETABLE (CT)

The SIRI Connection Timetable Service is used for the exchange of schedule data for potential feeder vehicle journeys to a connection zone. It is normally used in conjunction with the SIRI Connection Monitoring Service to establish the reference data of planned connections to be monitored by the real time systems. The service is location-related, i.e. all requests and replies relate to specific connection links, as identified by connection link identifiers.

- The Topics allow a Consumer system to specify that only Connection Timetables for a specific Connection Link, Line, or Line Direction or Vehicle are to be returned.
- The Delivery returns planned connections as *InterchangeJourneys*

### SIRI-CT: SUBSCRIPTION & REQUEST

**CONNECTION TIMETABLE REQUEST SUMMARY**

![UML Diagram of Connection Timetable Request Summary]

Figure 10-1 ConnectionTimetableRequest - Summary
The **ConnectionTimetableRequest** (Figure 10-2) is used to request stop arrival and departure data for a given stop or stops identified either by a **ConnectionTimeFilter** or by a **ConnectionJourneyFilter** — which will normally correspond to the identifier of a physical stop or platform, but might also be a “logical stop” which groups several departure boards.

Other criteria that can be used to filter results include Line (e.g. ‘N45’), Direction, Operator and Destination.

---

**Figure 10-2 ConnectionTimetableRequest - Detail**
The ConnectionTimetableDelivery (Figure 10-3) is returned by the SIRI-CT service and normally contains one or more TimetabledFeederArrival elements for each monitored connection link, each recording a InterchangeJourney.
Figure 10-4 shows attribute types for a ConnectionTimetableDelivery as well.
The SIRI Connection Monitoring Service exchanges information between different AVMS to coordinate the real-time arrival and departure of PTVs at an interchange through which passengers may make connecting journeys. The departure time of the outgoing ‘distributor’ (or ‘fetcher from’) service may be adjusted to accommodate delays in the incoming ‘feeder to’ service.

The service ensures that the AVMS are in a position to receive all the necessary data concerning the feeder vehicles to allow connection monitoring and dispatch to be carried out, and to inform passengers of changes. The operational methods of dispatch remain unaffected.

The Service can be used in conjunction with the SIRI Connection Timetable Service to exchange scheduled arrival times in the target connection links.

- The Topics allow a Consumer system to specify that only Connection Timetables for a specific Connection Link, Line, or Line Direction or Vehicle are to be returned.
- The Delivery can return information about both changes to feeder (e.g. late arrival) and departure (e.g. late departure, platform change)

**SIRI-CM: SUBSCRIPTION & REQUEST**

**CONNECTIONMONITORINGREQUEST SUMMARY**

---

**Figure 11-1 ConnectionMonitoringRequest - Summary**
The ConnectionMonitoringRequest (Figure 11-2) is used to request information about real-time changes to connecting feeder or distributor journeys, specified either by a ConnectionTimeFilter (identifying a ConnectionLink where changes take place), or by a ConnectionJourneyFilter (identifying a specific journey), which will normally correspond to the identifier of a physical stop or platform, but might also be a “logical stop” which groups several departure boards.

Other criteria that can be used to filter results include Line (e.g. ‘N45′), Direction, Operator and Destination.

Figure 11-2 ConnectionMonitoringRequest - Detail

SIRI-CM: DELIVERY

© Copyright Kizoom 2006-2007 Page 63
The SIRI-CM service returns two types of delivery (Figure 10-3Figure 5-3) that provide information about real-time changes at a monitored connection link.

- A **ConnectionMonitoringFeedDelivery** contains one or more **MonitoredFeedArrival** instances, each with an a **InterchangeJourney** describing the real-time arrival time of a feeder.

- A **ConnectionMonitoringDistributorDelivery** contains different types of change to the departing distributor: a delay (**WaitProlongedDeparture**); a cancellation (**Distribution-DepartureCancellation**); or a platform change (**StoppingPositionChangedRecording**).
Figure 11-4 shows attribute types for a ConnectionMonitoringDelivery.
12. SIRI GENERAL MESSAGE (GM)

The SIRI General Message service is used to transmit messages between the participants. The data to be published will typically be informative messages such as travel news and other operational advice, entered or forwarded into the system - normally by a control centre. The General Message service can segregate different types of informative message into separate information channels; each info channel can be assigned to a different operational message group type (errors, messages, warnings, traffic information, operational messages, etc.).

- The Topics allow a Consumer system to specify that only specific categories of message are to be returned.

The service allows arbitrary content to be embedded. To exchange structured messages see the SIRI-SX service.

SIRI-GM: SUBSCRIPTION & REQUEST

GENERALMESSAGEREQUEST SUMMARY

![Diagram of GeneralMessageRequest Summary](image)

Figure 12-1 GeneralMessageRequest - Summary
The **GeneralMessageRequest** (Figure 12-2) is used to request message data. Data may be filtered by an **InfoChannel**.

![Diagram of GeneralMessageRequest - Detail](image-url)

*Figure 12-2 GeneralMessageRequest - Detail*
The **GeneralMessageDelivery** (Figure 12-3) is returned by the SIRI-GM service and normally contains one or more **GeneralMessage** elements, each recording a **Content** message in a specified format.

**Figure 12-3 GeneralMessageDelivery - Summary**
Figure 12-4 shows attribute types for a **GeneralMessageDelivery**.
13. SIRI FACILITY MONITORING (FM) DRAFT

SIRI-FM: SUBSCRIPTION & REQUEST

FACILITY MONITORING REQUEST SUMMARY
[TODO]
Figure 13-1 Production Time Table Request - Summary

FACILITY MONITORING REQUEST DETAIL
[TODO]

Figure 13-2 Production Timetable Request - Detail

SIRI-FM: DELIVERY

FACILITY MONITORING DELIVERY SUMMARY
[TODO]

Figure 13-3 Production Timetable Delivery - Summary

FACILITY MONITORING DELIVERY DETAIL
[TODO]

Figure 13-4 Production Timetable Delivery - Detail
The SIRI-SX Service is for exchanging Situation content in real-time. It uses a structured Situation model for describing disruptions to services. The structured model includes element references that relate directly to the entities of other SIRI services. Incidents can thus be directly linked to stop points, lines, journeys, pathways, etc: and provide an explanation of the disruption. The service can be used to exchange incident information between control systems, and to distribute to journey planners and alert systems that wish to process and match incidents based on structured elements.

- The Topics allow a Consumer system to specify that only specific categories of message are to be returned.
- The Delivery embeds Structured incidents and updates.

### SIRI-SX: SUBSCRIPTION & REQUEST

#### SITUATIONEXCHANGEREQUEST SUMMARY

Figure 14-1 Summarises the SIRI-SX request elements.
The **SituationExchangeRequest** (Figure 12-2) is used to request message data. Data may be filtered by various structured message attributes such as a Severity, Mode, etc as well as filters for the Network, Line Stop place, etc.
The **SituationExchangeDelivery** (Figure 14-3) is returned by the SIRI-SX service and normally contains one or more **SituationElement** instances; these will either be a **BaseSituationElement** or an **UpdateSituationElement**. A Series of one or more **SituationElements** describes a Situation. Each **SituationElement** has a body (either of type **PISituationBody** or **RoadSituationBody** – not shown) which is itself made up of a number of groups of attributes and child elements (eg **SituationSource**, **Consequence**). For a further summary of the structured model, see below.

---

**Figure 14-3 SituationExchangeDelivery - Summary**

---

© Copyright Kizoom 2006-2007

Page 73
Figure 14-6 shows attribute types for a *SituationExchangeDelivery*. For a further summary of the structured model, see below.

Figure 14-4 SituationExchangeDelivery – Detail
Figure 14-4 shows the main elements and attributes of the *SituationElement*. It has four three basic groups of elements: the *TemporalGroup* (the time over which that Situation applies), *ClassifierGroup* (a categorisation of the situation) and a *DescriptionGroup* (human readable textual descriptions) and *SituationStatus*. Child elements describe the origin (*SituationSource*), *Consequence* and scope of effect (*Affects*).

Figure 14-5 Situation Main elements
The PtSituationBody (Figure 14-6) shows detailed attributes of a SituationElement.
The **AffectsScope** element is used to specify which parts of the network and which services are affected.

**Figure 14-7 SIRI-SX Affects Summary**
Figure 14-8 shows detailed attributes for the scope elements used to relate Situations to parts of the network and to specific journeys.

Figure 14-8 Affects Scope - Scheduled Journeys
SITUATION AFFECTS SCOPE STOP PLACE

Figure 14-9 shows detailed attributes for the scope elements used to relate SituationElements to parts of the stop place. These can include accessibility properties.

Figure 14-9 Affects Scope - Stop Place
SIRI Functional Services - UML Diagrams

15. SIRI COMMON DATA TYPES

COMMON SIRI DATA TYPES - XML SIMPLE TYPES

The SIRI-SX services use XML data types for primitive data types where possible (Figure 15-1).

![XML Data Types](image)

Figure 15-1 XML simple data types

COMMON SIRI DATA TYPES

The SIRI-SX services use a number of common SIRI complex data types (Figure 15-2).

![SIRI Complex Data Types](image)

Figure 15-2 UML Diagram of Common SIRI Data Types
As well as the XML simple data types, SIRI uses a number of additional simple data types see Figure 15-3. Many of these are just explicitly typed identifiers for entities – These have primitive type of string or NMTOKEN).

Figure 15-3 Common SIRI Simple Data types

Figure 15-4 shows simple data types use din SIRI to identify Transmodel entities.

Figure 15-4 Common Identifier types used in SIRI for Transmodel entities
COMMON GENERAL SIRI ENUMERATIONS

The SIRI-SX services use a number of common SIRI enumerations. The common SIRI enumerations are listed in Figure 15-5

Figure 15-5 UML Diagram of SIRI enumerations
Figure 15-6 summaries the enumerations that are specific to SIRI-SX. These also appear in context on individual diagrams.

**Figure 15-6 UML Diagram of SIRI-SX Enumerations**
Figure 15-7 summarises the IFOPT STOP PLACE enumerations that are used in SIRI-SX. These mostly also appear in context on individual UML diagrams.

Figure 15-7 UML Diagram of IFOPT Stop Place Enumerations
Figure 15.8 summarises the miscellaneous TPEG enumerations that are used in SIRI-SX. These mostly also appear in context on individual UML diagrams.

### TPEG Miscellaneous Enumerations

- **TpegDayTypeEnum**
  - unknown
  - monday
  - tuesday
  - wednesday
  - thursday
  - friday
  - saturday
  - sunday
  - mondayToFriday
  - mondayToSaturday
  - weekends
  - publicHoliday
  - holiday
  - regionalHoliday
  - nationalHoliday
  - sundaysAndPublicHolidays
  - schooldays
  - everyday
  - undefinedDayType

- **TpegSeverityEnum**
  - unknown
  - verySlight
  - slight
  - normal
  - severe
  - verySevere
  - noImpact
  - undefined

- **TpegServiceConditionEnum**
  - unknown
  - altered
  - cancelled
  - delayed
  - diverted
  - noService
  - disrupted
  - additionalService
  - specialService
  - onTime
  - normalService
  - intermittentService
  - shortFormedService
  - fullLengthService
  - extendedService
  - splittingTrain
  - replacementTransport
  - arrivesEarly
  - shuttleService
  - replacementService
  - undefinedServiceInformation

- **TpegReportEnum**
  - unknown
  - allTicketClassesValid
  - fullFareOnly
  - certainTicketsOnly
  - ticketWithReservation
  - specialFare
  - onlyTicketsOfSpecifiedOperator
  - noRestrictions
  - noORPpeakTickets
  - noWeekendReturnTickets
  - noReducedFareTickets
  - unknownTicketRestriction

- **TpegTicketRestrictionEnum**
  - unknown
  - network
  - route
  - point
  - individualService
  - undefined

- **TpegRoutePointEnum**
  - unknown
  - startPoint
  - destination
  - stop
  - via
  - notStopping
  - temporaryStop
  - temporaryNotStopping
  - exceptionalStop
  - additionalStop
  - requestStop
  - frontTrainDestination
  - rearTrainDestination
  - throughServiceDestination
  - notVia
  - alteredStartPoint
  - alteredDestination
  - undefinedRoutePoint

- **TpegInterchangeStatusEnum**
  - unknown
  - connection
  - replacement
  - alternative
  - connectionNotHeld
  - connectionHeld
  - statusOfConnection
  - undecided
  - undefined

- **Figure 15.8 UML Diagram of TPEG Enumerations**
TPEG MODE ENUMERATIONS

Figure 15-9 summarises the TPEG mode enumerations that are used in SIRI-SX. These mostly also appear in context on individual UML diagrams.

Figure 15-9 UML Diagram of Tpeg submodes
16. ANNEX A NOTATION

The diagrams in this document follow normal UML notation for class diagrams with the addition of colour (see below), and the use of certain conventions to represent composition as used in XML.

ENTITIES

Classes are indicated by square boxes with the name of the class across the top. Operations and Visibility are omitted. The attributes types, or all of the attributes may be suppressed in summary diagrams, or to show a summary reference.

ENUMERATIONS

Enumerations are generally shown as data types – a square box with an "<<enumeration>>" stereotype. They are included in diagrams in context if space permits, using a dependency relationship (dotted line) from the class with attributes that are constrained by the enumeration. They are also summarised on separate diagrams at the end. Visibilities are omitted.

GROUPS

As well as the normal use of Classes to indicate the entities of the model, classes are also used for named groups of reusable elements which occur on more than one entity, for example *AimedArrivalInfo*, or *ServiceInfo* – see discussion of serialisation and containment below. In this case a stereotype of "<<group>>" is shown. These can be considered as complex data types.

NOTES

Notes are indicated as boxes with turned up corners, generally connected to the class or relationship they annotate with a dotted dependency line.

RELATIONSHIPS

Normal UML relationships are used:

1. Inheritance: line with white arrow from subtype to supertype. The subtype has all the attributes and operations of the supertype.

2. Association: other unbroken lines.

- Cardinalities of associations are marked using UML conventions for multiplicities and optionality, i.e. min:max, for example [0:1] indicates there may be a minimum of zero and a maximum of one, [1:*] indicates there must be a minimum of one and there can be many. [1] by itself means [1:1], [*] by itself means [0:*]. The multiplicities indicate if there are one or many. The optionality indicates whether the end must be populated if the relationship is present.

- Aggregation is indicated by a black diamond (this typically corresponds to direct containment in an XML document): indicating the part is created and destroyed with the whole.
SIRI Functional Services - UML Diagrams

- A shared **composition** is indicated by a white diamond, in which case the child element is integral to the parent component, but the child exists independently (and typically will have a unique identifier).
- Direction of **Navigability** is indicated by an arrow head in the direction of navigability.

3. **Dependency**: Dotted Line. These are also used to show enumerated values

### USE OF COLOUR

To facilitate reading, Classes are coloured to indicate their nature. This is purely a local Handbook convention (not part of UML) and is used as follows

- **Purple**: Common Abstract Message Transport Framework elements. Typically these are the request & response wrapper elements. E.g. **ServiceDelivery** and are the same for all Functional services.
- **Salmon**: Common Abstract Transport Framework elements, Typically these are supertypes. E.g. **AbstractItem**
- **Orange**: Functional Service Elements. E.g. **StopMonitoringDelivery**. These are specific and different for each service, but populated to a common pattern, e.g. with xxxTopics. xxxPolicies, xxxDeliveries etc
- **Yellow**: Domain model elements that correspond to the main payload content of deliveries: typically these are views of Transmodel entities. **Dark yellow** indicates the concrete container class, e.g. **MonitoredVehicleJourney**. **Light Yellow** indicates an embedded reusable element that makes up part of a concrete composite (And may correspond to a Transmodel Entity).
- **White**: References to the identifiers domain model entities, corresponding to the Transmodel concepts.

### SERIALISATION: CONTAINMENT & REFERENCE

The primary concrete expression of SIRI is as an XML schema, for which object references must be serialised either through containment (i.e. expressing an aggregation by embedding a child entity within a parent element’s tags) or reference (i.e. serialising an association by including a reference to the identifier of the associated entity. It is therefore useful to adopt diagramming and naming conventions that indicate whether a particular relationship is expressed in the SIRI XML schema by containment or by reference.

- An explicit attribute is shown on the UML diagrams to indicate an aggregation relationship is implemented as physical **containment**, using the element name indicated by the attribute. The attribute name will be in the plural if the multiplicity is ‘many’. The data type of the attribute will be that of the contained element. For example, the **DatedCalls** attribute in Figure 16-2 below holds multiple instances of **DatedCall**.

- An explicit attribute is shown on the UML diagrams to indicate that an association is serialised as a **reference**. The attribute name on the referring entity generally ends in ‘Ref’ to indicate a reference to another entity, and the data type name generally ends in ‘Code’ or ‘Id’. The data type of the attribute will be the unique identifier of the referenced element.
SIRI Functional Services - UML Diagrams

For example, the StopPointRef attribute in Figure 16-2 below which implements the reference from DatedCall to StopPoint is of type StopPointCode.

In addition, Where attribute values are constrained to particular values a dotted line to a enumeration is shown. E.g. the line to ArrivalActivityEnum in Figure 16-2 below.

Where attributes are grouped as XML groups and used to compose different entities, a class is used to indicate the group. Such classes are usually shown in a lighter shade of colour with a stereotype of <<group>>. For example the AimedArrivalInfo class in Figure 16-2 below.

ALTERNATIVE REPRESENTATIONS OF XML STRUCTURES IN UML

Note that to depict a pure object model in UML one does not strictly need to show an explicit attribute in the parent for a child component (it could be represented just by an association to the contained element), but doing so helps to make clear the order in which attributes appear in the XML and the name of any wrapper tag used to group multiple child instances. Similarly other associations do not strictly need to show the implementing attribute. In the UML diagrams for SIRI we therefore generally show an attribute with which to implement each association.

UML supports a variety of ways for depicting the reuse of data structures, corresponding to different OO programming mechanisms, for example, by inheritance (single or multiple) using either class inheritance or interface conformance; or by aggregation, embedding complex data types in more than one entity. XML allows only single parent class inheritance, so the SIRI XML schema makes greater use of composition than of inheritance, assembling standard data structures (encoded as Groups in XML) into concrete classes. For clarity, we therefore often show these groups in the diagrams as distinct classes with a <<group>> stereotype, even though in the concrete XML they are repeated inline.

We illustrate these differences in Figure 16-1 and Figure 16-2 below, which show two different representations in UML of the same model of a timetable (this is a simplified version of the SIRI Dated Journey).

In Figure 16-1, no attributes are shown to implement the aggregation, and all the attributes are shown in-line. References to external entities are shown as attributes though these too might be omitted (JourneyPatternRef, BlockRef, CourseOfJourneyRef, StopPointRef).
In Figure 16-2, an attribute Calls is shown on DatedVehicleJourney to implement the DatedCalls aggregation. Furthermore, certain of the attributes which occur in groups that are reused elsewhere are shown as separate ‘view’ classes (JourneyPatternInfo, AimedArrivalInfo, AimedDepartureInfo, StopPointInSequence), with a `<group>` stereotype. These are inlined in the XML. Points where extensions may be added are indicated by an Extensions attribute. Operations are not shown.

The data structures are functionally equivalent.

Figure 16-2 Explicit representation of references and of groups
XML FRAGMENT FOR EXAMPLE

The following XML fragment shows a serialisation of some data in an XML document in accordance with Figure 16-2, (This is a simplified version of the actual SIRI DatedVehicleJourney entity)

```
<DatedVehicleJourney>
  <!-- Inherited properties -->
  <JourneyPatternRef>JP56789T</JourneyPatternRef>
  <!-- Specific properties -->
  <!--Journey Pattern Info -->
  <RouteRef>RT0004</RouteRef>
  <DirectionRef>Northbound</DirectionRef>
  <ExtraJourney>false</ExtraJourney>
  <!--Association to Block -->
  <BlockRef>013564</BlockRef>
  <!--Contained children - Calls -->
  <Calls>
    <!-- CALL 1 -->
    <DatedCall>
      <!--Stop point in sequence Group -->
      <StopPointRef>HLTS00101</StopPointRef>
      <StopName>Market Place</StopName>
      <DestinationDisplay>Hospital</DestinationDisplay>
      <!-- Departure Info Group -->
      <AimedDepartureTime>2001-12-17T09:32:47-05:00</AimedDepartureTime>
      <DeparturePlatformName>Stance 1</DeparturePlatformName>
    </DatedCall>
    <!-- CALL 2 -->
    <DatedCall>
      <!--Stop point in sequence Group -->
      <StopPointRef>HLTS00102</StopPointRef>
      <StopName>Hospital</StopName>
      <DestinationDisplay>Station</DestinationDisplay>
      <!-- Arrival Info Group -->
      <AimedArrivalTime>2001-12-17T09:38:47-05:00</AimedArrivalTime>
      <DepartureInfoGroup-->
      <AimedDepartureTime>2001-12-17T09:39:47-05:00</AimedDepartureTime>
    </DatedCall>
    <!-- CALL 3 -->
    <DatedCall>
      <!--Stop point in sequence Group -->
      <StopPointRef>HLTS00103</StopPointRef>
      <StopName>Main Station</StopName>
      <!-- Arrival Group -->
      <AimedArrivalTime>2001-12-17T09:40:47-05:00</AimedArrivalTime>
      <DepartureInfoGroup-->
      <AimedDepartureTime>2001-12-17T09:43:47-05:00</AimedDepartureTime>
      <DepartureBoardingActivity>NoBoarding</DepartureBoardingActivity>
    </DatedCall>
    ...
  </Calls>
</DatedVehicleJourney>
</DatedTimetableVersionFrame>
```
ORDER OF ATTRIBUTES

Attributes appear within classes within the same order as in the XML.

DIRECTION OF READING

Where possible a convention is followed to places parent elements above or left and child elements below, or to the right.

SIMPLE DATA TYPES

XML simple types are used, along with a number of common types such as string tagged with a language attribute. These are generally shown in lower Camel Case, e.g. `dateTime`.

Simple data type names that are defined for SIRI are shown in Upper Camel case.

REUSABLE COMPLEX DATA TYPES

A small number of basic complex type: `Location`, `FacilityChange`, `HalfOpenDate` `FramedVehicleJourneyRef` are used extensively and are not repeated on individual pages. They are shown on a separate page.